Pruning random structures

Arthur Carvalho Walraven da Cunha

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Élagage des structures aléatoires

Arthur Carvalho Walraven da Cunha

Laboratoire d'Informatique, de Signaux et Systèmes de Sophia Antipolis (I3S)
UMR7271 UCA CNRS

Présentée en vue de l'obtention du grade de docteur en INFORMATIQUE **d'**Université Côte d'Azur

Dirigée par : Emanuele NATALE, Chargé de Recherche, Inria, France

Soutenue le : 13 septembre 2023

Devant le jury, composé de :

Vincent GRIPON, Directeur de recherche, IMT-Atlantique, France

Marc LELARGE, Directeur de recherche, Inria, France

Frederic GIROIRE, Directeur de recherche, Inria, France

Konstantin AVRACHENKOV, Directeur de recherche, Inria, France

Pierluigi CRESCENZI, Directeur de recherche, Gran Sasso Science Institute, Italy

Laurent VIENNOT, Directeur de recherche, Inria, France

ÉLAGAGE DES STRUCTURES ALÉATOIRES

Pruning random structures

Arthur Carvalho Walraven da Cunha

Jury :

Rapporteurs

Vincent GRIPON, Directeur de recherche, IMT-Atlantique, France
Marc LELARGE, Directeur de recherche, Inria, France

Examinateurs

Frederic GIROIRE, Directeur de recherche, Inria, France
Konstantin AVRACHENKOV, Directeur de recherche, Inria, France
Pierluigi CRESCENZI, Directeur de recherche, Gran Sasso Science Institute, Italy

Directeur de thèse

Emanuele NATALE, Chargé de Recherche, Inria, France

Membres invités

Laurent VIENNOT, Directeur de recherche, Inria, France

Université Côte d'Azur

Arthur Carvalho Walraven da Cunha

Élagage des structures aléatoires

ix+158 p.

Ce document a été préparé avec LaTeX2e et la classe `these-ISSS` version v. 2.10.

Impression : main.tex – 11/12/2023 – 5:23

Révision pour la classe : these-ISSS.cls,v 2.10 2020/06/24 14:16:37 mpelleau

Élagage des structures aléatoires

Résumé

La *Strong Lottery Ticket Hypothesis (SLTH)* stipule que les réseaux de neurones contiennent, lors de l'initialisation aléatoire, des sous-réseaux qui fonctionnent bien sans aucun entraînement. Le réseau aléatoire doit cependant être sur-paramétré : avoir plus de paramètres qu'il n'en aurait besoin. La SLTH a d'abord été prouvée pour les réseaux entièrement connectés et suppose une sur-paramétrisation polynomiale. Puis, cela a été amélioré pour ne nécessiter qu'un surplus logarithmique, ce qui est essentiellement optimal. Ce fort résultat a tiré parti d'un beau théorème sur le *Subset Sum Problem (SSP)*. Il considère une version aléatoire du SSP dans laquelle on cherche à approximer une valeur cible en sommant des sous-ensembles d'un échantillon aléatoire donné. Le théorème affirme que garantir l'existence d'une solution avec une haute probabilité ne nécessite qu'une taille d'échantillon logarithmique par rapport à la précision des approximations. Nous présentons une preuve plus simple et plus directe pour ce résultat. Ensuite, en tirant parti du théorème sur le SSP, nous étendons le SLTH aux *Convolutional Neural Networks (CNNs)* : nous montrons que les CNN aléatoires contiennent des sous-CNN clairsemés qui n'ont pas besoin d'entraînement pour obtenir de bonnes performances. Nous avons également obtenu le résultat en supposant une sur-paramétrisation logarithmique. Bien que le surplus imposé par le SLTH puisse être compensé par la rareté des sous-réseaux obtenus, exploiter la rareté en pratique est très difficile si elle n'est pas structurée. Étendre les résultats sur le SLTH pour produire des sous-réseaux structurés nécessiterait une version multidimensionnelle du théorème sur le SSP. Nous prouvons la véracité d'une telle version et nous l'utilisons pour montrer que le SLTH est toujours valable pour les CNN si nous exigeons que les sous-réseaux soient structurés. Enfin, nous proposons une application des idées de cette thèse à la conception de circuits : nous exploitons l'aléatoire inhérent aux spécifications des composants électroniques intégrés pour obtenir des composants programmables hautement précis à partir de composants statiques de faible précision.

Mots-clés : Réseau de neurones, Algorithmes des graphes, Compression de modèles, Élagage

Pruning random structures

Abstract

The *Strong Lottery Ticket Hypothesis (SLTH)* states that neural networks contain, at random initialisation, sub-networks that perform well without any training. The random network needs, however, to be over-parameterized: to have more parameters than it would otherwise need. The SLTH was first proved for fully-connected networks and assumed polynomial over-parameterization. Soon after, this was improved to only require a logarithmic overhead, which is essentially optimal. This strong result leveraged a theorem on the *Subset Sum Problem (SSP)*. It considers a randomised version of the SSP in which one seeks to approximate a target value by summing subsets of a given random sample. The theorem asserts that ensuring the existence of a solution with high probability only requires a logarithmic sample size relative to the precision of the approximations. We present a simpler, more direct proof for this result. Then, leveraging the theorem on the SSP, we extend the SLTH to *Convolutional Neural Networks (CNNs)*: we show that random CNNs contain sparse sub-CNNs that do not require training to achieve good performance. We also obtained the result assuming a logarithmic over-parameterization. Even though the overhead imposed by the SLTH could be offset by the sparsity of the sub-networks obtained, exploiting sparsity in practice is very difficult if it is not structured. Extending the results on the SLTH to produce structured sub-networks would require a multidimensional version of the theorem on SSP. We prove such a version and use it to show that the SLTH still holds for CNNs if we require the sub-networks to be structured. Finally, we propose an application of the ideas in this thesis to the design of circuits: We harness the inherent randomness in the specs of integrated electronic components to obtain highly accurate programmable components from low-precision static ones.

Keywords: Neural network, Graph algorithms, Model compression, Pruning

Contents

Generalising the Strong LTH

Application

Annexes

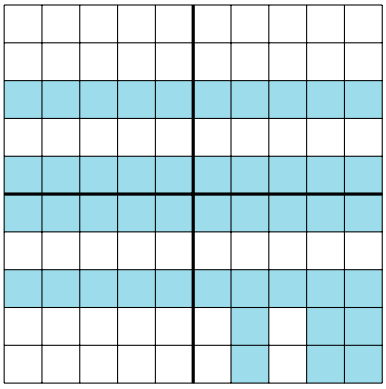
CHAPTER 1

Introduction

"Il semble que la perfection soit atteinte non quand il n'y a plus rien à ajouter, mais quand il n'y a plus rien à retrancher"

— Antoine de Saint-Exupéry, Terre des Hommes.

The diagram below represents a binary grid. Assuming you could freely turn cells *on* or *off*, how would you make it symmetric relative to both axes?



Binary grid experiment. Subjects can freely invert bits. They are asked to make the grid symmetric about the two centralised axes indicated by stronger lines.

Adams, Converse, Hales, and Klotz (2021) experimented with variations of this puzzle and some other tasks of similar nature. They found that the subjects tended to solve the puzzle by *adding* blue cells, neglecting that the task could be solved more easily by *removing* them. The tendency persisted even when subjects were cued with instructions such as "each piece that you add costs ten cents but removing pieces is free",[1] especially when they were under external cognitive load. Combined, the experiments by Adams et al. (2021) suggest that human thinking is significantly biased towards additive strategies.

Among other possible explanations for this heuristic avoidance of subtraction, the authors propose that it may simply be the case that our usual environment, probabilistically, offers more good opportunities to add than to subtract. We highlight one of the examples they bring to illustrate this possibility:

> "In designed environments, one may infrequently encounter artefacts from which the designers have not already subtracted the obviously negative components."

In this thesis, we discuss a subtractive idea in optimisation that is quite easy to overlook. It is uncommon for promising subtractive approaches to be neglected in optimisation since deletion is an integral part of it. The compelling example by Adams et al. (2021) alludes to this. Translated to our context, the example says that optimisation (design) should remove any obviously negative parameters (components) from models (artefacts); the researchers (designers) would spot them. However, we propose that something may be escaping our scissors by hiding at a more abstract level. Something we may neglect to prune in the concept of optimisation itself.

If we want to focus on subtractive strategies, addition easily becomes an "obviously negative component" to remove. Yet, to optimise a model, besides *adding* and *subtracting* parameters, we can also *tune* them. Thus, to isolate the subtraction component, we must also remove the tuning. Because tuning interferes with subtracting and is frequently at odds with it. We all feel this competition when we try to let go of something we have spent a lot of time and effort on. For example, being forced to do so when writing is known as "killing your darlings". Many darling

[1] Similarly to how we biased the reader with the title of the thesis, the abstract, the epigraph, etc.

paragraphs were murdered in the production of this document. Less subjectively, once we start to review the related literature, the intricacy of the interaction between calibration and removal will become evident quite soon. For instance, there has been an entire line of research inspired by the possibility of subtracting parameters from a neural network before effectively tuning what remains. Here, to investigate the capabilities of subtraction, we will completely forego tuning.

In all honesty, there are good reasons to be sceptical of this idea. Perhaps the main one is that without parameter calibration we become especially bound to the initialisation of the model. If we were given a sufficiently convenient initialisation it could mitigate this restriction. For example, consider a LEGO® set: the design of the bricks allows children to achieve quite a lot even though they cannot create new bricks or modify them. However, the problems we will consider all assume absolutely unremarkable, random starting points.

Daunting. This can be a good thing in research, though.

Also, we have essentially removed everything but removal itself. While not making things easier, this does make them simpler. The only tool left is probability. This will allow us to explore ideas in machine learning, algorithms, and electrical engineering while requiring much less background than one may expect.

1.0.1 Organisation of the thesis

After a brief presentation of our notation, in section 1.1 we introduce the general context and motivations of our work. Section 1.2 brings a more technical contextualisation, serving as a base for an overview of our contributions in section 1.3. There, we successively motivate and present the contributions of each chapter.

For the convenience of the reader, we grouped our works on the Subset Sum Problem (chapters 2 and 3) and on the Strong Lottery Ticket Hypothesis (chapters 4 and 5), while our work in circuit design (chapter 6) stands alone.

Finally, while the work (da Cunha, Natale, & Viennot, 2023) was produced during the years this thesis reports, we judged it to be out of the scope of this document and subtracted it from the manuscript.

1.0.2 Notation

Our notation is mostly standard and we revisit it periodically for convenience. We try to present more intricate notations close to the point where we use them. In this way, the immediate application of the definition serves as an example of its use and we avoid overloading the reader with too much notation at once. In the following, we offer more formal versions of definitions contained in the glossary (page 115).

Given $n \in \mathbb{N}$, we denote the set $\{1, \dots, n\}$ by $[n]$. Matrices, vectors, and scalars are considered tensors (of lower order). The symbol $\odot$ refers to the entry-wise (Hadamard) product of two tensors of the same shape. For $p \in \mathbb{R}_{\geq 0}$, the p-norms of tensors are denoted by $\|\cdot\|_p$ Those behave as vector norms rather than operator norms. That is, for a tensor $\boldsymbol{A}$, we have

$$\|\boldsymbol{A}\|_p = \left(\sum_{\boldsymbol{i}} |\boldsymbol{A}_{\boldsymbol{i}}|^p \right)^{\frac{1}{p}}, \tag{$*$}$$

where $\boldsymbol{i}$ goes over all possible indices of $\boldsymbol{A}$. We also consider the special cases $p = 0$ and $p = \infty$, which are taken as the respective limits of equation ($*$). Namely, $\|\cdot\|_0$ represents the number of

non-zero entries of a tensor while $\|\cdot\|_\infty$ denotes the maximum norm: the maximum among the absolute value of each entry. The only operator norm we use is the spectral norm for matrices, given by

$$\|\boldsymbol{M}\|_{\text{spectral}} = \sup_{\boldsymbol{x}\in\mathbb{R}^d:\|\boldsymbol{x}\|_2\leq 1} \|\boldsymbol{M}\boldsymbol{x}\|_2.$$

Finally, we represent the set of all possible sub-networks of f by $\mathfrak{Prune}(f)$.

For the sake of formality, we detail the notation $\mathfrak{Prune}$ a bit further. Given $\boldsymbol{A} \in \mathbb{R}^{\text{shape}}$, we represent by $\mathfrak{Prune}(\boldsymbol{A})$ the set tensors obtained by zeroing a subset of entries of $\boldsymbol{A}$. Namely,

$$\mathfrak{Prune}(\boldsymbol{A}) :- \left\{ \boldsymbol{m} \odot \boldsymbol{A} \,\middle|\, \boldsymbol{m} \in \{0,1\}^{\text{shape}} \right\}.$$

Since we denote a neural network $f(\,\cdot\,;\boldsymbol{\theta})$ simply as f when $\boldsymbol{\theta}$ is clear from the context, in this situation we also write $\mathfrak{Prune}(f)$ instead of the cumbersome $\{f(\,\cdot\,;\boldsymbol{\theta}')\}_{\boldsymbol{\theta}'\in\mathfrak{Prune}(\boldsymbol{\theta})}$.

Styles

a	A scalar (integer or real)
$\boldsymbol{a}$	A vector
$\boldsymbol{A}$	A matrix
$\boldsymbol{\mathsf{A}}$	A tensor
$\mathcal{A}$	A set
$\boldsymbol{\mathcal{A}}$	A family (of sets)
A	A scalar random variable
$\mathbf{A}$	A vector-valued random variable
$\mathbf{A}$	A matrix-valued random variable
$\boldsymbol{\mathsf{A}}$	A tensor-valued random variable
$\mathcal{A}$	A set-valued random variable

Table 1.1: Font styles associated with mathematical types. Usual objects are typeset in italics while their random variants use upright styles.

a_i	Element i of vector $\boldsymbol{a}$, with indexing starting at 1
$A_{i,j}$	Element i, j of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{i,:}$	Row i of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{:,i}$	Column i of matrix $\boldsymbol{A}$
$\boldsymbol{\mathsf{A}}_{i,j,k}$	Element (i, j, k) of a 3-D tensor $\boldsymbol{\mathsf{A}}$
$\boldsymbol{\mathsf{A}}_{:,:,i}$	2-D slice of a 3-D tensor
A_i	Element i of the random vector or matrix $\mathbf{A}$
A_i	Element i of the random tensor $\boldsymbol{\mathsf{A}}$

Table 1.2: Notation for indexation. Usual objects are typeset in italics while their random variants use upright styles.

Tables 1.1 and 1.2 are based on (Goodfellow, Bengio, & Courville, 2016, Notation).

1.1 Pruning is all you need

Deep neural networks are rapidly becoming the state-of-the-art method in many tasks across a wide range of domains, both by replacing previous techniques and by enabling new applications. Such progress has come with an accordingly fast increase in network complexity, particularly in parameter count, with modern networks often containing many millions or even billions of parameters. For example, from the AlexNet (Krizhevsky, Sutskever, & Hinton, 2012) to the AlphaGO Zero (Silver et al., 2017) milestones, the amount of computational power dedicated to training increased by a factor of 300,000 (Amodei et al., 2018). This "scale is all you need" paradigm has created an explosion in the overall computational cost of deep learning since larger models not only are more expensive to train and run but have higher demand due to their better capabilities. At the same time, the success of neural networks makes them increasingly desirable to deploy in more platforms (Balas, Roy, Sharma, & Samui, 2019), including those with constrained resources, such as mobile devices and embedded systems.

Fortunately, those models are known to be much larger than necessary. To such an extent that they can easily fit randomly labelled data (C. Zhang, Bengio, Hardt, Recht, & Vinyals, 2017) and that 5% of a model's parameters can be used to predict the other 95% (Denil, Shakibi, Dinh, Ranzato, & de Freitas, 2013). Perhaps the most striking example of this over-parameterization is the empirical success of pruning, the process of removing weights from a network by setting them to zero or completely erasing them from the architecture. Once the network is trained, pruning techniques can commonly reduce its parameter count by more than 90% with little to no loss in performance (Han, Pool, Tran, & Dally, 2015), and good accuracies have been reported even after pruning 99.9% of the parameters (Lin, Stich, Barba, Dmitriev, & Jaggi, 2020). We refer the reader to the surveys Hoefler, Alistarh, Ben-Nun, Dryden, and Peste (2021) and Blalock, Ortiz, Frankle, and Guttag (2020) for a comprehensive overview of pruning methods.

This striking level of over-parameterization naturally puts into question the need for large networks. The pragmatical reader could suspect that the cost of training is ultimately negligible since it only happens once and the resulting model is used for countless inferences. However, the sheer size of modern neural networks made the cost of training them difficult to offset. For instance, between 2019 and 2021, the total energy dedicated to ML at Google was estimated at 40% for training and 60% for inference (Patterson et al., 2022). The success of heavily over-parameterized neural networks is also a challenge to the theory. Common deep architectures can be trained to reach zero loss on train data and still perform significantly well on test data (C. Zhang et al., 2017). This phenomenon subverts the classical understanding that "a model with zero training error is overfit to the training data and will typically generalize poorly" (Hastie, Tibshirani, & Friedman, 2009, page 221) and, more broadly, the central ML concept of bias-variance trade-off.

Yet, training large dense models still prevails. The belief that over-parameterization is necessary for training has been supported by the influential information bottleneck theory (Tishby & Zaslavsky, 2015; Shwartz-Ziv & Tishby, 2017), even though the theory has been disputed (Saxe et al., 2018). More decisively, the most natural approaches to training small networks have failed. Han et al. (2015), H. Li, Kadav, Durdanovic, Samet, and Graf (2017), and See, Luong, and Manning (2016) tried the general algorithm of

1. training an over-parameterized network;
2. pruning it to a smaller size;
3. re-initialising the weights remaining in the pruned network;

4. re-training the pruned network;

where step (3) consists in sampling those weights again from the distribution used to initialise the original network. The studies found that the resulting networks performed meaningfully worse than the original pruned ones, obtained at step (2). Han et al. (2015), which was particularly influential on modern pruning, concluded that "It is better to retain the weights from the initial training phase for the connections that survived pruning than it is to reinitialize".

1.1.1 The original Lottery Ticket Hypothesis

This context made it quite surprising when a slight variation of the above algorithm was found to successfully produce trainable sparse networks. Instead of re-initialising the weights of the pruned network in step (2), Frankle and Carbin (2019) proposed to rewind them to their initial values, i.e., to the values originally sampled when first initialising the full network. The authors provided extensive empirical evidence that, on small-scale vision tasks, those rewound sparse sub-networks are trainable: they can match (and sometimes even surpass) the performance of the original dense network after training for at most the same number of iterations. This observation led the authors to conjecture the generality of the phenomenon.

Conjecture 1.1.1 (Lottery Ticket Hypothesis (LTH) – Frankle & Carbin, 2019)**.** *Practical neural networks contain sparse, trainable sub-networks at random initialization.*[2]

Notably, the empirical method presented in Frankle and Carbin (2019) requires first training the original network to completion, before pruning it[3]. Thus, this technique does not provide a direct way to reduce the overall cost of training. However, it does prove the existence of sparse sub-networks that "won the initialisation lottery", receiving a combination of weights and structure that makes them inherently trainable. Those sub-networks are called *winning tickets*. They can be as sparse as networks obtained by pruning for efficiency only (Renda, Frankle, & Carbin, 2020, Appendix E), so their occurrence implies that, in principle, successfully training small networks is possible, after all.

The original LTH has, since, faced several challenges (Hoefler et al., 2021, subsection 8.3.2). Primally, it has failed to generalise to large-scale settings (Frankle & Carbin, 2019), leading the original authors to relax the hypothesis to allow for the existence of winning tickets in early steps of training rather than at initialisation (Frankle, Dziugaite, Roy, & Carbin, 2020). Nonetheless, the promise of making training much more efficient and the possibility of fresh insights into the role of over-parameterization have motivated intense research on the LTH, as surveyed in Frankle (2023, Chapter 5).

1.1.2 The *Strong* Lottery Ticket Hypothesis

> *"To attain knowledge, add things every day. To attain wisdom, subtract things every day"*
>
> — Lao Tzu, Tao Te Ching.

[2] This is the phrasing chosen by the main author in Frankle (2023).

[3] Which demands multiple extra rounds of training to convergence, as the authors employ iterative magnitude pruning (Janowsky, 1989; Han et al., 2015).

1.1.2.1 Empirical motivations

Among the many investigations into the LTH, Zhou, Lan, Liu, and Yosinski (2019) brought some empirical support to an even more staggering phenomenon: the existence, at initialisation, of sub-networks that can perform unexpectedly well without any training. While analysing winning tickets, the authors noticed that even without training they performed meaningfully better than random (e.g., almost 40% accuracy on MNIST (LeCun, Cortes, & Burges, 2010)). Motivated by this observation, the authors devised an algorithm dedicated to locating what they called *supermasks*: sub-networks of random DNNs that achieve high performance before any optimisation of their parameters.

Their proposed method consists of randomly initialising a dense network and associating a score s to each weight w in the network. The core idea is to use the score s to decide whether or not to prune w and to optimise s via Stochastic Gradient Descent (SGD) while w is never modified. During the forward pass, keep w with probability $\sigma(s)$, where σ is the sigmoid function. That is, replace w with $w \cdot \text{Bernoulli}(\sigma(s))$, where the sampling takes place at each forward pass. For the backward pass, use the value of $\sigma(s)$ to back-propagate (Rumelhart, Hinton, & Williams, 1986a) the gradient through the Bernoulli sampling. Using this method, Zhou et al. (2019) obtained untrained networks that could reach accuracies over 95% on MNIST and 65% on CIFAR-10 (Krizhevsky & Hinton, 2009).

Suspecting that constantly sampling new networks might limit the performance of SGD, Ramanujan, Wortsman, Kembhavi, Farhadi, and Rastegari (2020) proposed to simply keep a target percentage of the weights with the highest score. More precisely, for each layer, each weight w behaves as $w \cdot \text{WTA}_k(s)$, where

$$\text{WTA}_k(s) = \begin{cases} 1 & \text{if } s \text{ is among the } k\% \text{ highest scores of its layer,} \\ 0 & \text{otherwise.} \end{cases}$$

The operator WTA_k is called *winner-takes-all*. It is not differentiable so Ramanujan et al. (2020) use the straight-through estimator (Bengio, Léonard, & Courville, 2013) for it, that is, it is treated as the identity function in the backward pass.

With this method, named EDGE-POPUP, Ramanujan et al. (2020) discovered sub-networks that could perform well without training within large-scale randomly initialised networks. Namely, experimenting on ImageNet (Deng et al., 2009), the authors showed that a Wide ResNet-50 (Zagoruyko & Komodakis, 2016) contains, at initialisation, a sub-network that is smaller than, but matched the performance of, a trained ResNet-34 (K. He, Zhang, Ren, & Sun, 2016). Similarly, a randomly-weighted ResNet-101 (K. He et al., 2016) holds a sub-network that is much smaller than VGG-16 (Simonyan & Zisserman, 2015) while performing better than it.

1.1.2.2 The hypothesis

Motivated by their empirical findings, the authors conjectured the generality of the phenomenon. The conjecture was later named the *Strong Lottery Ticket Hypothesis (SLTH)* (Malach, Yehudai, Shalev-Shwartz, & Shamir, 2020).

Conjecture 1.1.2 (Strong Lottery Ticket Hypothesis – Ramanujan et al., 2020)**.** *Within any sufficiently over-parameterized neural network with random weights (e.g., at initialisation), there exists, with high probability, a sub-network that performs well without any training. Specifically,*

the sub-network can match the test performance of a trained network with the same number of parameters. (Rephrased)

We will later precise the concept of "sufficient over-parameterization" in different ways depending on the context. For now, consider a generic measure of architecture complexity, provisionally referred to as "*size*".

Let G_{random} be a randomly initialised network. Given a target (trained) network f_{target}, Conjecture 1.1.2 associates to its architecture a *size* N such that if G_{random} is *larger* than N, then there exists, among all sub-networks of G_{random}, a g_{sub} that performs as well and has at most as many parameters as f_{target}. This is claimed with high probability on the sampling of the parameters of G_{random}, as it cannot hold deterministically.

In particular, the SLTH holds trivially if we allow for exponential over-parameterization since, for any target network, one would be likely to find a tight approximation of each of its parameters among so many random ones. Under this light, this text only concerns arguments for the hypothesis within polynomial bounds on N.

Even though the SLTH ensures the existence of sub-networks that do not require weight tuning, Conjecture 1.1.2 is still tied to training: it uses a "trained network" as a reference for a "good performance". Since our theoretical understanding of training is still limited, this can be a major hurdle in tackling the conjecture. Next, we will see that by making the hypothesis even stronger we can sidestep training altogether, ultimately making it easier to prove.

1.1.2.3 The *Stronger* Lottery Ticket Hypothesis

Soon after Zhou et al. (2019) and Ramanujan et al. (2020) raised attention to the SLTH, Malach et al. (2020) proved something even stronger. Section 1.2.1 provides a precise statement of their result as well as an overview of its proof. For now, however, their main result can be distilled as follows.

> [**Informal version of Theorem 1.2.1**] For any desired confidence and accuracy, *any* fully connected network can be approximated by pruning a random network which is two times deeper and has polynomially more neurons per layer.

This claim is stronger than the SLTH in that it ensures approximations of *any* network with a given architecture (width and depth). In particular, sufficiently over-parameterized networks contain, at initialisation, a sub-network that approximates the best possible set of weights for a given task, regardless of whether an optimisation process–no matter the kind–can find it. Thus, to conclude a formal version of the LTH, it is enough to assume that the architecture associated with the target network is sufficient to solve the task at hand optimally. This is the reason why we refer to the additional size of the random network (here, the width and depth) as *over-parameterization*.

The version of the LTH that we obtain by completely trivialising the notion of training, as above, diverges considerably from the essence of the original hypothesis. As an example that aggravates the matter, the sub-networks found by the algorithm introduced by Ramanujan et al. (2020) do not reach better performance when trained (Frankle, 2023). Hence, the two hypotheses are currently perceived as distinct objects of research, with investigations on the original LTH focusing on the dynamics of training while works on SLTH usually exploring the combinatorics of large networks.

1.1.2.4 The impact of the SLTH

Although definitely intriguing, the SLTH may first appear to be of little practical relevance. For instance, the algorithm proposed by Ramanujan et al. (2020), EDGE-POPUP, comes with a significant overhead when compared to usual training methods. Storing a score for each parameter of the random network and repeatedly constructing sub-networks is computationally expensive. Moreover, the technique tends to perform best when searching for sub-networks with sparsity levels of around 50%,[4] which is too low for many use cases.

However, being a novel approach, techniques around the SLTH have much potential for improvement. Indeed, multiple studies have already built upon EDGE-POPUP to make it significantly more efficient and produce sub-networks that perform better while also being sparser (Chen, Zhang, & Wang, 2022; Koster, Grothe, & Rettinger, 2022; Y. Zhang et al., 2021).

Furthermore, by putting pruning as a sound alternative to training, the SLTH provides a fresh perspective on neural network optimisation which is fertile ground for new ideas. As an example, Diffenderfer and Kailkhura (2021) proposed a method based on EDGE-POPUP that can train binary networks to match, and sometimes even surpass, the performance of their full-precision counterparts, reaching state-of-the-art results on the CIFAR-10 and ImageNet. Binary networks are the extreme case of quantised networks, as their weights are constrained to be either $+1$ or -1. Besides vastly reducing the memory footprint of the network, binary weights also allow for massive power savings and faster inference since the expensive multiply-accumulate operations can be replaced by simple XNOR and bit counting instructions (Qin et al., 2020). Multiple works have followed since with increasing success (Cheng et al., 2022; Gorbett & Whitley, 2023).

In another direction, the SLTH provides a promising way to overcome *catastrophic forgetting*, when DNNs experience a severe loss in performance on previously learned tasks upon being trained on a new one. In the context of the SLTH, however, the weights are not modified. Wortsman et al. (2020) and Kang et al. (2022) leverage this property to discover, within a single fixed random network, multiple sub-networks that are optimised for different tasks without interfering with each other.

The hypothesis has also inspired methods for Federated Learning (Pase, Isik, Gunduz, Weissman, & Zorzi, 2022; A. Li et al., 2021; Mozaffari, Shejwalkar, & Houmansadr, 2023; Vallapuram et al., 2022) and in chapter 6 we will propose applications to the design of integrated circuits.

More generally, the SLTH has provided a reference point for investigating training dynamics and over-parameterization. We can start by only allowing SGD to select which weights are kept, as in EDGE-POPUP, and then gradually relax the constraints to study the optimisation process. Before allowing for full control of the weights, we can, for instance, experiment with restricting weights to a small set of random values (Aladago & Torresani, 2021), permitting signs to flip (Koster et al., 2022; Chen, Zhang, & Wang, 2022), adding small amounts of noise (Xiong, Liao, & Kyrillidis, 2023), or re-randomising weights (Chijiwa, Yamaguchi, Ida, Umakoshi, & Inoue, 2021).

Finally, the reader will notice that in our extensive theoretical discussion of the SLTH, we will not deal with training whatsoever. There will be no references to gradients, backpropagation, or loss functions. The discussion will instead have an (extremal/probabilistic) combinatorial flavour: how large should the random network be to ensure a good probability of finding a sub-network that approximates a target? In principle, knowledge about artificial neural networks is only needed

[4] This is, in principle, to be expected as the sparsity of a uniformly sampled sub-networks is concentrated around 50% sparsity.

to relate the structure of the graphs to their behaviour as functions. This trait makes the SLTH a particularly inviting object of study for mathematicians and computer scientists who want to step into machine learning utilising their usual toolkit.

1.2 Technical context

In this section, we introduce the technical context in which our work took place. More precisely, we overview the arguments that immediately predated ours. We present results in their full formality, but we will only discuss the main ideas behind the proofs.

1.2.1 The original SLTH proof

In this section, we go over the main ideas behind the first result proving a version of the SLTH with polynomial over-parameterization, which we already discussed informally in section 1.1.2.3.

Theorem 1.2.1 (Malach et al., 2020)**.** *Let $\varepsilon, \delta \in (0,1)$. Given $\ell \in \mathbb{N}$, let $d_0, \dots, d_\ell \in \mathbb{N}$. Let $\boldsymbol{\mathcal{F}}$ be the class of functions from $[-1,1]^{d_0}$ to $\mathbb{R}^{d_\ell}$ such that for each $f \in \boldsymbol{\mathcal{F}}$,*

$$f(\boldsymbol{x}) = \boldsymbol{W}_\ell \operatorname{relu}(\boldsymbol{W}_{\ell-1} \cdots \operatorname{relu}(\boldsymbol{W}_1 \boldsymbol{x})),$$

where for $i \in [\ell]$ we have that $\boldsymbol{W}_i \in \mathbb{R}^{d_i \times d_{i-1}}$, $\|\boldsymbol{W}_i\|_\infty \leq 1/\sqrt{d_i}$, and $\|\boldsymbol{W}_i\|_{\text{spectral}} \leq 1$. Finally, let $G\colon \mathbb{R}^{d_0} \to \mathbb{R}^{d_\ell}$ be a 2ℓ-layered random network given by

$$G(\boldsymbol{x}) = \mathbf{V}_{2\ell} \operatorname{relu}(\mathbf{V}_{2\ell-1} \cdots \operatorname{relu}(\mathbf{V}_1 \boldsymbol{x})),$$

where the parameters of G are i.i.d. random variables following $\operatorname{Uniform}([-1,1])$ and for $i \in [\ell]$ the weight matrices $\mathbf{V}_{2i-1}$ and $\mathbf{V}_{2i}$ have shape $d_i n_i \times d_{i-1}$ and $d_i \times d_i n_i$, respectively, so that n_i is an integer over-parameterization factor.

Then, there exists a universal constant $C > 0$ such that if, for $i \in [\ell]$,

$$n_i \geq \frac{C d_i^4 \ell^2}{\varepsilon^2} \log \frac{d_i^2 \ell}{\delta}, \tag{1.1}$$

then, with probability at least $1 - \delta$,

$$\sup_{\boldsymbol{x} \in [-1,1]^{d_0}} \sup_{f \in \boldsymbol{\mathcal{F}}} \min_{g \in \mathfrak{Prune}(G)} \|f(\boldsymbol{x}) - g(\boldsymbol{x})\|_\infty < \varepsilon. \tag{1.2}$$

That is, considering a target network class $\boldsymbol{\mathcal{F}}$, suppose we have a random network G which is two times deeper and polynomially wider than networks in $\boldsymbol{\mathcal{F}}$. Theorem 1.2.1 states that for any given confidence and accuracy, one can prune G to approximate any network in $\boldsymbol{\mathcal{F}}$. It is noteworthy that the event of Theorem 1.2.1 is over the random initialisation of G and ensures that by pruning a single fixed network (usually a randomly initialised one) we can approximate any network in $\boldsymbol{\mathcal{F}}$. For instance, in the context of figure 1.1, for a fixed set of parameters for G we can approximate any f by deleting edges from G.

The approach underlying the proof of Theorem 1.2.1 is to leverage the extra neurons in G to dedicate a substructure to each parameter w in the target network. As highlighted in figure 1.2, this approach effectively provides a random gadget which we can control via pruning to approximate the behaviour of w. The strategy is to first establish an upper bound on the size, n, of the gadget

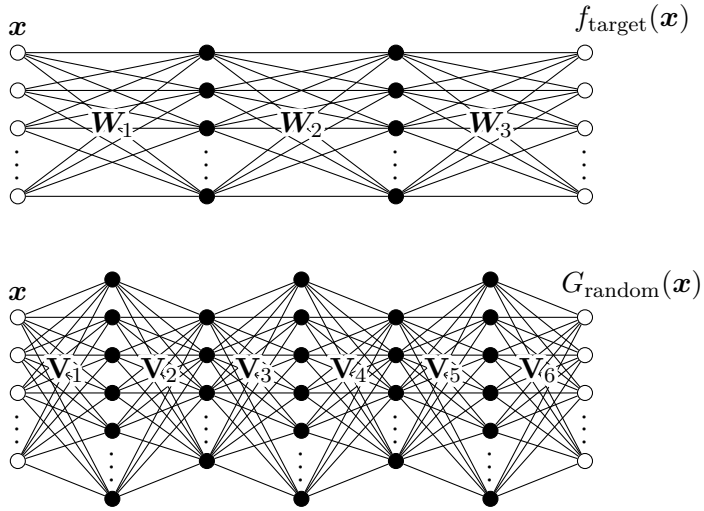


Figure 1.1: Illustration a neural network $f \in \boldsymbol{\mathcal{F}}$ with a 3-layered architecture (above) and the structure of the associated random network G (below) as in Theorem 1.2.1. The activation of filled nodes (in black) is computed via a non-linearity (relu).

that guarantees the approximation of w with specified confidence and accuracy. Then, we analyse the relationship between the global error in equation (1.2) and the error in each approximation, so we can ensure that the individual approximations are tight enough to yield the desired global accuracy. Finally, by proceeding similarly for the confidence, we can conclude the proof by taking a union bound over all parameters in the target network.

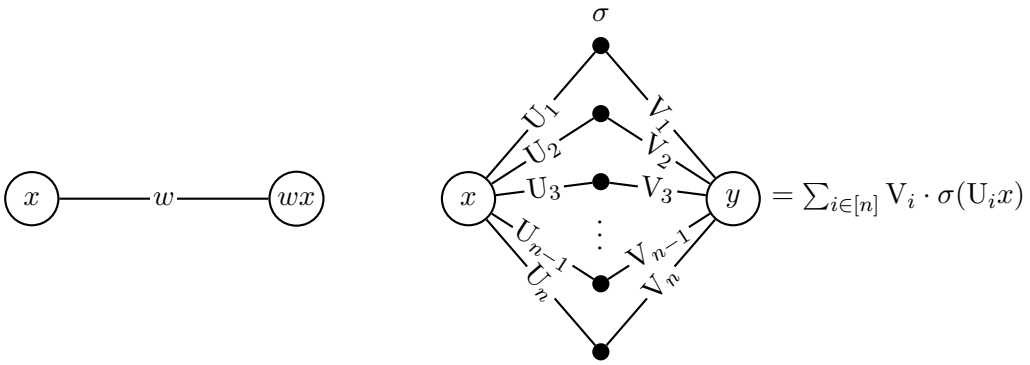


Figure 1.2: Illustration of the gadget used in the proof of Theorem 1.2.1. For each parameter w (on the left) in the target network, we dedicate a substructure to approximate it (on the right). Notice that w is a fixed given scalar, while the U_is and V_is are random variables. For the sake of generality, here we consider an arbitrary activation function, σ.

Therefore, we can start by focusing on the approximation of a single parameter $w \in [-1/\sqrt{n}, 1/\sqrt{n}]$. To understand how to control the behaviour of the gadget represented in figure 1.2, we first consider the case where the activation function σ is the identity. As the left diagram in figure 1.3 shows, in this case, by pruning all other edges in the gadget[5] we can select

[5] One can effectively prune both edges by setting $\mathrm{U}_i = 0$ or $\mathrm{V}_i = 0$ (or both).

i^* such that $\mathrm{U}_{i^*} \approx w$ and $\mathrm{V}_{i^*} \approx 1$ (or vice-versa), so that $y = \sum_{i \in [n]} \mathrm{V}_i \mathrm{U}_i x = \mathrm{V}_{i^*} \mathrm{U}_{i^*} x \approx wx$. Given $\varepsilon' \in (0, 1)$, since $\mathbf{U}, \mathbf{V} \sim \mathrm{Uniform}\,[-1, 1]^n$, we have that, for all $i \in [n]$,

$$\Pr\big[|w - \mathrm{U}_i| \leq \varepsilon'\big] \geq \frac{\varepsilon'}{2}, \tag{6}$$

and

$$\Pr\big[|1 - \mathrm{V}_i| \leq \varepsilon'\big] = \frac{\varepsilon'}{2}.$$

As the experienced reader may have anticipated, we can conclude that ensuring the existence of a suitable i^* with probability at least $1 - \delta'$ requires that

$$n = \mathcal{O}\left(\frac{1}{\varepsilon'^2} \log \frac{1}{\delta'}\right).$$

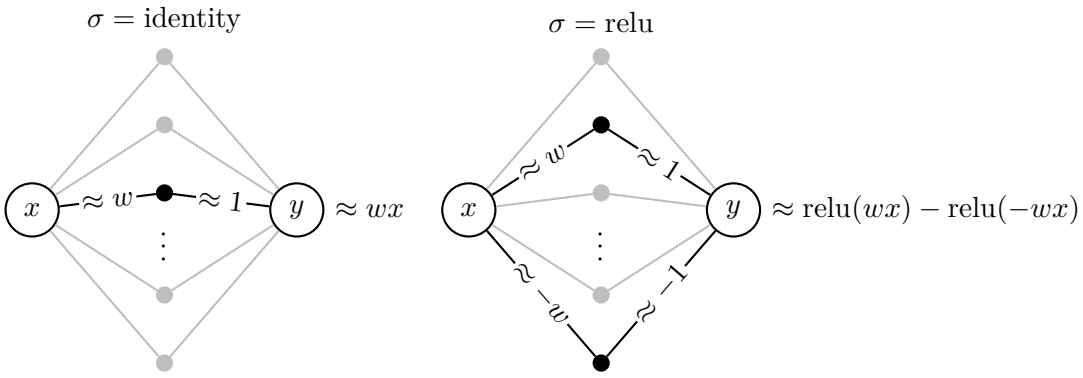


Figure 1.3: Pruning scheme to approximate w for the cases where the activation function σ is the identity (left) or relu (right).

Now, when the activation function σ is the relu, as in Theorem 1.2.1, we rely on the identity

$$x = \mathrm{relu}(x) - \mathrm{relu}(-x), \tag{1.3}$$

which holds for all $x \in \mathbb{R}$. To this end, as depicted in the right diagram in figure 1.3, we can select indices $i^+, i^- \in [n]$ that behave as i^* depending on the sign of x. Namely, we seek $\mathrm{U}_{i^+} \approx w$ and $\mathrm{V}_{i^+} \approx 1$, and $\mathrm{U}_{i^-} \approx -w$ and $\mathrm{V}_{i^-} \approx 1$.[7] For such indices, assuming without loss of generality that w is positive, if $x > 0$, we have that

$$\begin{aligned} \mathrm{V}_{i^+} \cdot \mathrm{relu}(\mathrm{U}_{i^+} \cdot x) + \mathrm{V}_{i^-} \cdot \mathrm{relu}(\mathrm{U}_{i^-} \cdot x) &\approx 1 \cdot \mathrm{relu}(wx) - 1 \cdot \mathrm{relu}(-wx) \\ &= wx - 0 \\ &= wx \end{aligned}$$

and, if $x < 0$, similarly,

$$\begin{aligned} \mathrm{V}_{i^+} \cdot \mathrm{relu}(\mathrm{U}_{i^+} \cdot x) + \mathrm{V}_{i^-} \cdot \mathrm{relu}(\mathrm{U}_{i^-} \cdot x) &\approx 0 - (-wx) \\ &= wx, \end{aligned}$$

[6] The probability is equal to ε' if $w \pm \varepsilon' \in [-1, 1]$.

[7] Once again, the order could be flipped.

where, for the sake of simplicity, we assumed the approximation to be precise enough to preserve signs. Since the cases $x = 0$ and $w = 0$ are trivial, the analysis above suffices to show that suitable pruning can shape the gadget in figure 1.2 into emulating any parameter of the target network.

The rest of the argument consists of using the strategy to approximate progressively larger structures from the target network: a neuron, then a layer, then multiple layers. Accordingly, we require progressively smaller values of δ' to keep the overall confidence within the threshold guaranteed in the statement of Theorem 1.2.1. Effectively, a union bound is used at each step of this process to ensure that the probability of failure is at most δ. In a similar way, each step demands gradually smaller values of ε' so that the accumulated error is below the threshold ε in the statement. The restrictions on the norms of the weight matrices of f and its domain, $[-1, 1]^{d_0}$, serve to constrain the error propagation. The following remark highlights that any approximation result about ReLU networks must make some form of those hypotheses.

Remark 1.2.1 – The ReLU function is *positive homogeneous*, that is, $\mathrm{relu}(\alpha x) = \alpha\,\mathrm{relu}(x)$ for all $\alpha \geq 0$ and $x \in \mathbb{R}$, thus, so are networks that have relu as the activation function. Therefore, if two such ReLU networks $f, g\colon \mathbb{R}^d \to \mathbb{R}$ disagree to any extent, then the divergence of their outputs can be made arbitrarily large. More precisely, if $|f(x) - g(x)| = \varepsilon > 0$ for some $x \in \mathbb{R}^n$, then $|f(\alpha x) - g(\alpha x)| = \alpha\varepsilon$ for all $\alpha \geq 0$.

In particular, for the type of dense network considered in Theorem 1.2.1 we can apply the same reasoning to the weight matrices (via linearity) and conclude that their norm must be bounded. Though there is no need to specify which norm is used, as they are all equivalent in finite-dimensional spaces, this context makes the use of the spectral norm more natural since

$$\|\boldsymbol{A}\|_{\text{spectral}} = \sup_{\boldsymbol{x}\in\mathbb{R}^d:\|\boldsymbol{x}\|_2\leq 1} \|\boldsymbol{A}\boldsymbol{x}\|_2.$$

1.2.2 Optimal bounds via Subset Sum Problem

By proving Theorem 1.2.1, Malach et al. (2020) obtained the first polynomial bound on the over-parameterization required to ensure the SLTH. As is common for first results, the bound obtained is not optimal and some of the steps may be somewhat overzealous. Mainly, when controlling the gadget illustrated in figure 1.2 to approximate a target weight, we can be a bit less restrictive.

Let us once again assume the activation function σ to be the identity to simplify things. Later, we can handle the actual case of $\sigma = \mathrm{relu}$ by leveraging identity equation (1.3) once more. As figure 1.4 summarises, we wish to have

$$\begin{aligned} wx &\approx y \\ &= \sum_{i\in[n]} \mathrm{V}_i \mathrm{U}_i x. \end{aligned}$$

Setting $\mathbf{T} = \mathbf{V} \odot \mathbf{U}$, we have that

$$\begin{aligned} \sum_{i\in[n]} \mathrm{V}_i \mathrm{U}_i x &= \sum_{i\in[n]} \mathrm{T}_i x \\ &= \left(\sum_{i\in[n]} \mathrm{T}_i\right) \cdot x. \end{aligned}$$

Moreover, by pruning $\mathbf{U}$ or $\mathbf{V}$ we can fully control which entries of $\mathbf{T}$ are set to zero. That is, altogether, our goal is to find a subset $\mathcal{S} \subseteq [n]$ for which

$$w \approx \sum_{i \in \mathcal{S}} \mathrm{T}_i.$$

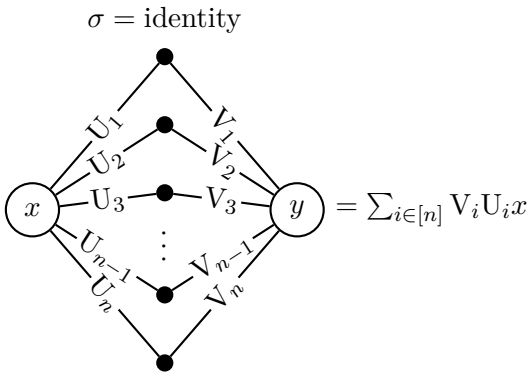


Figure 1.4: Random gadget for figure 1.2 with identity as activation function.

Orseau, Hutter, and Rivasplata (2020) takes advantage of this observation to obtain a better bound than Theorem 1.2.1. Namely, instead of equation (1.1), they show that

$$n_i \geq C d_i \log \frac{d_i \ell}{\varepsilon}, \tag{$*$}$$

where C is a universal constant different from that in Theorem 1.2.1 and we omitted the term on the confidence δ for simplicity. The main strategy behind this improvement is to select entries T_i that approximate different powers of two, i.e., 2^{-s} for $s \in \{1, 2, \ldots, \lceil \log 1/\varepsilon \rceil\}$. In this way, given any w we can approximate it by considering its binary representation and further pruning T to leave only entries corresponding to the "on" bits.[8]To be exact, the authors obtain the bound in equation ($*$) using a "goldary" representation instead of a binary one: they employ a decomposition in base $1/\phi$ where $\phi = (1 + \sqrt{5})/2 \approx 1.62$ is the golden ratio. Since the values associated with most bits are quite small, the authors need the distribution of the random weights to be concentrated around zero, so they assume each parameter of the random network to follow a hyperbolic distribution.

The concurrent work Pensia, Rajput, Nagle, Vishwakarma, and Papailiopoulos (2020), however, achieved essentially optimal bounds by recognising the setup as a randomized instance of the classical *Subset Sum Problem (SSP)*. In the SSP, one is given as input a set of n integers $\{x_1, x_2, \ldots, x_n\}$ and a target value z, and wishes to decide if there exists a subset of $\mathcal{S}$ that sums to z. That is, one is to reason about a subset $\mathcal{S} \subseteq [n]$ such that $\sum_{i \in \mathcal{S}} x_i = z$.

This problem is central to complexity theory, figuring among Garey and Johnson's six basic NP-hard problems (Garey & Johnson, 1979),[9] and has found a wide range of applications in computer science and beyond, which we summarise in section 2.1. Among the vast literature on the subject, when studying the expected optimum of a randomised version of the SSP, Lueker (1998) proved that

[8]Going back to an analogy from the first pages of the thesis, we could say the authors look for "LEGO® bricks" in the sample.

[9]Albeit as a slight variation. Section 2.1 details this relationship.

> [Informal version of Theorem 2.1.1] For a wide class of distributions, given a set of n independent random variables $\mathcal{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_n\}$, it suffices for n to be of the order of $\log 1/\varepsilon$ to ensure, with high probability, that for every $z \in [-1, 1]$ there exists a subset of $\mathcal{X}$ whose sum approximates z up to error ε.

This beautiful result fits our setup like a glove. To see its asymptotic optimality, notice that there are 2^n subsets of $\mathcal{X}$. Each subset corresponds to a sum (of its elements). Intersections apart, each sum can serve as a suitable approximation for all the values in an interval of radius ε around the sum. Since covering $[-1, 1]$ with intervals of radius ε takes at least $\lceil 2/2\varepsilon \rceil$ of them, we must have

$$2^n \geq \frac{1}{\varepsilon},$$

that is

$$n \geq \log \frac{1}{\varepsilon}.$$

Pensia et al. (2020) leveraged this optimality and other ideas to improve on Theorem 1.2.1, obtaining the same thesis while, instead of equation (1.1), only requiring that

$$n_i \geq C \log \frac{d_i \ell}{\min\{\varepsilon, \delta\}}, \tag{1.4}$$

again, for a different universal constant C. The authors were also able to lift one of the hypotheses on the parameters of the target network. For $i \in [\ell]$, their result assumes only that $\|\boldsymbol{W}_i\|_{\text{spectral}} \leq 1$ while Theorem 1.2.1 also needs $\|\boldsymbol{W}_i\|_\infty \leq 1/\sqrt{d_i}$. Finally, they carry the generality on the choice of distribution from Lueker's theorem. Namely, the weights of the random network may follow any law that "contains some scale of the uniform": a distribution whose probability density function φ satisfies $\varphi(x) \geq \alpha$ for all $x \in [-\beta, \beta]$, where α and β can be any positive constants.

Pensia et al. (2020) also showed that the bound in equation (1.4) is essentially optimal. To be more precise, let $\boldsymbol{\mathcal{F}}$ be the family of linear networks whose spectral norm is at most one, i.e.,

$$\boldsymbol{\mathcal{F}} = \Big\{ f(\,\cdot\,; \boldsymbol{W}) \colon [-1, 1]^d \to \mathbb{R}^d, f(\boldsymbol{x}; \boldsymbol{W}) = \boldsymbol{W}\boldsymbol{x} \;\Big|\; \|\boldsymbol{W}\|_{\text{spectral}} \leq 1 \Big\}.$$

The authors proved that given a random network G, regardless of the distribution of the weights, if $\mathfrak{Prune}(G)$ contains an approximation of each function in $\boldsymbol{\mathcal{F}}$ up to error ε (as in equation (1.2)), then G must have at least $d^2 \log 1/\varepsilon$ parameters. In particular, if G is a 2-layer network, its width must be at least $d \log 1/\varepsilon$. That is, G has to be "over-parameterized" by at least a factor of $\log 1/\varepsilon$. The argument underlying this lower bound is essentially the one we provided above for the optimality of Lueker's result.

Lastly, we remark that both Orseau et al. (2020) and Pensia et al. (2020) require the random network to be twice as deep as the target architecture.

1.3 Our contributions

With the technical context put in place in section 1.2, we now outline our contributions.

1.3.1 Simplified analysis of the Subset Sum Problem

In this section, we overview our alternative, simpler proof of Theorem 2.1.1.

1.3.1.1 Motivation

We saw in section 1.2 that a result by Lueker on the SSP became a powerful tool in the context of the SLTH. Its "rediscovery" by Pensia et al. (2020) was sure to spark a renewed interest in it, as the theorem is not only strong but also has a simplicity that makes it remarkably easy to understand and use.

This elegance turns out to be inherited from the SSP itself. Even though it is one of the most classic NP-complete problems (Garey & Johnson, 1979), it is also known to lead to simple analyses under many techniques, even the most intricate ones (Mertens, 2001). The problem has a neat recursive structure that makes it easy to approach via dynamic programming. To illustrate it, consider a set of values $x_1, \dots, x_n$. Then, we recursively define sets $\mathcal{A}_t$ to keep track of all possible subset-sums using the first i numbers. Assuming that summing all elements in the empty subset amounts to zero, we start with $\mathcal{A}_0 = \{0\}$. From there, we define

$$\mathcal{A}_1 = \mathcal{A}_0 \cup \{x_1\}$$
$$\mathcal{A}_2 = \mathcal{A}_1 \cup \{x_2, x_1 + x_2\}$$

and, more generally,

$$\mathcal{A}_{t+1} = \mathcal{A}_t \cup \{a + x_{t+1} \mid a \in \mathcal{A}_t\},$$

for $t \in [n-1]$. We can leverage this simple strategy to solve the SSP in polynomial time if ε is fixed (Bellman, 1966).

The union of a tendency towards elegance and inherent algorithmic complexity motivated Brian Hayes to name the SSP "the easiest hard problem" (Hayes, 2002).[10] This property gives the problem an important didactic role.

Upon exploring the original proof of Lueker's result, we realised that the inherent simplicity of the SSP does not hold on for long. Lueker (1998) approaches Theorem 2.1.1 through the underlying recursiveness that we touched above. The author uses it to keep track of random variables V_t associated with the proportion of the values in the interval $[-1, 1]$ that can be approximated by the sum of some subset of the first t variables, $\mathrm{X}_1, \dots, \mathrm{X}_t$.

However, this intuitive direction is quickly obscured by the need to tame the stochastic dependencies of the process. Lueker (1998) does so by employing tools from martingale theory, which only becomes possible after a non-linear transformation of V_t.[11] This not only hinders any intuition on the obtained martingale but also forces the argument into a somewhat cluttering case analysis.

1.3.1.2 Our main result

Motivated by the elegance of the problem, in chapter 2 we present a simplified proof of Theorem 2.1.1. We start in the same direction as the original argument, tracking the mass of values with suitable approximations as we reveal the values of the random variables $\mathrm{X}_1, \dots, \mathrm{X}_n$ one by one. However, we employ a random variable that maps more directly to the intuitive recursion of the SSP, while Lueker (1998) requires some modifications to it. We proceed to directly analyse this variable, without any transformations.

[10] More precisely, the variant where the value to be approximated is exactly half of the sum of all x_is. This version of the SSP is called the *Number Partition Problem*.

[11] The exact function is $\psi(x) = \log x - \ln(1-x) + x/2$.

As it is common in rumour spreading contexts (Doerr & Kostrygin, 2017), this analysis reveals two expected behaviours: as we consider the first variables, the proportion of approximated values grows very fast; then, after a certain point, the proportion of non-approximable values decreases very fast. The rumour spreading framework allows us to deal with the stochastic dependencies of the process with classical tools, such as Markov's inequality and Hoeffding's bounds. Ultimately, this results in a substantially more elementary proof that is also more direct and attains itself to the intuition of the problem.

We present the full discussion in chapter 2. The chapter is based on our work da Cunha, d'Amore, et al. (2022), which was accepted at *the Thirtieth European Symposium on Algorithms (ESA)*, track S (dedicated to simplifications of existing results).

1.3.2 Generalisation of the SLTH to Convolutional Neural Networks

In this section, we overview our generalisation of the results on the SLTH by Malach et al. (2020); Orseau et al. (2020); Pensia et al. (2020) to *Convolutional Neural Networks (CNNs)*. To this end, we need to extend our notation.

1.3.2.1 Notation

We denote slices of tensors by indexing them with colons. For example, the expression $\boldsymbol{X}_{:,:,i}$ represents a 2-D slice of a 3-D tensor. We refer to the axis of 4-D tensors as *rows*, *columns*, *channels*, and *filters*, in this order.[12] We reserve the term *kernel* to address entire 4-D tensors (a vector of filters).

We only consider explicitly 2-dimensional convolutions with multiple channels, multiple kernels and enough zero-padding to preserve the output shape. However, as we discuss in section 4.2.1, our results can be generalised to many other variants.

Definition 1.3.1 (Convolution)**.** Given a filter $\boldsymbol{K} \in \mathbb{R}^{d\times d\times c}$ and an input tensor $\boldsymbol{X} \in \mathbb{R}^{D\times D\times c}$, the 2-dimensional discrete convolution between $\boldsymbol{K}$ and $\boldsymbol{X}$ is the $D \times D$ matrix with entries given by

$$(\boldsymbol{K} * \boldsymbol{X})_{i,j} = \sum_{i',j'\in[d],k\in[c]} K_{i',j',k} \cdot X_{i-i'+1,j-j'+1,k} \quad \text{for } i,j \in [D],$$

where $\boldsymbol{X}$ is suitably zero-padded so that the width and height of the output match those of the input. As is usual for CNNs, this is not the case for the depth (number of channels) and, since we assume the depth of the input and the kernel to be the same (c), this implies that the output has a single channel.

For a kernel $\boldsymbol{K} \in \mathbb{R}^{d\times d\times c_0\times c_1}$ we perform the convolution with each of the c_1 filters independently and stack the results along the channel axis, obtaining a $D \times D \times c_1$ tensor. Hence, using tensor slices, we can define $\boldsymbol{K} * \boldsymbol{X}$ as the $D \times D \times c_1$ tensor such that

$$(\boldsymbol{K} * \boldsymbol{X})_{:,:,\ell} = \boldsymbol{K}_{:,:,:,\ell} * \boldsymbol{X} \quad \text{for } \ell \in [c_1].$$

Alternatively, $\boldsymbol{K} * \boldsymbol{X}$ is the tensor with entries given by

$$(\boldsymbol{K} * \boldsymbol{X})_{i,j,\ell} = \sum_{i',j'\in[d],k\in[c_0]} K_{i',j',k,\ell} \cdot X_{i-i'+1,j-j'+1,k} \quad \text{for } i,j \in [D], \ell \in [c_1].$$

[12] It is worth mentioning that Goodfellow et al. (2016) uses a different ordering since most of our notation comes from that reference.

A CNN is a neural network that uses convolutions (instead of the usual matrix multiplications) in at least one of its layers.

Finally, to take advantage of the full generality of Lueker's result in our statements, given $\alpha, \beta > 0$, let us define $\boldsymbol{\mathcal{P}}_{\alpha,\beta}$ to be the set of all probability distributions P over $\mathbb{R}$ such that

$$\varphi_P(x) \geq \alpha, \text{ for all } x \in [-\beta, \beta],$$

where φ_P is the probability density function of P.

1.3.2.2 Motivation

All empirical works leading to the SLTH we mentioned so far performed their experiments with CNNs, including the experiments of Frankle and Carbin (2019), which inaugurate the original LTH. There is good reason for that.

The convolution operation is a generalisation of the matrix multiplication underlying fully-connected layers, with convolutional layers being a regularised version of fully-connected ones. The first form of regularisation is structured sparsity: instead of connecting each output neuron to all input neurons, convolutional layers enforce a degree of spatial locality by only connecting to clusters of input neurons (usually small square regions). This structure per se already makes the network especially suited for processing visual data, as demonstrated by applications of Locally Connected Networks (LCNs) (Gregor & LeCun, 2010; Huang, Lee, & Learned-Miller, 2012; Taigman, Yang, Ranzato, & Wolf, 2014; Y. Sun, Liang, Wang, & Tang, 2015; Grönquist et al., 2021). The second form of regularisation inherent to convolutional layers is parameter sharing: unlike in LCNs, CNNs use the same set of weights for the connections to each input cluster. Compared to fully-connected networks, or even to LCNs, the weight sharing dramatically reduces the number of parameters in CNNs and makes it independent of the input size, which can be very large for many applications. This makes CNNs more robust to overfitting and much more efficient to train and evaluate. Those properties made CNNs central to the Renaissance of Deep Learning in the 2010s, starting with AlexNet (Krizhevsky et al., 2012) and dominating the field until very recently.

Correspondingly, all of the works we reviewed in section 1.2 suggest a generalisation of their results to CNNs as a natural next step. This may be somewhat unexpected, since the convolution is linear and, thus, can be encoded as a matrix multiplication. In fact, most implementations of convolutional layers in DL frameworks use matrix multiplications under the hood, a technique called *im2col* (Chellapilla, Puri, & Simard, 2006). However, to cover the weight-sharing property of the convolution this conversion usually requires a large number of redundant parameters. Hence, applying the result by Pensia et al. (2020) without taking into account the convolutional structure would lead to a polynomial bound instead of a logarithmic one.

On the other hand, to see a matrix-vector multiplication as a convolution, it suffices to consider each row of the matrix as a filter and, perhaps, reshape the tensors involved, depending on the convention used. Crucially, this is only a reduction to a special case and adds no overhead whatsoever.

1.3.2.3 Our main result

Now we overview our generalisation of the results by Malach et al. (2020), Orseau et al. (2020), and Pensia et al. (2020) to CNNs within logarithmic bounds on the over-parameterization.

For the convenience of the reader, we also state our main result below.

Theorem (4.2.3)**.** *Let $D, c_0, \ell \in \mathbb{N}$, and $\varepsilon, \delta, \alpha, \beta \in \mathbb{R}_{>0}$. For $i \in [\ell]$, let $d_i, c_i, n_i \in \mathbb{N}$. Let $\boldsymbol{\mathcal{F}}$ be the class of functions from $[0,1]^{D\times D\times c_0}$ to $\mathbb{R}^{D\times D\times c_\ell}$ such that, for each $f \in \boldsymbol{\mathcal{F}}$*

$$f(\boldsymbol{X}) = \boldsymbol{K}^{(\ell)} * \mathrm{relu}(\boldsymbol{K}^{(\ell-1)} * \cdots \mathrm{relu}(\boldsymbol{K}^{(1)} * \boldsymbol{X})),$$

where, for $i \in [\ell]$, $\boldsymbol{K}^{(i)} \in [-1,1]^{d_i\times d_i\times c_{i-1}\times c_i}$ and $\|\boldsymbol{K}^{(i)}\|_1 \leq 1$.

Finally, let $G\colon [0,1]^{D\times D\times c_0} \to \mathbb{R}^{D\times D\times c_\ell}$ be a 2ℓ-layered random CNN given by

$$G(\boldsymbol{X}) = \boldsymbol{\mathsf{L}}^{(2\ell)} * \mathrm{relu}(\boldsymbol{\mathsf{L}}^{(2\ell-1)} * \cdots \mathrm{relu}(\boldsymbol{\mathsf{L}}^{(1)} * \boldsymbol{X}))$$

where the parameters of G are i.i.d. random variables following a distribution $P \in \boldsymbol{\mathcal{P}}_{\alpha,\beta}$ and for $i \in [\ell]$ the kernels $\boldsymbol{\mathsf{L}}^{(2i-1)}$ and $\boldsymbol{\mathsf{L}}^{(2i)}$ have shape $d_i \times d_i \times c_{i-1} \times c_i n_i$ and $1 \times 1 \times c_i n_i \times c_i$, respectively, so that n_i is an integer over-parameterization factor.

Then, there exists $C_{\alpha,\beta} > 0$, such that if, for $i \in [\ell]$,

$$n_i \geq C_{\alpha,\beta} \log \frac{c_{i-1} c_i d_i^2 \ell}{\min\{\varepsilon, \delta\}}, \tag{1.5}$$

then, with probability at least $1-\delta$, the following holds for all $f \in \boldsymbol{\mathcal{F}}$:

$$\sup_{\boldsymbol{X}\in[0,1]^{D\times D\times c_0}} \min_{g\in\mathfrak{Prune}(G)} \|f(\boldsymbol{x}) - g(\boldsymbol{x})\|_\infty < \varepsilon,$$

where $\mathfrak{Prune}(G)$ is the set of all networks that can be obtained by pruning G.

The theorem above bares a deep analogy with Theorem 1.2.1 and even more so with the result by Pensia et al. (2020), given the logarithmic bound in equation (1.5). Accordingly, much of our discussion from section 1.2 applies here as well. To minimise repetition, we use this opportunity to assume a flipped perspective in our discussion this time. Instead of reasoning in terms of the over-parameterization required to approximate a target architecture, we can see Theorem 4.2.3 as a statement about the expressiveness of random CNNs.

Under this light, Theorem 4.2.3 establishes a lower bound on how much a "train by pruning" algorithm such as EDGE-POPUP can, in principle, achieve. Given suitable hypotheses, the result ensures that starting from a random CNN G, with high probability, it is possible to carve it to reveal sub-CNNs g that approximate any sufficiently smaller CNN f. More precisely, any f with half the number of layers of G and a channel/filter count that is smaller by a logarithmic factor. In particular, given a specific task, by pruning G we can obtain a g that approximates a CNN f with the best performance among all the networks in $\boldsymbol{\mathcal{F}}$. Therefore, we want the architecture associated with $\boldsymbol{\mathcal{F}}$ to be as expressive as possible, so that the best f is as good as possible. This is the reason to seek small bounds in equation (1.5).

Remarkably, the properties of convolutions allow for a bound that is independent of the input height and width D, which tend to be large in practice. There is, however, a weak dependency on the number of channels of the input c_0. This is unavoidable since we adopted the convention of having the depth of the filters match the depth of the input. Nonetheless, this dependency should cause little concern as it only applies to the first layer and the depth of the input tends to be small in applications.

Regarding the hypotheses of our result, the assumption that the kernel of every second layer has shape $1 \times 1 \times \dots$ may seem restrictive, but it is a mere artefact of the proof. We chose to keep it explicit in the statement for added generality since one can readily prune entries of an arbitrarily shaped tensor to enforce the desired shape.[13]

Moreover, while previous results restrict the parameters in terms of the spectral norm of the weight matrices, we employ the 1-norm of tensors. This choice is tied to our use of Young's Convolution Inequality to control the propagation of error through convolutions. We prove the exact instance of the inequality required for our argument in appendix F. As we discussed in Remark 1.2.1, the positive-homogeneity of ReLU networks demands a constraint on the norm of the weights and of the input. The hypothesis of non-negativity of the input, however, is not necessary. As we will bring in section 4.2.1, since the output of the ReLU activations is non-negative, this restriction is only pertinent to the first layer of the network. Regardless, as noted by the subsequent work Burkholz (2022a), one can easily adapt our original argument to allow for negative inputs. It suffices to place the kernels with shape $1 \times 1 \times \dots$ before the ones with shape $d_i \times d_i \times \dots$ rather than after.

Finally, we note that with little modification, if any, it is possible to adapt our result to support other types of convolutional layers, such as those with different values of stride, padding, or dilation, as well as other operations that can be reduced to convolutions, e.g., average pooling.

Chapter 4 provides a full discussion of the proof of Theorem 4.2.3, including sketches for the arguments used to obtain each result (see section 4.2). That chapter is based on our work (da Cunha, Natale, & Viennot, 2022), which was published in the proceedings of *the Tenth International Conference on Learning Representations, (ICLR)*.

1.3.3 Subset-Sum Problem in multiple dimensions

In this section, we overview our work on a multidimensional analogous of Theorem 2.1.1. Our discussion of the SSP so far should, ideally, persuade the reader that the importance of the problem alone justifies any natural generalisation of it. Still, our overall history leads quite seamlessly to some extra motivation.

1.3.3.1 Motivation

So far, we have only discussed pruning neural networks with complete freedom to remove any weight, which is referred to as *unstructured pruning*. However, this type of pruning usually leads to unstructured sparsity, which can be difficult to leverage in practice, as we will discuss further in section 1.3.4 and its associated chapter 5. In short, common hardware is best suited to operate on dense, regular patterns of data. This is also true on the software side, with DL frameworks often supporting sparse computations to a lower degree if compared to dense operations.

Fortunately, can effectively overcome these issues by constraining the pruning to follow specific patterns. This is referred to as *structured pruning*. In the extreme, we can only consider pruning whole neurons or even entire layers of the network. We can leverage both strategies to obtain a smaller dense version of the original network rather than a sparse one of the same size. Figure 1.5 illustrates this idea for the case of neuron pruning.

[13] In reality, our argument only needs the filters to have at most one non-zero entry per channel.

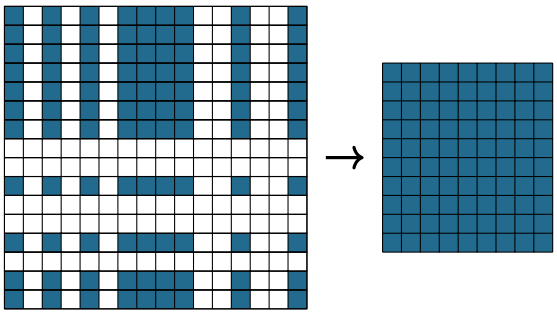


Figure 1.5: On the left, we illustrate the effect of structured pruning of neurons in a weight matrix of a fully-connected layer. The rows in white correspond to neurons pruned in this layer while the columns in white are the effect of removing neurons from the previous layers. On the right, we allude to the possibility of collapsing the pruned matrix into a smaller, dense one.

For those types of pruning, we end up with a network that not only has fewer parameters but also a dense layout that can be leveraged as usual by hardware and software. This makes it trivial to convert pruning into actual computational savings.

Therefore, it is only natural to consider a variant of the SLTH that handles structured pruning. Unfortunately, however, our main tool to work with the SLTH, Theorem 2.1.1, can only handle effectively individual random variables. That is, applying it to random vectors leads to exponential bounds.

1.3.3.2 Our main results

For the convenience of the reader, we also state the main result of chapter 3 below.

Theorem (3.1.2)**.** *Given $\varepsilon \in (0,1)$ and $d, n \in \mathbb{N}$, consider n independent d-dimensional standard normal random vectors $\mathbf{X}_1, \dots, \mathbf{X}_n$. There exists a universal constant $C > 0$ for which, if*

$$n \geq Cd^3 \log \frac{1}{\varepsilon} \cdot \log \frac{d}{\varepsilon},$$

then, with high probability, for all $\boldsymbol{z} \in [-1,1]^d$ there exists a subset $\mathcal{S}_{\boldsymbol{z}} \subseteq [n]$ for which

$$\left\| \boldsymbol{z} - \sum_{i \in \mathcal{S}_{\boldsymbol{z}}} \mathbf{X}_i \right\|_\infty \leq 2\varepsilon.$$

Moreover, the approximations can be achieved with subsets of size $\frac{n}{6\sqrt{d}}$.

This result is quite similar to Theorem 2.1.1. The replacement of the uniform distribution with a normal serves mainly to simplify the analysis, while the worse bounds on the sample size, as well as the note on the size of the subsets,[14] are by-products of our strategy.

We can infer a lower bound on the sample size required to achieve a given accuracy through a counting argument analogous to the one we provided for the single-dimensional case in section 1.2.2. Namely, in d dimensions, covering the set $[-1,1]^d$ requires at least $2^{d \log 1/\varepsilon}$ hypercubes of radius ε. Thus, since there are 2^n possible subsets of sample of n vectors, we need that $n = \Omega(d \log 1/\varepsilon)$ to be able to approximate every vector in $[-1,1]^d$.

[14] Notice that Theorem 2.1.1 does not precise how many elements are summed to obtain the approximations. In particular, results on the SLTH that leverage Theorem 2.1.1 cannot provide the exact sparsity of the resulting subnetworks.

Conversely, in expectation, having $n = \mathcal{O}(d \log 1/\varepsilon)$ is sufficient even if we only consider subsets of size $n/2$. There are $\binom{n}{n/2} \approx 2^{n-o(n)}$ such subsets, each summing to a random vector distributed as $\mathcal{N}(\mathbf{0}, \frac{n}{2} \cdot \boldsymbol{I}_d)$. Hence, given any $\boldsymbol{z} \in [-1, 1]^d$, each of those sums has probability approximately $\varepsilon^d (n/2)^{-\frac{d}{2}} = 2^{-d \log \frac{1}{\varepsilon} - \frac{d}{2} \log \frac{n}{2}}$ of being at most ε far from $\boldsymbol{z}$. We can then conclude that the expected number of approximations is $2^{n-o(n)} \cdot 2^{-d \log \frac{1}{\varepsilon} - \frac{d}{2} \log \frac{n}{2}}$, which is still of the order of $2^{n-o(n)}$ provided that $n \geq Cd \log 1/\varepsilon$ for a sufficiently large constant C.

Therefore, the multidimensional version of the problem brings us to the same general challenge as the single-dimensional one: to provide sufficiently strong concentration bounds while handling the stochastic dependency between subsets of the sample. Unfortunately, the approaches used to obtain Theorem 2.1.1—both the original (Lueker, 1998) and the one we propose in chapter 2—lead to exponential bounds if extended to multiple dimensions in any way we could envision.

Our argument goes in a different direction. We employ a second-moment approach and control dependencies by restricting the analysis to a family of subsets with sufficiently small pairwise intersections. We then proceed to carefully bound the contribution of these constrained intersections to the second moment of our variables of interest.

Finally, we illustrate the applicability of our result by considering the *Neural Network Evolution (NNE)* model recently introduced by Gorantla, Louis, Papadimitriou, Vempala, and Yadati (2019). It is natural to wonder whether their model is *universal*, in the sense that, with high probability, it can approximate any dense feed-forward neural network. While applying Theorem 2.1.1 to this end would yield exponential bounds on the required over-parameterization, in section 3.6 we use Theorem 3.1.2 to prove the universality of the NNE model within polynomial bounds.

Chapter 3 provides a full discussion of the results overviewed in this section. That chapter is based on our work (Becchetti et al., 2022).

1.3.4 Extension of the SLTH to structured pruning

In this section, we overview our proof of our extension of the SLTH to structured pruning.

1.3.4.1 Motivation

Wilkinson and Reinsch (1971) defines a sparse matrix as one that has enough zeros to make it worth taking advantage of them. This pragmatic definition alludes to the fact that leveraging sparse computations is not as straightforward as it might seem, as we previewed in section.

Regarding memory, saving the space otherwise occupied by zeros comes with the cost of storing the location of the remaining entries. Moreover, there are many different ways to represent sparse structures, each with trade-offs and overheads of its own (Pooch & Nieder, 1973). If the degree of sparsity is not high enough, those costs may not pay off.

As for the computational side, the main issue is that current commodity hardware is optimised for dense operations. For instance, experiments with GPU implementations by Han et al. (2017) found that even at almost 90% sparsity, their pruned networks were slower than the original dense ones. J. Yu et al. (2017) experimented extensively with sparse neural networks on both CPUs and GPUs. They found that in most configurations they tried pruning worsens performance despite an average sparsity of 80%. In particular, their CPU implementation of AlexNet ran 25% slower when 89% of the parameters were pruned.

Discussing the multiple reasons for hardware to have trended towards dense operations is beyond the scope of this thesis. However, for the sake of the next section, we remark that this tendency traces back to the fact that memory access is the main bottleneck in modern hardware. Moving memory takes orders of magnitude more time and energy than performing actual logical instructions. For example, in simpler architectures that cannot hide the latency of memory accesses, such as microcontrollers, pruning consistently improves the performance of neural networks (J. Yu et al., 2017).

The underwhelming effectiveness of general neural network pruning in practice is also due to software limitations. Sparse computations are historically targeted at scientific computing applications, where they are quite common (Davis & Hu, 2011). The levels of sparsity in those applications are typically higher than 99.9%, while network pruning usually achieves 50% to 99% (Gale, Zaharia, Young, & Elsen, 2020). Since most classical libraries for sparse computations are designed for scientific computing, implementing sparse neural networks with tools such as *Sparse BLAS* (Duff, Heroux, & Pozo, 2002) or *cuSPARSE* (Naumov, Chien, Vandermersch, & Kapasi, 2010) generally leads to weak performance.

Taking into account the above considerations, it seems fair to say that sparse neural networks are yet to win the *Hardware Lottery*: the phenomenon that some research ideas fail (thrive) not necessarily because of their inherent flaws (virtues), but because they happen to align poorly (well) with the hardware and software of the time (Hooker, 2021).

This situation might be changing, on both the hardware and software sides Nonetheless, until then, one alternative is to prune large structures of the network, such as neurons or entire layers. As we outlined in the previous section, this leads to smaller dense networks, which can directly run on the same hardware as the original ones without penalties. This approach works well in many cases, especially for CNNs when considering the pruning of entire filters (Kuzmin et al., 2019). However, constraining pruning to such coarse-grained structures severely limits the space of possible pruned networks and may prevent us from removing many parameters before performance degrades too much.

Alternatively, one can always win the (hardware) lottery by "cheating". While in the next section we will touch on the idea of designing dedicated hardware, in the context of pruning we can also "cheat" by removing structures in patterns tailored to specific devices. As it turns out, weaker structural constraints such as strided sparsity (Anwar, Hwang, & Sung, 2017) (figure 5.1b) or block sparsity (Siswanto, 2021) (figure 5.1b) are already sufficient to deliver the bulk of the computational gains that structured sparsity can offer, so long as the patterns are chosen to match the hardware. For example, Elsen, Dukhan, Gale, and Simonyan (2020) decides to prune blocks of size 16 in order to match the 16-wide L1 cache (and SIMD units) of the ARM CPUs they target.

1.3.4.2 Our main result

Despite representing the whole counterpart of unstructured pruning, to the best of our knowledge, there have been no previous results on structured pruning in the context of the SLTH. In chapter 5, we leverage the techniques developed in chapter 3 to fill this gap.

Before stating the main result of chapter 5 again for the convenience of the reader, we need to introduce versions of our notation $\mathfrak{Prune}$ for the types of structured pruning we consider.

Given $\boldsymbol{X} \in \mathbb{R}^{h \times w \times c \times f}$, we denote by $\mathfrak{FilterPrune}(\boldsymbol{X})$ the set of tensors obtained by zeroing some of the filters of $\boldsymbol{X}$. That is, setting to zero all entries of $\boldsymbol{X}_{:,:,:,i}$ for all i in some $\mathcal{S} \in 2^{[f]}$. Similarly, given a CNN f, we use the notation $\mathfrak{FilterPrune}(f)$ to denote the set of CNNs obtained

by zeroing some of the filters of some of the kernels of f. We remark that pruning a filter of a convolutional layer makes the corresponding channel of the following layer futile, so it can also be pruned without affecting the output of the network. The effect of neuron pruning we depicted in figure 1.5 is an instance of this. Also, as in the figure, we can collapse the resulting CNN into a smaller one by completely removing the pruned filters/channels.

Given a positive integer n, a tensor $\boldsymbol{X} \in \mathbb{R}^{h\times w\times c\times cn}$ is called n-channel-blocked if and only if

$$\boldsymbol{X}_{i,j,k,l} = \begin{cases} 1 & \text{if } \left\lceil \frac{l}{n} \right\rceil = k, \\ 0 & \text{otherwise,} \end{cases}$$

for all $i, j \in [d]$, $k \in [c]$, and $l \in [cn]$.

Theorem (5.3.1)**.** *Let $D, c_0, \ell \in \mathbb{N}$, and $\varepsilon \in \mathbb{R}_{>0}$. For $i \in [\ell]$, let $d_i, c_i, n_i \in \mathbb{N}$. Let $\boldsymbol{\mathcal{F}}$ be the class of functions from $[-1,1]^{D\times D\times c_0}$ to $\mathbb{R}^{D\times D\times c_\ell}$ such that, for each $f \in \boldsymbol{\mathcal{F}}$*

$$f(\boldsymbol{X}) = \boldsymbol{K}^{(\ell)} * \operatorname{relu}(\boldsymbol{K}^{(\ell-1)} * \cdots \operatorname{relu}(\boldsymbol{K}^{(1)} * \boldsymbol{X})),$$

where, for $i \in [\ell]$, $\boldsymbol{K}^{(i)} \in \mathbb{R}^{d_i\times d_i\times c_{i-1}\times c_i}$ and $\|\boldsymbol{K}^{(i)}\|_1 \leq 1$.

Let also $H\colon [-1,1]^{D\times D\times c_0} \to \mathbb{R}^{D\times D\times c_\ell}$ be a 2ℓ-layered random CNN given by

$$H(\boldsymbol{X}) = \mathbf{L}^{(2\ell)} * \operatorname{relu}(\mathbf{L}^{(2\ell-1)} * \cdots \operatorname{relu}(\mathbf{L}^{(1)} * \boldsymbol{X}))$$

where the parameters of H are i.i.d. random variables following a standard normal distribution and for $i \in [\ell]$ the kernels $\mathbf{L}^{(2i-1)}$ and $\mathbf{L}^{(2i)}$ have shape $1 \times 1 \times c_{i-1} \times 2n_ic_{i-1}$ and $d_i \times d_i \times 2n_ic_{i-1} \times c_i$, respectively.

Finally, let H_{block} be the random network obtained by pruning H just enough to make each $\mathbf{L}^{(2i-1)}$ n_i-channel-blocked, for $i \in [\ell]$.

Then, there exists a universal constant $C > 0$, such that if, for $i \in [\ell]$,

$$n_i \geq Cd^{13}c_i^6 \log^3 \frac{d^2c_ic_{i-1}\ell}{\varepsilon},$$

then, with probability at least $1-\varepsilon$, for all $f \in \boldsymbol{\mathcal{F}}$, we have that

$$\sup_{\boldsymbol{X}\in[-1,1]^{D\times D\times c_0}} \min_{h\in\mathfrak{Filter}\mathfrak{Prune}(H_{\text{block}})} \|f(\boldsymbol{x}) - g(\boldsymbol{x})\|_\infty < \varepsilon.$$

As expected, we prove the theorem above using a multidimensional variation of Theorem 2.1.1. We cannot apply Theorem 3.1.2 directly because the weight-sharing property of CNNs leads to stochastic dependencies between the random vectors while Theorem 3.1.2 assumes them to be independent. Yet, we use techniques similar techniques in our argument, as hinted by the normal distribution of the random parameters instead of the uniform one in Theorem 4.2.3.

In our proof, the pruning of filters takes place at layers $1, 3, \ldots, 2\ell - 1$. If we completely remove those filters and the corresponding channels in the next layer, the overall modification yields a CNN with kernels $\tilde{\mathbf{L}}^{(1)}, \ldots, \tilde{\mathbf{L}}^{(2\ell)}$ such that, for $i \in [\ell]$,

$$\operatorname{shape}(\tilde{\mathbf{L}}^{(2i-1)}) = 1 \times 1 \times c_{i-1} \times 2c_{i-1}m_i$$

and

$$\operatorname{shape}(\tilde{\mathbf{L}}^{(2i)}) = d_i \times d_i \times 2c_{i-1}m_i \times c_i,$$

where $m_i = \sqrt{n_i/(C_1 \log \frac{1}{\varepsilon})}$ for a universal constant C_1. As we alluded to in section 1.3.3.2, we can only specify the exact shape of the kernels because the strategy we employ to prove results on the multidimensional Subset Sum Problem involves subsets with a specific size.

Finally, we have also integrated into our statement the tactic of placing the kernels with shape $1 \times 1 \times \ldots$ before the ones with shape $d_i \times d_i \times \ldots$ to allow for negative inputs (Burkholz, 2022a).

Chapter 5 provides a full discussion of the results overviewed in this section. That chapter is based on our work (da Cunha, d'Amore, & Natale, 2023), which was submitted to this year's *Conference and Workshop on Neural Information Processing Systems (NeurIPS 2023)*.

1.3.5 Application to circuit design

In this section, we overview our proof of our extension of the SLTH to structured pruning.

1.3.5.1 Motivation: $\mathcal{O}(1)$ matrix multiplication

Multiplying two numbers in a digital computer involves a circuit with about a thousand transistors (an electrically controlled switch, for our purposes). Analogically, on the other hand, Ohm's law tells us that applying a voltage v across a resistance r induces a current of magnitude v/r to flow through it. Expressing it in terms of the inverse resistance, the conductance, we obtain the desired product vg, where $g = 1/r$.

The analogue computation happens faster and especially more efficiently than the mere switching of a single transistor of the digital circuit. Moreover, in conventional computers, recovering the value from memory to operate on it takes much more time and energy than performing the actual computation. Analogically, however, the memory is built into the computing circuit. All of this with a circuit based on a resistor, a component so simple that many times the challenge is to avoid creating them unintentionally. Moreover, resistors are particularly small, even when compared to transistors. Their exact area varies with the technology, but it is usually at least dozens of times smaller than that of the smallest transistors. The surface area of an integrated circuit is quite important since it is strongly correlated to the manufacturing cost, among other reasons.

Going one dimension up, we can accumulate the results of many parallel multiplications with an even simpler analogue principle. We replicate the multiplier circuit n times to have a vector of voltages $\boldsymbol{v} \in \mathbb{R}^n$ and another of conductances $\boldsymbol{g} \in \mathbb{R}^d$. Then, we simply connect their outputs. By Kirchhoff's current law, the total current leads to a total current of $\sum_{i=1}^{n} vg = \boldsymbol{v} \cdot \boldsymbol{g}$.

The next step is to distribute the n sources over n copies of the dot-product circuit. The final arrange, known as *resistive crossbar*, can be neatly packed into an $n \times n$ grid of conductances (see figure 6.2) with n outputs, $y_1, \ldots, y_n$. Referring to underlying the matrix of conductances as $\boldsymbol{G}$, this circuit performs the matrix-vector product $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x}$ in time $\mathcal{O}(1)$, approximately (Z. Sun & Huang, 2021).

Since almost all of the astronomical computational costs of deep learning lie in matrix multiplications, there is surely great potential in this analogue approach. It does not take much consideration to start noticing the challenges preventing it from becoming commonplace.

Initially, we note the conductances are static. They are manufactured with a fixed value and cannot be changed. Still, having a static circuit is interesting as many applications, especially for mobile and embedded systems, only execute a model that is trained beforehand elsewhere. However, the imprecisions of the manufacturing process always reflect on analogue circuits. Even

though large neural networks are remarkably robust to noise, with the current technology, fabricating the resistors to the required precision, if it is even possible, would require prohibitively expensive post-processing to calibrate each resistor.

In practice, actual implementations use alternative components as sources of conductance. The potential analogue computing in deep learning has inspired many different technologies, each with its own advantages and disadvantages. Discussing them fairly is beyond the scope of this work, so we refer the reader to the surveys (T. P. Xiao, Bennett, Feinberg, Agarwal, & Marinella, 2020) and (Chakraborty et al., 2020). What unifies them is that, in the search to implement programmable and precise sources of conductance, they all introduce new challenges and overheads that prevent implementations from harnessing most of the capabilities of the analogue computations.

Programmability seems quite essential in this context, though. Without it, we would be left with many tiny static units with random properties. At this point, the plot twist might have become apparent to the reader.

1.3.5.2 Our main result

We propose a method to combine standard, inaccurate resistors to obtain a precise and programmable source of conductance.

Starting with a set of resistors, we connect a transistor to each of them. We design the circuit with its equivalent conductance in mind. If all transistors are on, the total conductance is the sum of the individual conductances of the resistors. By controlling the transistors, we can select the subset of the resistors that will take part in the sum.

In practice, we can measure the conductance of each resistor once and store it. Then, given a target conductance, we can use the stored values to solve an instance of the SSP. It is possible to do so by using general optimisation software, but if the number of problems is very high, dynamic programming can be used. Theorem 2.1.1 ensures the number of resistors required scales logarithmically with the desired precision of the approximations. As for measuring the conductances in the first place, we can leverage the same circuitry that we already use for converting the outputs to the digital domain.

Naturally, implementing devices in the real world is a complex endeavour with many challenges. While we propose some strategies to overcome them in the associated chapter 6, an exhaustive list would quickly scape the scope of this thesis.

Finally, for simplicity, we described the ideas behind our approach in the context of resistors even though they apply to many other types of components. The same holds for the specific application. The use of transistors to select resistors is particularly troublesome as it would make the circuit much larger and, thus, much more costly. This issue can be avoided, for example, by using highly non-linear ReRAM devices instead (Luo et al., 2016; Midya et al., 2017). Even within the context of neuromorphic computing, there are many other possible applications. For instance, we could easily rearrange the circuit to have each resistor implement a weight one weight in that matrix. In this way, we could select weights via "train-by-pruning" strategies, ultimately replacing the SSP solver with Stochastic Gradient Descent. More generally, our contribution hopes to bring a new perspective to circuit design.

Chapter 5 provides a full discussion of the results overviewed in this section. That chapter is based on the content submitted for patenting (Da Cunha, Natale, & Viennot, 2022).

PART

Random Subset-Sum Problem

CHAPTER 2

Revisiting the Random Subset-Sum Problem

The average properties of the well-known Subset Sum Problem *can be studied by the means of its random version, where we are given a target value z, random variables $\mathrm{X}_1, \ldots, \mathrm{X}_n$, and an error parameter $\varepsilon > 0$, and we seek a subset of the X_is whose sum approximates z up to error ε. In this setup, it has been shown that, under mild assumptions on the distribution of the random variables, a sample of size $\mathcal{O}(\log(1/\varepsilon))$ suffices to obtain, with high probability, approximations for all values in $[-1/2, 1/2]$. Recently, this result has been rediscovered outside the algorithms community, enabling meaningful progress in other fields. In this chapter, we present an alternative proof for this theorem, with a more direct approach and resourcing to more elementary tools.*

2.1 Introduction

In the *Subset Sum Problem (SSP)*, one is given as input a set of n integers $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ and a target value z, and wishes to decide if there exists a subset of $\mathcal{X}$ that sums to z. That is, one is to reason about a subset $\mathcal{S} \subseteq [n]$ such that $\sum_{i \in \mathcal{S}} x_i = z$. The special case where z is half of the sum of X is known as the *Number Partition Problem (NPP)*. The converse reduction is also rather immediate.[1]

Be it in either of these forms, the SSP finds applications in a variety of fields, ranging from combinatorial number theory (Z.-W. Sun, 2003) to cryptography (Gemmell & Johnston, 2001; Kate & Goldberg, 2011). In complexity theory, the SSP is a well-known NP-complete problem, being a common base for NP-completeness proofs. In fact, the NPP version figures among Garey and Johnson's six basic NP-hard problems (Garey & Johnson, 1979). Under certain circumstances, the SSP can be challenging even for heuristics that perform well for many other NP-hard problems (Johnson, Aragon, McGeoch, & Schevon, 1991; Ruml, Ngo, Marks, & Shieber, 1996), and a variety of dedicated algorithms have been proposed to solve it (Helm & May, 2018; Bringmann & Wellnitz, 2021; Jin & Wu, 2019; Jin, Vyas, & Williams, 2021). Nonetheless, it is not hard to solve it in polynomial time if we restrict the input integers to a fixed range (Bellman, 1966). It suffices to recursively list all achievable sums using the first i integers: we start with $A_0 = \{0\}$ and compute A_{i+1} as $A_i \cup \{a + x_{i+1} \mid a \in A_i\}$. For integers in the range $[0, R]$, the search space has size $\mathcal{O}(nR)$.

Studying how the problem becomes hard as we consider larger ranges of integers (relative to n) requires a randomised version of the problem, the *Random Subset Sum Problem (RSSP)*, where the input values are taken as independently and identically distributed random variables. In this setup, Borgs, Chayes, and Pittel (2001) proved that the problem experiences a phase transition in its average complexity as the range of integers increases.

The result we approach in this chapter comes from related studies on the typical properties of the problem. In Lueker (1998) the author proves that, under fairly general conditions, the expected minimal distance between a subset sum and the target value is exponentially small. More specifically, they show the following result.

Theorem 2.1.1 (Lueker, 1998)**.** *Let* $\mathrm{X}_1, \ldots, \mathrm{X}_n$ *be independent uniform random variables over* $[-1, 1]$*, and let* $\varepsilon \in (0, 1/3)$*. There exists a universal constant* $C > 0$ *such that, if* $n \geq C \log(1/\varepsilon)$*, then, with probability at least* $1 - \varepsilon$*, for all* $z \in [-1, 1]$ *there exists* $\mathcal{S}_z \subseteq [n]$ *for which*

$$\left| z - \sum_{i \in \mathcal{S}_z} \mathrm{X}_i \right| \leq \varepsilon.$$

That is, a rather small number (of the order of $\log \frac{1}{\varepsilon}$) of random variables suffices to have a high probability of approximating not only a single target z, but all values in an interval.

Even though Theorem 2.1.1 is stated and proved for uniform random variables over $[-1, 1]$, it is not hard to extend the result to a wide class of distributions.[2] With this added generality, the theorem becomes a powerful tool for the analysis of random structures, and has recently proven to be particularly useful in the field of Machine Learning, taking part in a proof of the Strong Lottery

[1] To find a subset of $\mathcal{X}$ summing to z, one only needs to solve the NPP for the set $\mathrm{X} \cup \{2z, \sum_{i \in [n]} x_i\}$. By doing so, one of the parts must consist of the element $\sum_{i \in [n]} x_i$ alongside the desired subset.

[2] Distributions whose probability density function φ satisfies $\varphi(x) \geq b$ for all $x \in [-a, a]$, for some constants $a, b > 0$ (see Corollary 3.3 from Lueker (1998)).

Ticket Hypothesis (Pensia et al., 2020) and in subsequent related works (da Cunha, Natale, & Viennot, 2022; Fischer & Burkholz, 2021; Burkholz, Laha, Mukherjee, & Gotovos, 2022), and in Federated Learning (C. Wang et al., 2021).

Generalisations of the RSSP have played important roles in the study of random Knapsack problems (Beier & Vöcking, 2003, 2004), and to random binary integer programs (Borst, Dadush, Huiberts, & Tiwari, 2023; Borst, Dadush, Huiberts, & Kashaev, 2023). In particular, Becchetti et al. (2022), Borst, Dadush, Huiberts, and Kashaev (2023), and Borst, Dadush, Huiberts, and Tiwari (2023) recently provided an extension of Theorem 2.1.1 to multiple dimensions. As for the equivalent Random Number Partitioning Problem, Chen, Jin, Randolph, and Servedio (2022) recently generalised Borgs et al. (2001) and the integer version of the RSSP to non-binary integer coefficients.

The simplicity and ubiquity of the SSP have granted the related results a special didactic place. Be it as a first example of NP-complete problem (Garey & Johnson, 1979), a path to science communication (Hayes, 2002), or simply as a frame for the demonstration of advanced techniques (Mertens, 2001), it has been a tool to make important, but sometimes complicated, ideas easier to communicate.

This chapter offers a substantially simpler alternative to the original proof of Theorem 2.1.1 by following a general framework introduced in the context of the analysis of Rumour Spreading algorithms (Doerr & Kostrygin, 2017). Originally, Lueker (1998) approaches Theorem 2.1.1 by considering the random variable associated to the proportion of the values in the interval $[-1, 1]$ that can be approximated up to error ε by the sum of some subset of the first t variables, $\mathrm{X}_1, \ldots, \mathrm{X}_t$.

After restricting to some specific types of subsets, they proceed to evaluate the expected per-round growth of this proportion, conditioned on the outcomes of $\mathrm{X}_1, \ldots, \mathrm{X}_t$. Their strategy is to analyse this expected increase by martingale theory, which only becomes possible after a non-linear transformation of the variables of interest. Those operations hinder any intuition for the obtained martingale. Nonetheless, a subsequent application of the Azuma-Hoeffding bound (Azuma, 1967) followed by a case analysis leads to the result.

The argument presented here starts in the same direction as the original one, tracking the mass of values with suitable approximations as we reveal the values of the random variables $\mathrm{X}_1, \ldots, \mathrm{X}_n$ one by one. However, we quickly diverge from Lueker (1998), managing to obtain an estimation of the expected growth of this mass without discarding any subset-sum. We eventually restrict the argument to some types of subsets, but we do so at a point where the need for such restriction is clear.

We proceed to directly analyse the estimation obtained, without any transformations. Following Doerr and Kostrygin (2017), this estimation reveals two expected behaviours in expectation, which can be analysed in a similar way: as we consider the first variables, the proportion of approximated values grows very fast; then, after a certain point, the proportion of non-approximable values decreases very fast.

We remark that, while Theorem 2.1.1 crucially relies on tools from martingale theory such as Azuma-Hoeffding's inequality, which are not part of standard Computer Science curricula, our argument makes use of much more elementary results[3] which should make it accessible enough for an undergraduate course on randomised algorithms.

[3] Namely, the intermediate value theorem, Markov's inequality, and standard Hoeffding bounds.

2.2 Our argument

In this section, we provide an alternative argument for proving Theorem 2.1.1. It takes shape much like the pseudo-polynomial algorithm we described in the introduction. Leveraging the recursive nature of the problem, we construct a process which, at time t, describes the proportion of the interval $[-1, 1]$ that can be approximated by some subset of the first t variables.

We will show that with a suitable number of uniform variables (proportional to $\log(1/\varepsilon)$) a factor of $1 - \varepsilon/2$ of the values in $[-1, 1]$ can be approximated up to error ε. This implies that any $z \in [-1, 1]$ which cannot be approximated within error ε is at most ε away from a value that can. Therefore it is possible to approximate z up to error 2ε.

2.2.1 Preliminaries

Let $\mathrm{X}_1, \ldots, \mathrm{X}_n$ be realisations of random variables as in Theorem 2.1.1, and, without loss of generality, fix $\varepsilon > 0$. We say a value $z \in \mathbb{R}$ is *ε-approximated at time t* if and only if there exists $S \subseteq [t]$ such that $|z - \sum_{i \in S} \mathrm{X}_i| < \varepsilon$. For $0 \leq t \leq n$, let $f_t \colon \mathbb{R} \to \{0, 1\}$ be the indicator function for the event "z is ε-approximated at time t". Therefore, we have $f_0 = \mathbf{1}_{(-\varepsilon,\varepsilon)}$, since only the interval $(-\varepsilon, \varepsilon)$ can be approximated by an empty set of values. From there, we can exploit the recurrent nature of the problem: a value z can be ε-approximated at time $t + 1$ if and only if either z or $z - \mathrm{X}_{t+1}$ could already be approximated at time t. This implies that for all $z \in \mathbb{R}$ we have that

$$f_{t+1}(z) = f_t(z) + (1 - f_t(z)) f_t(z - \mathrm{X}_{t+1}). \tag{2.1}$$

To keep track of the proportion of values in $[-1, 1]$ that can be ε-approximated at each step, we define, for each $0 \leq t \leq n$, the random variable

$$v_t = \frac{1}{2} \int_{-1}^{1} f_t(z) \, \mathrm{d}z.$$

For better readability, throughout the text we will refer to v_t simply as "the volume."

As we mentioned, it suffices to show that, with high probability, at time n, enough of the interval is ε-approximated (more precisely, that $v_n \geq 1 - \varepsilon/2$) to conclude that the entire interval is 2ε-approximated.

2.2.1.1 Expected behaviour

Our first lemma provides a lower bound on the expected value of v_t.

Lemma 2.2.1. *For all $0 \leq t < n$, it holds that*

$$\mathbb{E}[v_{t+1} \mid \mathrm{X}_1, \ldots, \mathrm{X}_t] \geq v_t \left[1 + \frac{1}{4}(1 - v_t)\right].$$

Proof. The definition of v_t and the recurrence in equation (2.1) give us that

$$\begin{aligned}
\mathbb{E}[v_{t+1} \mid \mathrm{X}_1, \ldots, \mathrm{X}_t] &= \mathbb{E}\left[\frac{1}{2}\int_{-1}^{1} f_{t+1}(z)\,\mathrm{d}z \,\middle|\, \mathrm{X}_1, \ldots, \mathrm{X}_t\right] \\
&= \int_{-1}^{1} \frac{1}{2}\left(\frac{1}{2}\int_{-1}^{1} f_t(z) + (1 - f_t(z)) f_t(z - x)\,\mathrm{d}z\right)\mathrm{d}x \\
&= \frac{1}{2}\int_{-1}^{1} f_t(z)\,\mathrm{d}z \int_{-1}^{1}\frac{1}{2}\,\mathrm{d}x + \frac{1}{2}\int_{-1}^{1}\frac{1}{2}\int_{-1}^{1}(1 - f_t(z)) f_t(z - x)\,\mathrm{d}z\,\mathrm{d}x \\
&= v_t + \frac{1}{4}\int_{-1}^{1}(1 - f_t(z))\int_{-1}^{1} f_t(z - x)\,\mathrm{d}x\,\mathrm{d}z \\
&= v_t + \frac{1}{4}\int_{-1}^{1}(1 - f_t(z))\int_{z-1}^{z+1} f_t(y)\,\mathrm{d}y\,\mathrm{d}z,
\end{aligned}$$

where the last equality holds by substituting $y = z - x$. For the previous ones we apply basic properties of integrals and Fubini's theorem to change the order of integration.

We now look for a lower bound for the last integral in terms of v_t. To this end, we exploit that, since all integrands are non-negative, for all $u \in [-1/2, 1/2]$ we have that

$$\begin{aligned}
\int_{-1}^{1}(1 - f_t(z))\int_{z-1}^{z+1} f_t(y)\,\mathrm{d}y\,\mathrm{d}z &\geq \int_{u-\frac{1}{2}}^{u+\frac{1}{2}}(1 - f_t(z))\int_{z-1}^{z+1} f_t(y)\,\mathrm{d}y\,\mathrm{d}z \\
&\geq \int_{u-\frac{1}{2}}^{u+\frac{1}{2}}(1 - f_t(z))\int_{u-\frac{1}{2}}^{u+\frac{1}{2}} f_t(y)\,\mathrm{d}y\,\mathrm{d}z.
\end{aligned}$$

Both inequalities come from range restrictions: in the first we use that $u \in [-1/2, 1/2]$ implies $[u - 1/2, u + 1/2] \subseteq [-1, 1]$; for the second, we have that $[u - 1/2, u + 1/2] \subseteq [z - 1, z + 1]$ for all $z \in [u - 1/2, u + 1/2]$.[say this before showing the integrals... people are scared of them] [Split into two inequalities.]

To relate the expression to v_t explicitly, we choose u in a way that the window $[u-1/2, u+1/2]$ entails exactly half of v_t. The existence of such u may become clear by recalling the definition of v_t. To make it formal, consider the function given by

$$h(u) = \frac{1}{2}\int_{u-\frac{1}{2}}^{u+\frac{1}{2}} f_t(y)\,\mathrm{d}y,$$

and observe that

$$\min\{h(-1/2),\, h(1/2)\} \leq \frac{v_t}{2}, \qquad \text{and} \qquad \max\{h(-1/2),\, h(1/2)\} \geq \frac{v_t}{2}.$$

Thus, by the intermediate value theorem, there exists $u^* \in [-1/2, 1/2]$ for which $h(u^*) = v_t/2$, that is, for which

$$\frac{1}{2}\int_{u^*-\frac{1}{2}}^{u^*+\frac{1}{2}} f_t(y)\,\mathrm{d}y = \frac{v_t}{2}.$$

Altogether, we can conclude that

$$\begin{aligned}
\mathbb{E}[v_{t+1} \mid \mathrm{X}_1, \dots, \mathrm{X}_t] &= v_t + \frac{1}{4} \int_{-1}^{1} (1 - f_t(z)) \int_{z-1}^{z+1} f_t(y) \, \mathrm{d}y \, \mathrm{d}z \\
&\geq v_t + \frac{1}{2} \int_{u^* - \frac{1}{2}}^{u^* + \frac{1}{2}} (1 - f_t(z)) \left(\frac{1}{2} \int_{u^* - \frac{1}{2}}^{u^* + \frac{1}{2}} f_t(y) \, \mathrm{d}y \right) \mathrm{d}z \\
&= v_t + \left(\frac{1}{2} - \frac{v_t}{2} \right) \frac{v_t}{2} \\
&= v_t \left[1 + \frac{1}{4} (1 - v_t) \right].
\end{aligned}$$

□

Lemma 2.2.1 tells us that, if v_t were to behave as expected, it should grow exponentially up to $1/2$, at which point $1 - v_t$ starts to decrease exponentially. The rest of the proof follows accordingly, with section 2.2.2 analysing the progress of v_t up to one half, and section 2.2.3 analogously following the complementary value, $1 - v_t$, starting from one half. By building on the results from section 2.2.2, we obtain fairly straightforward proofs in section 2.2.3. Thus, the following subsection comprises the core of our argument.

2.2.2 Growth of the volume up to 1/2

Arguably, the main challenge in analysing the RSSP is the existence of over-time dependencies and deciding how to overcome it sets much of the course the proof will take. Our strategy consists in constructing another process which dominates the original one while being free of dependencies.

Let τ_1 be the first time at which the volume exceeds $1/2$, that is, let

$$\tau_1 = \min\{t \geq 0 : v_t > 1/2\}.$$

We just proved that up to time τ_1 the process v_t enjoys exponential growth in expectation. In the following lemma we apply a basic concentration inequality to translate this property into a constant probability of exponential growth for v_t itself.

Lemma 2.2.2. *Given $\beta \in (0, 1/8)$, let $p_\beta = 1 - \frac{7}{8(1-\beta)}$. For all integers $0 \leq t < \tau_1$ it holds that*

$$\Pr[v_{t+1} \geq v_t(1 + \beta) \mid \mathrm{X}_1, \dots, \mathrm{X}_t, \, t < \tau_1] \geq p_\beta.$$

Proof. The result shall follow easily from reverse Markov's inequality (Boyd, Ghosh, Prabhakar, & Shah, 2006, Lemma 4) and the bound from Lemma 2.2.1. However, doing so requires a suitable upper bound on v_{t+1} and, while $2v_t$ would serve the purpose, such bound does not hold in general.

We overcome this limitation by fixing t and considering how much v_t would grow in the next step if we were to consider only values ε-approximated at time t that happen to lie in $[-1, 1]$ after being translated by X_{t+1}. Making it precise by the means of the recurrence in equation (2.1), we define

$$\tilde{v} = \frac{1}{2} \int_{-1}^{1} \left[f_t(z) + (1 - f_t(z)) f_t(z - \mathrm{X}_{t+1}) \cdot \mathbf{1}_{[-1,1]}(z - \mathrm{X}_{t+1}) \right] \mathrm{d}z.$$

This expression differs from the one for v_{t+1} only by the inclusion of the characteristic function of $[-1, 1]$. This not only implies that $\tilde{v} \leq v_{t+1}$, but also that $\tilde{v}$ can replace v_{t+1} in the bound

from Lemma 2.2.1, since the argument provided there eventually restricts itself to integrals within $[-1,1]$, trivialising $\mathbf{1}_{[-1,1]}$. Moreover, as we obtain $\tilde{v}$ without the influence of values from outside $[-1,1]$, we must have $\tilde{v} \leq 2v_t$. Finally, using that $t < \tau_1$ implies $v_t < 1/2$ and chaining the previous conclusions in respective order[This region is quite bad. We need to split the following align, but haven't managed to], we conclude that

$$\begin{aligned}\Pr[v_{t+1} \geq v_t(1+\beta) \mid \mathrm{X}_1, \dots, \mathrm{X}_t, t < \tau_1] &\geq \Pr[\tilde{v} \geq v_t(1+\beta) \mid \mathrm{X}_1, \dots, \mathrm{X}_t, t < \tau_1] \\ &\geq \frac{\mathbb{E}[\tilde{v} \mid \mathrm{X}_1, \dots, \mathrm{X}_t, t < \tau_1] - v_t(1+\beta)}{2v_t - v_t(1+\beta)} \\ &\geq \frac{\frac{9}{8}v_t - v_t(1+\beta)}{2v_t - v_t(1+\beta)} \\ &= 1 - \frac{7}{8(1-\beta)},\end{aligned}$$

where we applied the reverse Markov's inequality in the second step. □

The previous lemma naturally leads us to look for bounds on τ_1, that is, to estimate the time needed for the process to reach volume $1/2$. As expected, the exponential nature of the process yields a logarithmic bound.

Lemma 2.2.3. *Let t be an integer and given $\beta \in (0, 1/8)$, let $p_\beta = 1 - \frac{7}{8(1-\beta)}$ and $i^* = \left\lceil \frac{\log \frac{1}{2\varepsilon}}{\log(1+\beta)} \right\rceil$. If $t \geq i^*/p_\beta$, then*

$$\Pr[\tau_1 \leq t] \geq 1 - \exp\left[-\frac{2p_\beta^2}{t} \left(t - \frac{i^*}{p_\beta} \right)^2 \right].$$

Proof. The main idea behind the proof is to define a new random variable which stochastically dominates τ_1 while being simpler to analyse. We begin by discretising the domain $(0, 1/2]$ of the volume into sub-intervals $\{I_i\}_{0 \leq i \leq i^*}$ defined as follows:

$$\begin{cases} I_0 = (0, \varepsilon], \\ I_i = \left(\varepsilon(1+\beta)^{i-1}, \varepsilon(1+\beta)^i \right] \text{ for } 1 \leq i < i^*, \\ I_{i^*} = \left(\varepsilon(1+\beta)^{i^*-1}, \frac{1}{2} \right], \end{cases}$$

where i^* is the smallest integer for which $\varepsilon\,(1+\beta)^{i^*} \geq 1/2$, that is, $i^* = \left\lceil \frac{\log \frac{1}{2\varepsilon}}{\log(1+\beta)} \right\rceil$.

Now, for each $i \geq 0$, we direct our interest to the number of steps required for v_t to exit the sub-interval I_i after first entering it. By Lemma 2.2.2, this number is majorised by a geometric random variable $\mathrm{Y}_i \sim \mathrm{Geom}(p_\beta)$. Therefore, we can conclude that τ_1 is stochastically dominated by the sum of such variables, that is, for $t \in \mathbb{N}$, we have that

$$\Pr[\tau_1 \geq t] \leq \Pr\left[\sum_{i=1}^{i^*} \mathrm{Y}_i \geq t \right]. \tag{2.2}$$

Let $\mathrm{B}_t \sim \mathrm{Bin}(t, p_\beta)$ be a binomial random variable. For the sum of geometric random variables, it holds that $\Pr[\sum_{i=1}^{i^*} \mathrm{Y}_i \leq t] = \Pr[\mathrm{B}_t \geq i^*]$. Since $\mathbb{E}[\mathrm{B}_t] = tp_\beta$, the Hoeffding bound

for binomial random variables (Dubhashi & Panconesi, 2009, Theorem 1.1) implies that, for all $\lambda \geq 0$, we have that $\Pr[\mathrm{B}_t \leq tp_\beta - \lambda] \leq \exp(-2\lambda^2/t)$. Setting t such that $tp_\beta - \lambda = i^*$, we obtain that

$$\Pr\left[\sum_{i=1}^{i^*} \mathrm{Y}_i \geq t\right] \leq \Pr[\mathrm{B}_t \leq i^*] \leq \exp\left[-\frac{2}{t}\left(tp_\beta - i^*\right)^2\right] = \exp\left[-\frac{2p_\beta^2}{t}\left(t - \frac{i^*}{p_\beta}\right)^2\right],$$

which holds as long as $\lambda = tp_\beta - i^* \geq 0$, that is, for all $t \geq \frac{1}{p_\beta}\left\lceil \frac{\log \frac{1}{2\varepsilon}}{\log(1+\beta)} \right\rceil$.

The thesis follows by applying this to equation (2.2) and passing to complementary events. □

2.2.3 Growth of the volume from 1/2

Here we study the second half of the process: from the moment the volume reaches 1/2 up to the time it gets to $1 - \varepsilon/2$. We do so by analysing the complementary stochastic process, i.e., by tracking, from time τ_1 onwards, the proportion of the interval $[-1, 1]$ that does not admit an ε-approximation. More precisely, we consider the process $\{w_t\}_{t\geq 0}$, defined by $w_t = 1 - v_{\tau_1+t}$.

We shall obtain results for w_t similar to those we have proved for v_t. Fortunately, building on the previous results makes those proofs quite straightforward. We start by noting that a statement analogous to Lemma 2.2.1 follows immediately from the definition of w_{t+1} and Lemma 2.2.1.

Corollary 2.2.4. *For all $t \geq 0$, it holds that*

$$\mathbb{E}[w_{t+1} \mid \mathrm{X}_1, \ldots, \mathrm{X}_{\tau_1+t}] \leq w_t\left[1 - \frac{1}{4}(1 - w_t)\right].$$

Let τ_2 the first time that w_t gets smaller than or equal to $\varepsilon/2$, that is, let

$$\tau_2 = \min\left\{t \geq 0 : w_t \leq \varepsilon/2\right\}.$$

The following lemma bounds this quantity, in analogy to Lemma 2.2.3.

Lemma 2.2.5. *For all $t > 0$, it holds that*

$$\Pr[\tau_2 \leq t] \geq 1 - \frac{1}{\varepsilon}\left(\frac{7}{8}\right)^t.$$

Proof. Applying that $1 - w_t = v_{\tau_1+t} > 1/2$ to Corollary 2.2.4 gives the bound

$$\mathbb{E}[w_{t+1} \mid \mathrm{X}_1, \ldots, \mathrm{X}_{\tau_1+t}] \leq \frac{7}{8} w_t. \tag{2.3}$$

Moreover, from the conditional expectation theory, for any two random variables X and Y, we have $\mathbb{E}[\mathbb{E}[\mathrm{X} \mid \mathrm{Y}]] = \mathbb{E}[\mathrm{X}]$. From this and equation (2.3), we can conclude that

$$\mathbb{E}[w_t] = \mathbb{E}[\mathbb{E}[w_t \mid \mathrm{X}_1, \ldots, \mathrm{X}_{\tau_1+t-1}]] \leq \frac{7}{8}\,\mathbb{E}[w_{t-1}],$$

which, by recursion, yields that

$$\mathbb{E}[w_t] \leq \left(\frac{7}{8}\right)^t \mathbb{E}[w_0] \leq \frac{1}{2}\left(\frac{7}{8}\right)^t.$$

Finally, by Markov's inequality,

$$\Pr[\tau_2 \geq t] \leq \Pr\left[w_t \geq \frac{\varepsilon}{2}\right] \leq \frac{2\,\mathbb{E}[w_t]}{\varepsilon} \leq \frac{1}{\varepsilon}\left(\frac{7}{8}\right)^t,$$

and the thesis follows from considering the complementary event. □

2.2.4 Putting everything together

In this section we conclude our argument, finally proving Theorem 2.1.1. We first prove a more general statement and then detail how it implies the theorem.

Let $\tau = \tau_1 + \tau_2$, the first time at which the process $\{v_t\}_{t\geq 0}$ reaches at least $1 - \varepsilon/2$.

Lemma 2.2.6. *Let $\varepsilon \in (0, 1/3)$. There exist constants $C' > 0$ and $\kappa > 0$ such that for every $t \geq C' \log \frac{1}{\varepsilon}$, it holds that*

$$\Pr[\tau \leq t] \geq 1 - 2\exp\left[-\frac{1}{\kappa t}\left(t - C' \log \frac{1}{\varepsilon}\right)^2\right].$$

Proof. Let $\beta = \frac{1}{16}$ and $p_\beta = 1 - \frac{7}{8(1-\beta)} = \frac{1}{15}$. The definition of τ allows us to apply Lemmas 2.2.3 and 2.2.5 quite directly. Indeed if, for the sake of Lemma 2.2.3, we assume $t \geq \frac{2}{p_\beta}\left\lceil \frac{\log \frac{1}{2\varepsilon}}{\log(1+\beta)} \right\rceil$, we have that

$$\begin{aligned}
\Pr[\tau \leq t] &= \Pr[\tau_1 + \tau_2 \leq t] \\
&\geq \Pr[\tau_1 \leq t/2, \tau_2 \leq t/2] \\
&\geq \Pr[\tau_1 \leq t/2] + \Pr[\tau_2 \leq t/2] - 1 \\
&\geq 1 - \exp\left[-\frac{p_\beta^2}{t}\left(t - \frac{2}{p_\beta}\left\lceil \frac{\log \frac{1}{2\varepsilon}}{\log(1+\beta)} \right\rceil\right)^2\right] - \frac{1}{\varepsilon}\left(\frac{7}{8}\right)^{t/2} \\
&= 1 - \exp\left[-\frac{1}{15^2 t}\left(t - 30\left\lceil \frac{\log \frac{1}{2\varepsilon}}{\log \frac{17}{16}} \right\rceil\right)^2\right] - \frac{1}{\varepsilon}\left(\frac{7}{8}\right)^{t/2}, \qquad (2.4)
\end{aligned}$$

where the second inequality holds by the union bound. The remaining of the proof consists in computations to connect this expression to the one in the statement.

Consider the first exponential term in equation (2.4). Taking $t \geq \frac{60}{\log \frac{17}{16}} \cdot \log \frac{1}{\varepsilon}$, since $\varepsilon < 1/3$, it follows that

$$\exp\left[-\frac{1}{15^2 t}\left(t - 30\left\lceil \frac{\log \frac{1}{2\varepsilon}}{\log \frac{17}{16}} \right\rceil\right)^2\right] \leq \exp\left[-\frac{1}{15^2 t}\left(t - \frac{60}{\log \frac{17}{16}} \cdot \log \frac{1}{\varepsilon}\right)^2\right].$$

Now, consider the second exponential term in equation (2.4). It holds that

$$\begin{aligned}
\frac{1}{\varepsilon}\left(\frac{7}{8}\right)^{\frac{t}{2}} &= \exp\left[\log \frac{1}{\varepsilon} - \frac{t}{2}\log \frac{8}{7}\right] \\
&\leq \exp\left[\log \frac{1}{\varepsilon} - \frac{t}{15}\right] = \exp\left[-\frac{1}{15} \cdot \frac{1}{t - 15 \cdot \log \frac{1}{\varepsilon}} \cdot \left(t - 15 \cdot \log \frac{1}{\varepsilon}\right)^2\right].
\end{aligned}$$

Moreover, for $t \geq 15 \cdot \log \frac{1}{\varepsilon}$,

$$\begin{aligned}
\exp\left[-\frac{1}{15} \cdot \frac{1}{t - 15 \cdot \log \frac{1}{\varepsilon}} \cdot \left(t - 15 \cdot \log \frac{1}{\varepsilon}\right)^2\right] &\leq \exp\left[-\frac{1}{15t}\left(t - 15 \cdot \log \frac{1}{\varepsilon}\right)^2\right] \\
&\leq \exp\left[-\frac{1}{15^2 t}\left(t - \frac{60}{\log \frac{17}{16}} \cdot \log \frac{1}{\varepsilon}\right)^2\right].
\end{aligned}$$

Altogether, we have that

$$\exp\left[-\frac{p_\beta^2}{t}\left(t-\frac{2}{p_\beta}\left\lceil\frac{\log\frac{1}{2\varepsilon}}{\log(1+\beta)}\right\rceil\right)^2\right]+\frac{1}{\varepsilon}\cdot\left(\frac{7}{8}\right)^{t/2}\leq 2\exp\left[-\frac{1}{15^2 t}\left(t-\frac{60}{\log\frac{17}{16}}\cdot\log\frac{1}{\varepsilon}\right)^2\right],$$

and the thesis follows by setting $\kappa = 15^2$ and $C' = 60/\log(17/16)$. □

The expression in the claim of Lemma 2.2.6 can be reformulated as

$$\Pr\left[v_t \geq 1-\frac{\varepsilon}{2}\right] \geq 1-2\exp\left[-\frac{1}{\kappa t}\left(t-C'\log\frac{1}{\varepsilon}\right)^2\right];$$

hence, Theorem 2.1.1 follows by taking $C \geq 3C'$ and observing that once we can approximate all but an $\varepsilon/2$ proportion of the interval $[-1,1]$, any $z \in [-1,1]$ either is ε-approximated itself, or is at most ε away from a value that is, which implies that z is 2ε-approximated.

CHAPTER 3

Multidimensional Random Subset-Sum Problem

In the Random Subset Sum Problem, given n i.i.d. random variables $\mathrm{X}_1, \ldots, \mathrm{X}_n$, we wish to approximate any point $z \in [-1, 1]$ as the sum of a suitable subset $\mathrm{X}_{i_1(z)}, \ldots, \mathrm{X}_{i_s(z)}$ of them, up to error ε. Despite its simple statement, this problem is of fundamental interest to both theoretical computer science and statistical mechanics. More recently, it gained renewed attention for its implications in the theory of Artificial Neural Networks. An immediate multidimensional generalisation of the problem is to consider n i.i.d. d-dimensional random vectors and aim to approximate every point $\boldsymbol{z} \in [-1, 1]^d$. In 1998, G. S. Lueker showed that, in the one-dimensional setting, having $n = \mathcal{O}(\log 1/\varepsilon)$ samples is enough to solve the problem with high probability.
In this chapter, we prove that to solve the d-dimensional version it suffices to have $n = \mathcal{O}(d^3 \log(1/\varepsilon) \log(d/\varepsilon))$. As an application highlighting the potential interest of this result, we prove that a recently proposed neural network model exhibits universality*: with high probability, the model can approximate any neural network within a polynomial overhead in the number of parameters.*

3.1 Introduction

In the *Random Subset Sum Problem (RSSP)*, given a target value z, an error parameter $\varepsilon \in \mathbb{R}_{>0}$ and n independent random variables $\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_n$, one is interested in estimating the probability that there exists a subset $\mathcal{S} \subseteq [n]$ for which

$$\left| z - \sum_{i \in \mathcal{S}} \mathrm{X}_i \right| \leq \varepsilon.$$

Historically, the analysis of this problem was mainly motivated by research on the average case of its deterministic counterpart, the classic Subset Sum Problem, and the equivalent Number Partition Problem. These investigations lead to a number of insightful results, mostly in the 80s and 90s (Lueker, 1982; Karmarkar, Karp, Lueker, & Odlyzko, 1986; Lueker, 1998). In addition, research on the phase transition of the problem extended to the early 2000s, with interesting applications in statistical physics (Mezard & Montanari, 2009; Borgs et al., 2001; Borgs, Chayes, Mertens, & Pittel, 2004).

More recently, one of the results on the RSSP has attracted quite some attention. A simplified statement for it would be

Theorem 3.1.1 (Lueker, (Lueker, 1998))**.** *Let $\mathrm{X}_1, \ldots, \mathrm{X}_n$ be i.i.d. uniform random variables over $[-1, 1]$, and let $\varepsilon \in (0, 1)$. There exists a universal constant $C > 0$ such that, if $n \geq C \log_2 \frac{1}{\varepsilon}$, then, with high probability, for all $z \in [-1, 1]$ there exists a subset $\mathcal{S}_z \subseteq [n]$ for which*

$$\left| z - \sum_{i \in \mathcal{S}_z} \mathrm{X}_i \right| \leq 2\varepsilon.$$

That is, a rather small number (of the order of $\log \frac{1}{\varepsilon}$) of random variables suffices to have a high probability of approximating not only a single target z, but all values in an interval. In fact, this result is asymptotically optimal, since each of the 2^n subsets can cover at most one of two values more than 2ε apart and, hence, we must have $n = \Omega(\log \frac{1}{\varepsilon})$. Also, the original work generalises the result to a wide class of distributions.

Those features allowed Theorem 3.1.1 to be quite successful in applications. In the field of Machine Learning, particularly, many recent works, such as Pensia et al. (2020); da Cunha, Natale, and Viennot (2022); Fischer and Burkholz (2021); Burkholz et al. (2022); Ferbach, Tsirigotis, Gidel, and Bose (2022); C. Wang et al. (2021), leverage this result. We discuss those contributions in more detail in section 3.2.

In this chapter, we investigate a natural multidimensional generalisation of Theorem 3.1.1. Mainly, we prove

Theorem 3.1.2 (Main Theorem)**.** *Given $\varepsilon \in (0, 1)$ and $d, n \in \mathbb{N}$, consider n independent d-dimensional standard normal random vectors $\mathbf{X}_1, \ldots, \mathbf{X}_n$. There exists a universal constant $C > 0$ for which, if*

$$n \geq C d^3 \log_2 \frac{1}{\varepsilon} \cdot \left(\log_2 \frac{1}{\varepsilon} + \log_2 d \right),$$

then, with high probability, for all $\boldsymbol{z} \in [-1, 1]^d$ there exists a subset $s S_{\boldsymbol{z}} \subseteq [n]$ for which

$$\left\| \boldsymbol{z} - \sum_{i \in \mathcal{S}_{\boldsymbol{z}}} \mathbf{X}_i \right\|_\infty \leq 2\varepsilon.$$

Moreover, the approximations can be achieved with subsets of size $\frac{n}{6\sqrt{d}}$.

We believe many promising applications of the RSSP can become feasible with this extension of Theorem 3.1.1 to multiple dimensions. To illustrate this, we consider the *Neural Network Evolution (NNE)* model recently introduced by Gorantla et al. (2019). It is natural to wonder whether their model is *universal*, in the sense that, with high probability, it can approximate any dense feed-forward neural network. While applying Theorem 3.1.1 to this end would yield exponential bounds on the required over-parameterization, in section 3.6 we prove the universality of the model within polynomial bounds. To broaden the scope of our result, we additionally provide some useful generalisations in appendix C. In particular, we extend it to a wide class of distributions, proving an analogous extension to the one (Lueker, 1998) given for Theorem 3.1.1. Finally, in appendices D and E we discuss a discretization of our result and potential applications in the context of nondeterministic random walks.

Organisation of the chapter. After discussing related works in section 3.2, we present a high level overview of the difficulties posed by the problem and of our proof of Theorem 3.1.2 (section 3.3). We then introduce our notation in section 3.4 in preparation for the presentation of our analysis in section 3.5. We follow up with an application of our result to the NNE model (Gorantla et al., 2019) and conclude with some notes on the tightness of our analysis in section 3.7. Finally, generalisations of our results, further extensions, as well as all omitted proofs can be found in the Appendix.

3.2 Related work

As remarked in the Introduction, the first studies of the RSSP were mainly motivated by average-case analyses of the classic Subset Sum and Number Partition problems (Karmarkar et al., 1986; Lueker, 1982, 1998). Both can be efficiently solved if the precision of the values considered is sufficiently low relative to the size of the input set. In particular, Mertens (1998) applies methods from statistical physics to indicate that this is a fundamental property of the problem: the amount of exact solutions of the randomised version exhibits a phase transition when the precision increases relative to the sample size. Borgs et al. (2001) later confirmed formally the existence of a phase transition.

The work of G. S. Lueker on the RSSP dates back to Lueker (1982). In Lueker (1982), the author proves a weaker version of Theorem 3.1.1 and uses it as a tool to analyse the integrality gap of the one-dimensional integer Knapsack problem, i.e., the additive gap between the optimal solution of the integer problem and that of its linear programming relaxation, when the inputs are sampled according to some probability distribution. Later, the same author provided a tighter result of the RSSP in Lueker (1998), which we stated in Theorem 3.1.1. Recently, da Cunha, d'Amore, et al. (2022) exhibited a simpler alternative to the original proof. Dyer and Frieze (1989) generalized the result of the RSSP from Lueker (1982) to tackle the multidimensional formulation of the Knapsack problem. In particular, it is proved that if the number of input variables is $\Theta(d \log \frac{1}{\varepsilon})$, then, with probability $e^{-\mathcal{O}(d)}$, there exists a subset approximating a given target in $\mathbb{R}^d$. Using the latter result as a black box, it is easy to see that one would require $e^{\mathcal{O}(d)} \log \frac{1}{\varepsilon}$ input variables to increase the success probability to a constant value. The result in Dyer and Frieze (1989) has recently been improved by Borst, Dadush, Huiberts, and Tiwari (2023), where tighter bounds on the integrality gap of the multidimensional Knapsack problem were obtained. More specifically,

Borst, Dadush, Huiberts, and Tiwari (2023) showed that, at the cost of an extra polynomial number of input random variables, the success probability of approximating a single target in the space can be increased to a constant value: this probability is achieved whenever the number of variables n satisfies the following relations: $n \geq d^{\frac{9}{4}}$ and $n = \Theta(d^{\frac{3}{2}} \log \frac{1}{\varepsilon})$. Both the analyses in Dyer and Frieze (1989) and Borst, Dadush, Huiberts, and Tiwari (2023) employ the second moment method to estimate the probability that at least one subset approximating the target value exists. Following the same approach, in this chapter we refine the analysis for the second moment method technique. In order to prove Theorem 3.1.2, we show that $n = \Theta(d^2 \log \frac{1}{\varepsilon})$ variables yield constant probability that a subset approximating a single target exists. Our bound is better than that in Borst, Dadush, Huiberts, and Tiwari (2023) for all approximation errors ε which are not exponentially small in the dimension of the space, that is, $\varepsilon = e^{-\mathcal{O}(d^{\frac{3}{4}})}$. We also remark that the result in Borst, Dadush, Huiberts, and Tiwari (2023) is generalised to all distributions whose convergence to a Gaussian is "fast enough", which is a wider class of distribution with respect to the one we provide in this chapter. Nevertheless, as we share the same approach of Borst, Dadush, Huiberts, and Tiwari (2023), with the same arguments we can extend our results to a similar class of distributions. In a recent follow-up (Borst, Dadush, & Mikulincer, 2023), weaker bounds on the multidimensional RSSP are exhibited, which hold for an even wider class of distributions. [1] In Borst, Dadush, and Mikulincer (2023), the number of variables required to solve the problem is $\Theta(d^6 \log \frac{1}{\varepsilon})$. The discrete setting of a variant of the RSSP has also been recently studied in Chen, Jin, et al. (2022) which proves that an integral linear combination (with coefficients in $\{-1, 0, 1\}$) of the sample variables can approximate a range of target values. .

In the last few years, Theorem 3.1.1 has been very useful in studying the *Strong Lottery Ticket Hypothesis*, which states that Artificial Neural Networks (ANN) with random weights are likely to contain an approximation of any sufficiently smaller ANN as a subnetwork. In particular, such claim poses the deletion of connections (pruning) as a theoretically solid alternative to careful calibration of their weights (training). Pensia et al. (2020) uses Theorem 3.1.1 to prove the hypothesis under optimal over-parameterization for dense ReLU neural networks. da Cunha, Natale, and Viennot (2022) extends this result to convolutional networks and Ferbach et al. (2022) further extends the latter to the class of equivariant networks. Also, Burkholz et al. (2022) applies Theorem 3.1.1 to construct neural networks that can be adapted to a variety of tasks with minimal retraining.

3.3 Overview of our analysis

3.3.1 Insights on the difficulty of the problem

In d dimensions, since we need $2^{\Theta(d \log \frac{1}{\varepsilon})}$ hypercubes of radius ε to cover the set $[-1, 1]^d$, we need a sample of $\Omega(d \log \frac{1}{\varepsilon})$ vectors to be able to approximate (up to error ε) every vector in $[-1, 1]^d$.

On the other hand, having $n = \mathcal{O}(d \log \frac{1}{\varepsilon})$ vectors is enough in expectation. To see it, it is sufficient to consider subsets of the sample with $\frac{n}{2}$ vectors. There are $\binom{n}{n/2} \approx 2^{n-o(n)}$ such subsets, each summing to a random vector distributed as $\mathcal{N}(\mathbf{0}, \frac{n}{2} \cdot \boldsymbol{I}_d)$. Thus, given any $\boldsymbol{z} \in$

[1] Borst, Dadush, Huiberts, and Tiwari (2023) considers Gaussian or uniform input random variables, and extends its result to distributions that converge quickly to Gaussian ones. In Borst, Dadush, and Mikulincer (2023), the authors solve the RSSP for input random vectors whose entries follow uniform distributions on finite, discrete sets, and for input log-concave random vectors (i.e., when the density function of these vectors is a log-concave function).

$[-1,1]^d$, each of those sums has probability approximately $\varepsilon^d(\frac{n}{2})^{-\frac{d}{2}} = 2^{-d\log\frac{1}{\varepsilon}-\frac{d}{2}\log\frac{n}{2}}$ of being at most ε far from $\boldsymbol{z}$. We can then conclude that the expected number of approximations is $2^{n-o(n)}\cdot 2^{-d\log\frac{1}{\varepsilon}-\frac{d}{2}\log\frac{n}{2}}$, which is still of order $2^{n-o(n)}$ provided that $n \geq Cd\log\frac{1}{\varepsilon}$ for a sufficiently large constant C.

It would thus suffice to prove concentration bounds on the expectation. The technical challenge is handling the stochastic dependency between subsets of the sample, as pairs of those typically intersect, with many random variables thus appearing for both resulting sums. The original proof of Theorem 3.1.1 (Lueker, 1998) and the simplified one (da Cunha, d'Amore, et al., 2022) address dependencies in similar ways. Both keep track of the fraction of values in $[-1,1]$ that can be approximated by a sum of a subset of the first i random variables, $\mathrm{X}_1,\ldots,\mathrm{X}_i$. Their core goal is to bound the proportional increase in this fraction when an additional random variable X_{i+1} is considered. As it turns out, the *conditional expectation* of this increment can be bounded by a constant factor, regardless of the values of $\mathrm{X}_1,\ldots,\mathrm{X}_i$. Unfortunately, naively extending those ideas to d dimensions leads to an estimation of this increment that is exponentially small in d. It is not clear to the authors how to make the estimation depend polynomially on d without leveraging some knowledge of the actual values of $\mathrm{X}_1,\ldots,\mathrm{X}_i$. In fact, even which kind of assumption on the previous samples could work in this sense is not totally clear.

As for other classical concentration techniques that might appear suitable at first, we remark our failed attempts to leverage an average bounded differences argument (Warnke, 2016). Specifically, we could not identify any natural function related to the fraction of values that can be approximated, which was also Lipschitz relative to the sample vectors. Moreover, both Janson's variant of Chernoff bound (Janson, 2004) and a recent refinement of it (Y. Wang, Ramon, & Guo, 2017) seem to capture the stochastic dependence of the subset sums too loosely for our needs.

3.3.2 Our approach

Our strategy to overcome the difficulties highlighted in the previous subsection consists in a second-moment approach.

Unlike the proofs for the single dimensional case, our argument, at first, analyses the probability of approximating a single target value $\boldsymbol{z} \in [-1,1]^d$. To this end, consider a sample of n independent random vectors $\mathbf{X}_1,\ldots,\mathbf{X}_n$ and a family $\boldsymbol{\mathcal{C}}$ of subsets of the sample. Let Y be the number of subsets in $\boldsymbol{\mathcal{C}}$ whose sum approximates $\boldsymbol{z}$ up to error ε.

For a single subset, it is not hard to estimate the probability with which a subset-sum $\sum_{i\in\mathcal{S}}\mathbf{X}_i$ lies close to $\boldsymbol{z}$. This allows us to easily obtain good bounds on $\mathbb{E}[\mathrm{Y}]$.

We, then, proceed to estimate the variance of Y, circumventing the obstacles mentioned in the previous section by restricting the analysis to families of subsets with sufficiently small pairwise intersections. While this restriction limits the maximum amount of subsets that are available, a standard probabilistic argument allows us to prove the existence of large families of subsets with the desired property, ensuring that $\mathbb{E}[\mathrm{Y}]$ can be large enough for our purposes.

For each pair of subsets, $\mathcal{S}$ and $\mathcal{T}$, we leverage the hypothesis on the size of intersections to consider partitions $\mathcal{S}=\mathcal{S}_\mathrm{A}\cup\mathcal{S}_\mathrm{B}$ and $\mathcal{T}=\mathcal{T}_\mathrm{C}\cup\mathcal{T}_\mathrm{B}$, with $\mathcal{S}_\mathrm{A}$ and $\mathcal{T}_\mathrm{C}$ being large, stochastically independent parts, and the smaller parts $\mathcal{S}_\mathrm{B}$ and $\mathcal{T}_\mathrm{B}$ containing $\mathcal{S}\cap\mathcal{T}$. The bulk of our analysis then consists in deriving careful bounds on their reciprocal dependencies and consequent contributions to the second moment of Y.

The resulting estimate allows us to apply Chebyshev's inequality to Y, obtaining a constant lower bound on $\Pr[\mathrm{Y}\geq 1]$. That is, we conclude that with at least some constant probability at

least one of the subsets yields a suitable approximation of $\boldsymbol{z}$. Finally, we employ a probability-amplification argument in order to apply a union bound over all possible target values in $[-1,1]^d$.

3.4 Preliminaries

Notation Throughout the text we identify the different types of objects by writing their symbols in different styles. This applies to scalars (e.g., x), real random variables (e.g., X), vectors (e.g., $\boldsymbol{x}$), random vectors (e.g., $\mathbf{X}$), matrices (e.g., $\boldsymbol{X}$). and tensors (e.g., $\boldsymbol{\mathcal{X}}$). In particular, for $d \in \mathbb{N}$, the symbol $\boldsymbol{I}_d$ represents the d-dimensional identity matrix, where $\mathbb{N}$ refers to the set of positive integers. Let $n \in \mathbb{N}$. We denote the set $\{1,\ldots,n\}$ by $[n]$, and given a set $\mathcal{S}$ employ the notation $\binom{\mathcal{S}}{n}$ to refer to the family of all subsets of $\mathcal{S}$ containing exactly n elements of $\mathcal{S}$. Let $\boldsymbol{x} \in \mathbb{R}^d$. The notation $\|\boldsymbol{x}\|_2$ represents the euclidean norm of $\boldsymbol{x}$ while $\|\boldsymbol{x}\|_\infty$ denotes its maximum-norm. Moreover, given $r \in \mathbb{R}_{>0}$ we denote the set $\{\boldsymbol{y} \in \mathbb{R}^d : \|\boldsymbol{y}-\boldsymbol{x}\|_\infty \leq r\}$ by $\mathfrak{B}_\infty(\boldsymbol{x}, r)$. We represent the variance of an arbitrary random variable X by σ^2_{X} and its density function by φ_{X}. Finally, the notation $\log(\cdot)$ refers to the binary logarithm. Let $d, n \in \mathbb{N}$ and $\varepsilon \in \mathbb{R}_{>0}$, and consider $\boldsymbol{z} \in [-1,1]^d$ and n independent standard normal d-dimensional random vectors $\mathbf{X}_1,\ldots,\mathbf{X}_n$. Given $\mathcal{S} \subseteq [n]$ we define the random variable

$$\mathrm{Y}_{\mathcal{S},\varepsilon,\boldsymbol{z},\mathbf{X}_1,\ldots,\mathbf{X}_n} = \begin{cases} 1 & \text{if } \|\boldsymbol{z} - \sum_{i\in\mathcal{S}}\|_\infty \leq \varepsilon, \\ 0 & \text{otherwise,} \end{cases}$$

that we represent simply by $\mathrm{Y}_{\mathcal{S}}$ when the other parameters are clear from context. Since we are interested in studying families of subsets, we also define, for $\boldsymbol{\mathcal{C}}$ contained in the power set of $[n]$, the random variable

$$\mathrm{Y}_{\boldsymbol{\mathcal{C}},\varepsilon,\boldsymbol{z},\mathbf{X}_1,\ldots,\mathbf{X}_n} = \sum_{\mathcal{S}\in\boldsymbol{\mathcal{C}}} \mathrm{Y}_{\mathcal{S}},$$

which we represent simply as Y.

For the sake of the analysis, let $\boldsymbol{\mathcal{C}}$ be the family of subsets of $[n]$ with size αn, for any $\alpha > 0$. Notice that the expected intersection size of two elements of $\boldsymbol{\mathcal{C}}$ drawn uniformly at random is $\alpha^2 n$. By choosing α small enough and by using the probabilistic method, we can control the stochastic dependency among subsets.

As $\binom{n}{k} \in [(\frac{n}{k})^k, (\frac{en}{k})^k]$, the following lemma holds.

Lemma 3.4.1. *For all $n \in \mathbb{N}$ and $\alpha \in (0, \frac{1}{2})$, let $\boldsymbol{\mathcal{C}} = \binom{[n]}{\alpha n}$. Then $|\boldsymbol{\mathcal{C}}| = \binom{n}{\alpha n} \in \left[\left(\frac{1}{\alpha}\right)^{\alpha n}, \left(\frac{e}{\alpha}\right)^{\alpha n}\right]$.*

Notice that, while $|\boldsymbol{\mathcal{C}}|$ is still exponential, it already imposes, in expectation, $n = \Omega\left(\frac{d}{\alpha \log \frac{1}{\alpha}} \log \frac{1}{\varepsilon}\right)$ if we are to approximate a single point in $[-1,1]^d$ up to error ε. Indeed, consider a vector $\boldsymbol{v}$ of d Gaussian entries with variance αn, and a point $\boldsymbol{x} \in [-1,1]^d$. The i-th entry of $\boldsymbol{v}$ has probability $\mathcal{O}(\frac{\varepsilon}{\alpha n}) = p < 1$ to lie in the interval $[-\varepsilon + x_i, x_i + \varepsilon] \subseteq [-\varepsilon - 1, 1 + \varepsilon]$. Hence, the vector itself has probability $p^d = \mathcal{O}((\frac{\varepsilon}{\alpha n})^d)$ to approximate the given point up to error ε. As we can dispose of at most $(\frac{e}{\alpha})^{\alpha n}$ such vectors, the expected number of vectors approximating $\boldsymbol{x}$ is

$$\mathcal{O}\left(\left(\frac{e}{\alpha}\right)^{\alpha n} \cdot \left(\frac{\varepsilon}{\alpha n}\right)^d\right),$$

that is at least one only if $n = \Omega\left(\frac{d}{\alpha \log \frac{1}{\alpha}} \log \frac{1}{\varepsilon}\right)$.

3.5 Proof of the main result

As we frequently consider values relatively close to the origin, approximation of the normal distribution by a uniform one is sufficient for many of our estimations.

Lemma 3.5.1. *Let $d \in \mathbb{N}$, $\varepsilon \in (0,1)$, $\sigma \in \mathbb{R}_{>0}$, and $\boldsymbol{z} \in [-1,1]^d$. If $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \cdot \boldsymbol{I}_d)$, then*

$$e^{-\frac{2d}{\sigma^2}} \cdot \frac{(2\varepsilon)^d}{(2\pi\sigma^2)^{\frac{d}{2}}} \leq \Pr[\mathbf{X} \in \mathfrak{B}_\infty(\boldsymbol{z}, \varepsilon)] \leq \frac{(2\varepsilon)^d}{(2\pi\sigma^2)^{\frac{d}{2}}}.$$

As a corollary, we bound the first moment of the random variable Y.

Corollary 3.5.2. *Given $d, n \in \mathbb{N}$, $\varepsilon \in (0,1)$, and $\alpha \in (0, \frac{1}{2})$, let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be independent standard normal d-dimensional random vectors. Then, for all $\boldsymbol{z} \in [-1,1]^d$ and $\mathcal{C} \subseteq \binom{[n]}{\alpha n}$, it holds that*

$$e^{-\frac{2d}{\alpha n}} \frac{(2\varepsilon)^d |\mathcal{C}|}{(2\pi\alpha n)^{\frac{d}{2}}} \leq \mathbb{E}[\mathrm{Y}] \leq \frac{(2\varepsilon)^d |\mathcal{C}|}{(2\pi\alpha n)^{\frac{d}{2}}}.$$

Proof. Let $\mathcal{S} \in \mathcal{C}$ and, hence, $|\mathcal{S}| = \alpha n$. Since $\mathbf{X}_i \sim \mathcal{N}\,\mathbf{0}, \boldsymbol{I}_d$ for all $i \in [n]$, we have that $\sum_{i \in \mathcal{S}} \mathbf{X}_i \sim \mathcal{N}\,\mathbf{0}, \alpha n \cdot \boldsymbol{I}_d$. Therefore, as $\Pr[\mathrm{Y}_\mathcal{S} = 1] = \Pr[\sum_{i \in \mathcal{S}} \mathbf{X}_i \in \mathfrak{B}_\infty(\boldsymbol{z}, \varepsilon)]$, by Lemma 3.5.1, we have that

$$e^{-\frac{2d}{\alpha n}} \frac{(2\varepsilon)^d}{(2\pi\alpha n)^{\frac{d}{2}}} \leq \Pr[\mathrm{Y}_\mathcal{S} = 1] \leq \frac{(2\varepsilon)^d}{(2\pi\alpha n)^{\frac{d}{2}}},$$

and we can conclude the thesis by noting that $\mathbb{E}[\mathrm{Y}] = \sum_{\mathcal{S} \in \mathcal{C}} \Pr[\mathrm{Y}_\mathcal{S} = 1]$. □

We proceed by estimating the second moment of Y.

Lemma 3.5.3. *Given $d, n \in \mathbb{N}$, $\varepsilon \in (0,1)$, and $\alpha \in (0, \frac{1}{6}]$, let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be independent d-dimensional standard normal random vectors, $\boldsymbol{z} \in [-1,1]^d$, and $\mathcal{C} = \binom{[n]}{\alpha n}$. If $n \geq \frac{81}{\alpha(1-2\alpha)}$, then*

$$\mathbb{E}[\mathrm{Y}^2] \leq \mathbb{E}[\mathrm{Y}] + \binom{n}{\alpha n}^2 \cdot \frac{(2\varepsilon)^{2d}}{(2\pi)^d} \cdot \left[\frac{1}{(\alpha n)^d} \cdot (1 - 4\alpha^2)^{-\frac{d}{2}} + \exp\left[-\frac{\alpha^2 n}{3} \right] \right].$$

Proof. We have

$$\begin{aligned} \mathbb{E}[\mathrm{Y}^2] &= \sum_{\mathcal{S}, \mathcal{T} \in \mathcal{C}} \mathbb{E}[\mathrm{Y}_\mathcal{S} \cdot \mathrm{Y}_\mathcal{T}] \\ &= \sum_{\mathcal{S}, \mathcal{T} \in \mathcal{C}} \Pr[\mathrm{Y}_\mathcal{S} = 1, \mathrm{Y}_\mathcal{T} = 1]. \end{aligned} \tag{3.1}$$

Let $(\mathcal{S}, \mathcal{T}) \in \mathcal{C} \times \mathcal{C}$ be a pair of subsets from $\mathcal{C}$ sampled uniformly at random, and let $\mathrm{K}_{\mathcal{S},\mathcal{T}} \in [\alpha n]$ be the size of their intersection. As $\Pr[\mathrm{Y}_\mathcal{S} = 1, \mathrm{Y}_\mathcal{T} = 1]$ depends only on the size of $\mathcal{S} \cap \mathcal{T}$, we may rewrite equation (3.1) as follows.

$$\begin{aligned} \mathbb{E}[\mathrm{Y}^2] &= \binom{n}{\alpha n}^2 \cdot \sum_{k=0}^{\alpha n} \Pr[\mathrm{K}_{\mathcal{S},\mathcal{T}} = k] \Pr[\mathrm{Y}_\mathcal{S} = 1, \mathrm{Y}_\mathcal{T} = 1 \mid \mathrm{K}_{\mathcal{S},\mathcal{T}} = k] \\ &= \mathbb{E}[\mathrm{Y}] + \binom{n}{\alpha n}^2 \cdot \sum_{k=0}^{\alpha n - 1} \Pr[\mathrm{K}_{\mathcal{S},\mathcal{T}} = k] \Pr[\mathrm{Y}_\mathcal{S} = 1, \mathrm{Y}_\mathcal{T} = 1 \mid \mathrm{K}_{\mathcal{S},\mathcal{T}} = k], \end{aligned} \tag{3.2}$$

where the latter equality follows by observing that $\Pr[\mathrm{K}_{\mathcal{S},\mathcal{T}} = \alpha n] = \binom{n}{\alpha n}^{-1}$.

The core of our argument is to upper bound the joint probability $\Pr[\mathrm{Y}_{\mathcal{S}} = 1, \mathrm{Y}_{\mathcal{T}} = 1 \mid \mathrm{K}_{\mathcal{S},\mathcal{T}} = k]$. For the sake of simplicity, assume the outcome of $(\mathcal{S}, \mathcal{T})$ to be a pair of subsets $(\mathcal{S}, \mathcal{T}) \in \mathcal{C}^2$ with $|\mathcal{S} \cap \mathcal{T}| = k$: as we argued before, this assumption is justified as $\Pr[\mathrm{Y}_{\mathcal{S}} = 1, \mathrm{Y}_{\mathcal{T}} = 1]$ depends only on the cardinality of $\mathcal{S} \cap \mathcal{T}$.

Let $\mathrm{X}_1, \cdots, \mathrm{X}_n \sim \mathcal{N}(0,1)$ and $z \in [-1,1]$. Consider the partitions $\mathcal{S} = \mathcal{S}_{\mathrm{A}} \cup \mathcal{S}_{\mathrm{B}}$ and $\mathcal{T} = \mathcal{T}_{\mathrm{C}} \cup \mathcal{T}_{\mathrm{B}}$, with $\mathcal{S}_{\mathrm{B}} = \mathcal{T}_{\mathrm{B}} = \mathcal{S} \cap \mathcal{T}$, and let

$$\mathrm{A} = \sum_{i \in \mathcal{S}_{\mathrm{A}}} \mathrm{X}_i, \qquad \mathrm{C} = \sum_{i \in \mathcal{T}_{\mathrm{C}}} \mathrm{X}_i, \qquad \mathrm{B} = \sum_{i \in \mathcal{S} \cap \mathcal{T}} \mathrm{X}_i.$$

In this way, we have $\sum_{i \in \mathcal{S}} \mathrm{X}_i = \mathrm{A} + \mathrm{B}$ and $\sum_{i \in \mathcal{T}} \mathrm{X}_i = \mathrm{C} + \mathrm{B}$, with A, C independent random variables distributed as $\mathcal{N}(0, \sigma_{\mathrm{A}}^2)$ and $\mathrm{B} \sim \mathcal{N}(0, \sigma_{\mathrm{B}}^2)$, with $\sigma_{\mathrm{A}}^2 = \alpha n - k$ and $\sigma_{\mathrm{B}}^2 = k$.

With this setup, we have,

$$\Pr[\mathrm{Y}_{\mathcal{S}} = 1, \mathrm{Y}_{\mathcal{T}} = 1] = \left(\Pr[\mathrm{A} + \mathrm{B} \in (z-\varepsilon, z+\varepsilon),\ \mathrm{C} + \mathrm{B} \in (z-\varepsilon, z+\varepsilon)]\right)^d.$$

By the law of total probability, it holds that

$$\begin{aligned}
&\Pr[\mathrm{A} + \mathrm{B} \in (z-\varepsilon, z+\varepsilon),\ \mathrm{C} + \mathrm{B} \in (z-\varepsilon, z+\varepsilon)] \\
&\quad = \int_{\mathbb{R}} \varphi_{\mathrm{B}}(x) \cdot \Pr[\mathrm{A} + x \in (z-\varepsilon, z+\varepsilon),\ \mathrm{C} + x \in (z-\varepsilon, z+\varepsilon)]\ \mathrm{d}x \\
&\quad = \int_{\mathbb{R}} \varphi_{\mathrm{B}}(x) \cdot \Pr[\mathrm{A} \in (z-x-\varepsilon, z-x+\varepsilon),\ \mathrm{C} \in (z-x-\varepsilon, z-x+\varepsilon)]\ \mathrm{d}x \\
&\quad = \int_{\mathbb{R}} \varphi_{\mathrm{B}}(x) \cdot \left(\Pr[\mathrm{A} \in (z-x-\varepsilon, z-x+\varepsilon)]\right)^2\ \mathrm{d}x, \qquad (3.3)
\end{aligned}$$

where the last equality follows from the independence of A and C.

Since A is a normal random variable with 0 average, by Claim A.5, we have that

$$\begin{aligned}
\int_{\mathbb{R}} \varphi_{\mathrm{B}}(x) \cdot \left(\Pr[\mathrm{A} \in (z-x-\varepsilon, z-x+\varepsilon)]\right)^2\ \mathrm{d}x &\le \int_{\mathbb{R}} \varphi_{\mathrm{B}}(x) \cdot \left(\Pr[\mathrm{A} \in (x-\varepsilon, x+\varepsilon)]\right)^2\ \mathrm{d}x \\
&= \int_{\mathbb{R}} \varphi_{\mathrm{B}}(x) \cdot \left(\int_{x-\varepsilon}^{x+\varepsilon} \varphi_{\mathrm{A}}(y)\ \mathrm{d}y\right)^2\ \mathrm{d}x. \qquad (3.4)
\end{aligned}$$

As $\Pr[\mathrm{Y}_{\mathcal{S}} = 1, \mathrm{Y}_{\mathcal{T}} = 1]$ increases monotonically with $|\mathcal{S} \cap \mathcal{T}|$, we devide the proof in two cases.

First case: $\mathrm{K}_{\mathcal{S},\mathcal{T}} \le 2\alpha^2 n$. As the joint probability $\Pr[\mathrm{Y}_{\mathcal{S}} = 1, \mathrm{Y}_{\mathcal{T}} = 1 \mid \mathrm{K}_{\mathcal{S},\mathcal{T}} = k]$ increases with k, we just bound $\Pr[\mathrm{Y}_{\mathcal{S}} = 1, \mathrm{Y}_{\mathcal{T}} = 1 \mid \mathrm{K}_{\mathcal{S},\mathcal{T}} = 2\alpha^2 n]$. Hence, $\sigma_{\mathrm{A}}^2 = \alpha n(1-2\alpha)$ and $\sigma_{\mathrm{B}}^2 = 2\alpha^2 n$.

The hypothesis on n implies that $2\sigma_a^2 \ge 162$, so, by Claim A.6,

$$\begin{aligned}
\left(\int_{x-\varepsilon}^{x+\varepsilon} \varphi_{\mathrm{A}}(y)\ \mathrm{d}y\right)^2 &\le \left[\int_{x-\varepsilon}^{x+\varepsilon} \frac{\exp\left(-\frac{(x+\varepsilon)^2}{2\sigma_{\mathrm{A}}^2}\right) + \exp\left(-\frac{(x-\varepsilon)^2}{2\sigma_{\mathrm{A}}^2}\right)}{2\sqrt{2\pi\sigma_{\mathrm{A}}^2}} \cdot \exp\left(\frac{\varepsilon^2}{2\sigma_{\mathrm{A}}^2}\right)\ \mathrm{d}y\right]^2 \\
&= \frac{(2\varepsilon)^2}{2\pi\sigma_{\mathrm{A}}^2} \cdot \frac{\exp\left(-\frac{(x+\varepsilon)^2}{\sigma_{\mathrm{A}}^2}\right) + \exp\left(-\frac{(x-\varepsilon)^2}{\sigma_{\mathrm{A}}^2}\right) + 2\exp\left(-\frac{x^2+\varepsilon^2}{\sigma_{\mathrm{A}}^2}\right)}{4} \cdot \exp\left(\frac{\varepsilon^2}{\sigma_{\mathrm{A}}^2}\right) \\
&= e^{\varepsilon^2/\sigma_{\mathrm{A}}^2} \cdot \frac{1}{\sqrt{2}} \cdot \frac{(2\varepsilon)^2}{\sqrt{2\pi\sigma_{\mathrm{A}}^2}} \cdot \frac{\varphi_{\mathrm{A}/\sqrt{2}}(x+\varepsilon) + \varphi_{\mathrm{A}/\sqrt{2}}(x-\varepsilon) + 2e^{-\varepsilon^2/\sigma_{\mathrm{A}}^2} \cdot \varphi_{\mathrm{A}/\sqrt{2}}(x)}{4}.
\end{aligned}$$

Moreover, it holds that

$$\begin{aligned}
&\int_{\mathbb{R}} \varphi_{\mathrm{B}}(x) \cdot \left[\varphi_{\mathrm{A}/\sqrt{2}}(x+\varepsilon) + \varphi_{\mathrm{A}/\sqrt{2}}(x-\varepsilon) + 2e^{-\varepsilon^2/\sigma_{\mathrm{A}}^2} \cdot \varphi_{\mathrm{A}/\sqrt{2}}(x)\right] \mathrm{d}x \\
&\quad = (\varphi_{\mathrm{B}} * \varphi_{\mathrm{A}/\sqrt{2}})(\varepsilon) + (\varphi_{\mathrm{B}} * \varphi_{\mathrm{A}/\sqrt{2}})(-\varepsilon) + 2e^{-\varepsilon^2/\sigma_{\mathrm{A}}^2} \cdot (\varphi_{\mathrm{B}} * \varphi_{\mathrm{A}/\sqrt{2}})(0) \\
&\quad = \varphi_{\mathrm{B+A}/\sqrt{2}}(\varepsilon) + \varphi_{\mathrm{B+A}/\sqrt{2}}(-\varepsilon) + 2e^{-\varepsilon^2/\sigma_{\mathrm{A}}^2} \cdot \varphi_{\mathrm{B+A}/\sqrt{2}}(0) \\
&\quad = \frac{2e^{-\varepsilon^2/\sigma^2_{\mathrm{B+A}/\sqrt{2}}} + 2e^{-\varepsilon^2/\sigma_{\mathrm{A}}^2}}{\sqrt{2\pi\sigma^2_{\mathrm{B+A}/\sqrt{2}}}} \\
&\quad \leq 4 \cdot \frac{e^{-\varepsilon^2/\sigma_{\mathrm{A}}^2}}{\sqrt{2\pi\sigma^2_{\mathrm{B+A}/\sqrt{2}}}},
\end{aligned}$$

here $*$ denotes the convolution operation, and the last inequality comes from the hypothesis $\alpha \leq \frac{1}{6}$, which implies that $\sigma^2_{\mathrm{B+A}/\sqrt{2}} \leq \sigma^2_{\mathrm{A}}$.

Altogether, we have

$$\begin{aligned}
\Pr[\mathrm{Y}_{\mathcal{S}} = 1, \mathrm{Y}_{\mathcal{T}} = 1] &\leq \left(e^{\varepsilon^2/\sigma_{\mathrm{A}}^2} \cdot \frac{1}{\sqrt{2}} \cdot \frac{(2\varepsilon)^2}{\sqrt{2\pi\sigma_{\mathrm{A}}^2}} \cdot \frac{e^{-\varepsilon^2/\sigma_{\mathrm{A}}^2}}{\sqrt{2\pi\sigma^2_{\mathrm{B+A}/\sqrt{2}}}}\right)^d \\
&= \left(\frac{(2\varepsilon)^2}{2\pi} \cdot \frac{1}{\sqrt{2\sigma_{\mathrm{A}}^2\sigma^2_{\mathrm{B+A}/\sqrt{2}}}}\right)^d \\
&= \frac{(2\varepsilon)^{2d}}{(2\pi\alpha n)^d} \cdot (1-4\alpha^2)^{-\frac{d}{2}},
\end{aligned} \tag{3.5}$$

where the last equality follows from recalling that $\sigma_{\mathrm{B}}^2 = 2\alpha^2 n$ and $\sigma_{\mathrm{A}}^2 = \alpha n(1-2\alpha)$, and, thus, $\sigma^2_{\mathrm{B+A}/\sqrt{2}} = 2\alpha^2 n + \frac{\alpha n}{2}(1-2\alpha)$.

Second case: $\mathrm{K}_{\mathcal{S},\mathcal{T}} > 2\alpha^2 n$. As $\int_{x-\varepsilon}^{x+\varepsilon} \varphi_{\mathrm{A}}(y)\,\mathrm{d}y \leq \int_{-\varepsilon}^{+\varepsilon} \varphi_{\mathrm{A}}(y)\,\mathrm{d}y$ for any $x \in \mathbb{R}$, a trivial application of Lemma 3.5.1 in equation (3.4) implies that

$$\begin{aligned}
\Pr[\mathrm{A}+\mathrm{B} \in (z-\varepsilon, z+\varepsilon), \mathrm{C}+\mathrm{B} \in (z-\varepsilon, z+\varepsilon)] &\leq \frac{(2\varepsilon)^2}{(2\pi\sigma_{\mathrm{A}}^2)} \int_{\mathbb{R}} \varphi_{\mathrm{B}}(x)\,\mathrm{d}x \\
&\leq \frac{(2\varepsilon)^2}{(2\pi\sigma_{\mathrm{A}}^2)},
\end{aligned}$$

which is maximum when $\sigma_A^2 = 1$ (i.e., the intersection has size $k = \alpha n - 1$). Hence,

$$\Pr[Y_{\mathcal{S}} = 1, Y_{\mathcal{T}} = 1] \leq \frac{(2\varepsilon)^{2d}}{(2\pi)^d}. \tag{3.6}$$

Consider now equation (3.2). $\mathrm{K}_{\mathcal{S},\mathcal{T}}$ follows a hypergeometric distribution $\mathcal{H}(n, \alpha n, \alpha n)$, and its expectation is $\alpha^2 n$. The common Chernoff bounds (Lemma A.2) hold also for hypergeometric distributions (Doerr, 2011, Theorem 1.17). Hence, we have

$$\Pr[\mathrm{K}_{\mathcal{S},\mathcal{T}} \geq 2\alpha^2 n] \leq \exp\left[-\frac{\alpha^2 n}{3}\right]. \tag{3.7}$$

By plugging equations (3.5) to (3.7) in equation (3.2) we obtain

$$\mathbb{E}[\mathrm{Y}^2] \leq \mathbb{E}[\mathrm{Y}] + \binom{n}{\alpha n}^2 \cdot \frac{(2\varepsilon)^{2d}}{(2\pi)^d} \cdot \left[\frac{1}{(\alpha n)^d} \cdot (1-4\alpha^2)^{-\frac{d}{2}} + \exp\left[-\frac{\alpha^2 n}{3}\right]\right].$$

□

Remark 3.5.1 – In the proof of Lemma 3.5.3, after applying the law of total probability it is possible to employ Lemma 3.5.1 to estimate the joint probability. While this simplifies the argument, doing so would ultimately weaken the bound in Theorem 3.5.5. In fact, in section 3.7 we argue that the estimation we provide is essentially optimal.

We now apply the second moment method to estimate the probability that $\mathrm{Y} \geq 1$.

Lemma 3.5.4. *Given $d, n \in \mathbb{N}$, $\varepsilon \in (0,1)$, and $\alpha \in (0, \frac{1}{6}]$, let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be independent d-dimensional standard normal random vectors, $\boldsymbol{z} \in [-1,1]^d$, and $\boldsymbol{\mathcal{C}} = \binom{[n]}{\alpha n}$. If $\alpha \leq \frac{1}{6\sqrt{d}}$, and*

$$n \geq \max\left\{\frac{18d\log\frac{d}{\alpha}}{\alpha^2}, \frac{8d}{\alpha\log\frac{1}{\alpha}}\left(\log d + \log\frac{1}{\varepsilon} + 1\right) + \frac{8}{\alpha}\right\},$$

then

$$\Pr[\mathrm{Y} \geq 1] \geq \frac{4}{9}.$$

Proof. Since Y is an integer-valued random variable, by the second moment method, it holds that

$$\begin{aligned}\Pr[\mathrm{Y} > 0] &= \Pr[\mathrm{Y} \geq 1] \\ &\geq \frac{\mathbb{E}[\mathrm{Y}]^2}{\mathbb{E}[\mathrm{Y}^2]}.\end{aligned}$$

Using Lemmas 3.4.1 and 3.5.3 and Corollary 3.5.2, we obtain

$$\begin{aligned}\frac{\mathbb{E}[\mathrm{Y}^2]}{\mathbb{E}[\mathrm{Y}]^2} &\leq \frac{\exp\left[\frac{4d}{\alpha n}\right] \cdot (2\pi\alpha n)^{\frac{d}{2}}}{\binom{n}{\alpha n} \cdot (2\varepsilon)^d} + \exp\left[\frac{4d}{\alpha n}\right] \cdot (1-4\alpha^2)^{-\frac{d}{2}} + \exp\left[\frac{4d}{\alpha n} - \frac{\alpha^2 n}{3}\right] \cdot (\alpha n)^d \\ &\leq \frac{\exp\left[\frac{4d}{\alpha n}\right] \cdot (2\pi\alpha n)^{\frac{d}{2}}}{\frac{1}{\alpha^{\alpha n}} \cdot (2\varepsilon)^d} + \exp\left[\frac{4d}{\alpha n}\right] \cdot (1-4\alpha^2)^{-\frac{d}{2}} + \exp\left[\frac{4d}{\alpha n}\right] \cdot \exp\left[-\frac{\alpha n}{3}\left(\alpha - \frac{d\log\alpha n}{\alpha n}\right)\right].\end{aligned}$$

As $0 < \alpha \leq \frac{1}{2}$ and $n \geq \frac{4d}{\alpha\log\frac{1}{\alpha}}\left[\log\frac{1}{\varepsilon} + \log 2\pi d\right] + \frac{8}{\alpha}$, Claim A.3 implies that

$$\frac{\exp\left[\frac{4d}{\alpha n}\right] \cdot (2\pi\alpha n)^{\frac{d}{2}}}{\frac{1}{\alpha^{\alpha n}} \cdot (2\varepsilon)^d} \leq \varepsilon.$$

At the same time, as $n \geq \frac{68d}{\alpha}$ and $\alpha \leq \frac{1}{6\sqrt{d}}$, by Claim A.4

$$\frac{e^{\frac{4d}{\alpha n}}}{(1-4\alpha^2)^{\frac{d}{2}}} \leq 1 + \frac{1}{8}.$$

Furthermore, $n \geq \frac{68d}{\alpha}$ implies that

$$\begin{aligned}\exp\left[\frac{4d}{\alpha n}\right] &\leq \exp\left[\frac{1}{16}\right] \\ &\leq 1 + \frac{1}{8}.\end{aligned}$$

Finally, as $\frac{d \log \alpha n}{\alpha n}$ decreases (in n) when $n \geq \frac{6d \log \frac{d}{\alpha}}{\alpha^2}$ and $0 < \alpha \leq \frac{1}{2}$, we have

$$\begin{aligned}\exp\left[-\frac{\alpha n}{3}\left(\alpha - \frac{d \log \alpha n}{\alpha n}\right)\right] &\leq \exp\left[\frac{-\alpha n}{3}\left(\alpha - \frac{\alpha \log\left(\frac{6d}{\alpha} \log \frac{d}{\alpha}\right)}{6 \log \frac{d}{\alpha}}\right)\right] \\ &= \exp\left[-\frac{\alpha n}{3}\left(\alpha - \frac{\alpha \log\left(\frac{d}{\alpha} \log\left(\left(\frac{d}{\alpha}\right)^6\right)\right)}{\log\left(\left(\frac{d}{\alpha}\right)^6\right)}\right)\right] \\ &\leq \exp\left[-\frac{\alpha^2 n}{6}\right],\end{aligned}$$

where for the last inequality we have used that, for $\alpha \leq \frac{1}{6}$,

$$\frac{\log\left(\frac{d}{\alpha} \log\left(\left(\frac{d}{\alpha}\right)^6\right)\right)}{\log\left(\left(\frac{d}{\alpha}\right)^6\right)} \leq \frac{1}{2};$$

thus, for $n \geq \frac{6 \log 9}{\alpha^2}$,

$$\exp\left[-\frac{\alpha^2 n}{6}\right] \leq \frac{1}{9}.$$

Hence, as $\varepsilon \leq 1$, if $\alpha \in (0, \frac{1}{6\sqrt{d}})$ and $n \geq \max\left\{\frac{18d \log \frac{d}{\alpha}}{\alpha^2}, \frac{8d}{\alpha \log \frac{1}{\alpha}}\left(\log d + \log \frac{1}{\varepsilon} + 1\right) + \frac{8}{\alpha}\right\}$, it holds that

$$\begin{aligned}\frac{\mathbb{E}[\mathrm{Y}]^2}{\mathbb{E}[\mathrm{Y}^2]} &\geq \frac{1}{\varepsilon + \frac{9}{8} + \frac{1}{8}} \\ &\geq \frac{4}{9}.\end{aligned}$$

□

Applying an union bound, we amplify the last lemma to get our main result.

Theorem 3.5.5. *Let $\varepsilon \in (0,1)$ and given $d, n \in \mathbb{N}$ let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be independent standard normal d-dimensional random vectors, and let $N = \max\left\{d^2(1 + \log d), \frac{d^{\frac{3}{2}}}{\log d}\left(1 + \log d + \log \frac{1}{\varepsilon}\right)\right\}$. There exists a universal constant $C > 0$ such that, if $n \geq dCN \log \frac{1}{\varepsilon}$, then, with probability at least*

$$1 - \exp\left[-\ln 2 \cdot \left(\frac{n - dCN \log \frac{1}{\varepsilon}}{CN}\right)\right],$$

for all $\boldsymbol{z} \in [-1,1]^d$ *there exists a subset* $\mathcal{S}_{\boldsymbol{z}} \subseteq [n]$ *for which*

$$\left\| \boldsymbol{z} - \sum_{i \in \mathcal{S}_{\boldsymbol{z}}} \mathbf{X}_i \right\|_\infty \leq 2\varepsilon.$$

Moreover, this remains true even when restricted to subsets of size $\frac{n}{6\sqrt{d}}$.

Theorem 3.1.2 follows directly from Theorem 3.5.5.

3.6 Application to Neural Net Evolution

In this section, we present an application of our main result on the multidimensional RSSP (see Theorem 3.1.2) to a neural network model recently introduced in Gorantla et al. (2019).

We first provide a description of their model in a setting relevant to our application. Then, we prove that their model exhibits *universality*; namely, with high probability, it can approximate any neural network within a polynomial overhead in the number of parameters.

3.6.1 The NNE model

The *Neural Net Evolution* (NNE) model (Gorantla et al., 2019) has been recently introduced as an alternative approach to train neural networks, based on evolutionary methods. The aim is to provide a biologically inspired alternative to the backpropagation process behind ANNs (Rumelhart, Hinton, & Williams, 1986b; Goodfellow et al., 2016), which happens in evolutionary time, instead of lifetime.

The NNE model is inspired by a standard update rule in population genetics and, in Gorantla et al. (2019), it is shown to succeed in creating neural networks that can learn linear classification problems reasonably well with no explicit backpropagation.

To define the NNE model, we first need to define random genotypes. Given a vector $\boldsymbol{p} \in [0,1]^n$, a random *genotype* $\boldsymbol{x} \in \{0,1\}^n$ is sampled by setting $x_i = 1$ with probability p_i, independently for each i. Each entry x_i indicates whether or not a *gene* is active.

Then, for each i, a random tensor $\boldsymbol{\Theta}^{(i)} \in \mathbb{R}^{\ell \times d \times d}$ is sampled. In the original version of the model (Gorantla et al., 2019), each entry of the tensor is chosen independently and uniformly at random from $[-1,1]$ with probability β, while it is set to 0, otherwise. For the sake of our application, we here consider a natural variant where the entries of the tensor are independently drawn from a standard normal distribution.

Now, given a genotype $\boldsymbol{x} \in \{0,1\}^n$, we define

$$\boldsymbol{\Theta}_{\boldsymbol{x}} = \sum_{i\,:\,x_i=1} \boldsymbol{\Theta}^{(i)}. \tag{3.8}$$

Each genotype is then associated with a *feed-forward neural network*, represented by a weighted complete multipartite directed graph. The graph is formed by layers $\{L_i\}_{i=0}^{\ell}$ of d nodes and two consecutive layers are fully connected via a biclique whose edge weights are determined by the tensor $\boldsymbol{\Theta}_{\boldsymbol{x}}$ in the following manner: for every $i \in [\ell]$, the edge between the j-th node of layer L_{i-1} and the k-th node of layer L_i has weight $(\boldsymbol{\Theta}_{\boldsymbol{x}})_{ijk}$.

Equation (3.8) tells us that if a gene is active then it gives a random contribution to each weight of the genotype network.

The learning process in the NNE model works by updating the genotype probabilities $\boldsymbol{p}$ according to some standard population genetics equations (Bürger, 2000; Chastain, Livnat, Papadimitriou, & Vazirani, 2014). In Gorantla et al. (2019), it is proved that the adopted update rule indirectly performs backpropagation and enables to decrease the loss function of the networks.

3.6.2 Universality and RSSP

Let $f\colon \mathbb{R}^d \to \mathbb{R}^d$ be a feed-forward neural network of the form

$$f(\boldsymbol{y}) = \boldsymbol{W}_\ell \operatorname{relu}(\boldsymbol{W}_{\ell-1} \dots \operatorname{relu}(\boldsymbol{W}_1 \boldsymbol{y})), \tag{3.9}$$

where $\boldsymbol{W}_i \in \mathbb{R}^{d\times d}$ is a weight matrix.

The restrictions on the weight matrix sizes $d \times d$ aim only to ease presentation and can be adapted to any arbitrary dimensions.

Let us construct a third-order tensor $\boldsymbol{\Theta}_f \in \mathbb{R}^{\ell\times d\times d}$ by stacking the weight matrices $\boldsymbol{W}_1, \dots, \boldsymbol{W}_\ell$. We correspondingly denote f by $f_{\boldsymbol{\Theta}}$. Conversely, every tensor $\boldsymbol{\Theta} \in \mathbb{R}^{\ell\times d\times d}$ is associated with a neural network $f_{\boldsymbol{\Theta}}$ in the form of equation (3.9) whose corresponding weight matrices are the tensor slices, that is, $\boldsymbol{W}_m = (\boldsymbol{\Theta})_{\substack{i=m\\ j,k\in[d]}}$ for every $m \in [\ell]$.

We can use Theorem 3.1.2 to prove a notion of universality for the NNE model.

Theorem 3.6.1. *Let $\varepsilon > 0$ and $n, d, \ell \in \mathbb{N}$. Let $\mathcal{F}$ be the class of neural networks $f\colon \mathbb{R}^d \to \mathbb{R}^d$ of the form given in equation (3.9) such that their corresponding tensor satisfies $\max_{ijk}|(\boldsymbol{\Theta}_f)_{ijk}| < 1$. A constant $C > 0$ exists such that, if $n \geq C(\ell \cdot d \cdot d)^3 \log\frac{1}{\varepsilon} \cdot \left(\log\frac{1}{\varepsilon} + \log(\ell \cdot d \cdot d)\right)$, then, with high probability, the tensors $\boldsymbol{\Theta}^{(1)}, \dots, \boldsymbol{\Theta}^{(n)}$ associated to each gene are such that, for any $f \in \mathcal{F}$, there is a genotype $\boldsymbol{x} \in \{0,1\}^n$ which satisfies*

$$\max_{\substack{i\in[\ell]\\ j,k\in[d]}} |(\boldsymbol{\Theta}_f)_{ijk} - (\boldsymbol{\Theta}_{\boldsymbol{x}})_{ijk}| < 2\varepsilon.$$

We note that standard techniques (e.g., Pensia et al. (2020); da Cunha, Natale, and Viennot (2022)) can be used to provide bounds on the approximation of the output of neural networks, as well as translating Theorem 3.6.1 for general network architectures (e.g., convolutional neural networks).

3.7 Tightness of analysis

In Lemma 3.4.1 we prove the existence of a suitable family of subsets via a probabilistic argument, sampling their elements uniformly at random. The same argument also implies that the pairwise intersections of almost all subsets is at least $\frac{\alpha^2 n}{2}$. In the next result, we assume such lower bound and prove that our estimation of the joint probability $\Pr[\mathrm{Y}_{\mathcal{S}} = 1, \mathrm{Y}_{\mathcal{T}} = 1]$ in Lemma 3.5.3 (specifically, in equation (3.5)), is essentially tight. Namely, the next lemma implies that it is not possible to obtain a high-probability bound on Y in Lemma 3.5.4.

Lemma 3.7.1. *Given $d, n \in \mathbb{N}$, $\varepsilon \in (0,1)$, and $\alpha \in (0, \frac{1}{2})$, let $\mathbf{X}_1, \dots, \mathbf{X}_n$ be independent standard normal d-dimensional random vectors and $\boldsymbol{z} \in [-1,1]^d$. If any two subsets in $\boldsymbol{\mathcal{C}}$ intersect in at least $\frac{\alpha^2 n}{2}$ elements and $n \geq \frac{10}{\alpha(2-\alpha)}$, then*

$$\frac{\alpha^2 n}{2} \leq |\mathcal{S} \cap \mathcal{T}| \leq 2\alpha^2 n$$

it holds that

$$\Pr[\mathrm{Y}_{\mathcal{S}} = 1, \mathrm{Y}_{\mathcal{T}} = 1] \geq \frac{(2\varepsilon)^{2d}}{(2\pi\alpha n)^d} \cdot (1 - \frac{\alpha^2}{4})^{-\frac{d}{2}} \cdot \exp\left(-\frac{3d}{\alpha n}\right).$$

We can extend the above result by letting $\boldsymbol{z}$ lie in a wider range. This will be useful for the generalisation section appendix C.

Remark 3.7.1 – If $\lambda > 1$ and $\boldsymbol{z} \in [-\lambda\sqrt{n}, \lambda\sqrt{n}]^d$, then we have

$$\Pr[\mathrm{Y}_{\mathcal{S}} = 1, \mathrm{Y}_{\mathcal{T}} = 1] \geq \frac{(2\varepsilon)^{2d}}{(2\pi\alpha n)^d} \cdot \left(1 - \frac{\alpha^2}{4}\right)^{-\frac{d}{2}} \cdot \exp(-\frac{3\lambda^2 d}{\alpha}).$$

PART

Generalising the Strong LTH

CHAPTER 4

Convolutional Neural Networks

The lottery ticket hypothesis states that a randomly-initialized neural network contains a small subnetwork which, when trained in isolation, can compete with the performance of the original network. Recent theoretical works proved an even stronger version: every sufficiently over-parameterized (dense) neural network contains a subnetwork that, even without training, achieves accuracy comparable to that of the trained large network. These works left extending the result to convolutional neural networks (CNNs) as an open problem. In this chapter, we provide such generalization by showing that, with high probability, it is possible to approximate any CNN by pruning a random CNN whose size is larger by a logarithmic factor.

4.1 Introduction

Many impressive successes in machine learning are reached through neural network architectures with a huge number of trainable parameters. Consequently, substantial research in the field aims at reducing the size of such networks while maintaining good accuracy; e.g., for deployment in resource constrained devices (Yang, Chen, & Sze, 2017).

A major empirical fact of such endeavour is the contrast between the initial model over-parametrization, which appears necessary for effective training, and the extent to which the size of the resulting model can be reduced through compression techniques. Among the latter, *pruning methods* appear as a mature and efficient way of achieving significant compression, often without incurring any accuracy loss (Blalock et al., 2020). Recently, the aforementioned contrast between the initial and final number of parameters has been addressed by the *lottery ticket hypothesis* (Frankle & Carbin, 2019), or LTH for short, which states that any randomly initialized network contains *lottery tickets*; that is, sparse subnetworks that can be trained just once and reach the performance of the fully-trained original network. This hypothesis was first verified experimentally, leveraging pruning methods to identify the lottery tickets (Frankle & Carbin, 2019; N. Lee, Ajanthan, & Torr, 2019).

Ramanujan et al. (2020) then proposed a stronger version of the hypothesis, named *strong lottery ticket hypothesis (SLTH)* by Pensia et al. (2020): it stipulates that a network with random weights contains, with high probability, sub-networks that can approximate any given sufficiently-smaller neural network. In other words, a sufficiently large and randomly initialized network that can be successfully trained for a task, could instead be suitably pruned to obtain a network that, even without training, achieves good accuracy. Experimental support for this stronger version were reported by Ramanujan et al. (2020); Zhou et al. (2019); Y. Wang et al. (2020), which find lottery tickets in a range of architectures, including convolutional neural networks (CNNs). A first rigorous proof of the SLTH was given by Malach et al. (2020) for the case of dense networks (i.e., consisting of fully connected layers). Pensia et al. (2020) and Orseau et al. (2020) successively improved this result by showing that logarithmic over-parametrization is sufficient. Their results are also restricted to dense networks and they leave as an open problem to extend it to CNNs.

Our contributions. We extend and complete the proof of the SLTH (and thus, also, of the LTH), for classical network architectures which can combine convolutional and fully connected layers. More precisely, we prove that any CNN with given weights can be approximated by pruning a CNN with random weights (random CNN for short), with the latter being larger than the former by a logarithmic factor. We also provide basic experiments showing that starting from a random CNN which is roughly 30 times larger than LeNet5, it is possible to compute in few hours a pruning mask that allows to approximate the trained convolutional layers of LeNet5 with relative error 10^{-3}, even when ignoring some hypothesis of our theoretical result. Our theoretical analysis follows the approach of Malach et al. (2020) and make use of two layers to approximate one. We borrow from Pensia et al. (2020) the use of random subset sum (RSS) (Lueker, 1998) to approximate a given weight via the sum of a subset of a sample of random weights, and carefully design instances of RSS via a combination of two convolutional layers. By controlling the error accumulated by each layer with Young's convolution inequality, we establish the following result.

> **Informal version of Theorem 4.2.3.** Given $\varepsilon, \delta > 0$, any CNN with k parameters and ℓ layers, and kernels with ℓ_1 norm at most 1, can be approximated within error

ε by pruning a random CNN with $\mathcal{O}\left(k \log \frac{k\ell}{\min\{\varepsilon,\delta\}}\right)$ parameters and 2ℓ layers with probability at least $1-\delta$.

This result generalizes those by Pensia et al. (2020), Orseau et al. (2020), and Malach et al. (2020) as dense layers can be regarded as convolutional layers where kernel and input sizes match.

Roadmap. After discussing related work in the next section, we state our theoretical results alongside a high-level idea of the proofs. Successively, we report our experimental results. Finally, in section 4.4, we provide detailed proofs of our statements.

4.1.1 Related Work

Pruning methods are classical neural network compression strategies that date back to the 80s (LeCun, Denker, & Solla, 1989; M. Mozer & Smolensky, 1988). We recommend the recent survey Blalock et al. (2020) for an overview of the current state of research on these techniques.

As for the lottery ticket hypothesis, Lange (2020) summarizes the progress on the topic until the results by Malach et al. (2020). In the following we briefly mention works which are not discussed in Lange (2020). Cosentino, Zaiter, Pei, and Zhu (2019) shows that lottery tickets can be adversarially trained, yielding sparse and robust neural networks. Soelen and Sheppard (2019) shows that lottery tickets are transferable, in the sense of showing remarkable accuracy for tasks other than the original one for which they have been found. Sabatelli, Kestemont, and Geurts (2021) further shows that minimal retraining on a new task allows lottery tickets to often achieve better generalization than models trained ad-hoc for the task. H. Yu, Edunov, Tian, and Morcos (2020) empirically supports that the LTH holds also in the context of reinforcement learning and natural language processing. Fischer and Burkholz (2021) extends works on the SLTH to accommodate biases in practical settings. Diffenderfer and Kailkhura (2021) shows that lottery tickets are robust to extreme quantization of the weights. Aladago and Torresani (2021) provides a method to train networks where each initial weight is restricted to few possible random values. An extreme case of the latter is to share only a single (random) value among all weights, and focus the training solely on finding the best architecture (Gaier & Ha, 2019).

This chapter also relates to recent papers investigating properties of random CNNs, such as Ulyanov, Vedaldi, and Lempitsky (2020) which observes that random CNNs already seem to capture some natural image statistics required for tasks such as de-noising and inpainting.

4.2 Theoretical Results

We start by introducing some of our notation.

Given $n \in \mathbb{N}$, we denote the set $\{1,\dots,n\}$ by $[n]$. The symbol $*$ represents the convolution operation, $\odot$ represents the element-wise (Hadamard) product, and σ represents ReLU activation function. Finally, the notation $\|\cdot\|_1$ refers to the sum of the absolute values of each entry in a tensor while $\|\cdot\|_\infty$ denotes the maximum norm: the maximum among the absolute value of each entry. Those are akin to vector norms and should not be confused with operator norms.

We restrict our setting to convolutional neural networks $f\colon [0,1]^{D\times D\times c_0} \to \mathbb{R}^{D\times D\times c_\ell}$ of the form

$$f(\boldsymbol{X}) = \boldsymbol{K}^\ell * \sigma(\boldsymbol{K}^{\ell-1} * \cdots \sigma(\boldsymbol{K}^1 * \boldsymbol{X})),$$

where $\boldsymbol{K}^i \in \mathbb{R}^{d_i \times d_i \times c_{i-1} \times c_i}$, and the convolutions have no bias and are suitably padded with zeros. The restrictions on tensor sizes and the exclusion of bias terms[1] aim only to ease presentation.

Our initial goal is to approximate a convolution with a single kernel, as depicted in figure 4.1, using convolutions with (pruned) random kernels. We achieve this by the means of the structure presented in figure 4.2, using two convolutions with random tensors.

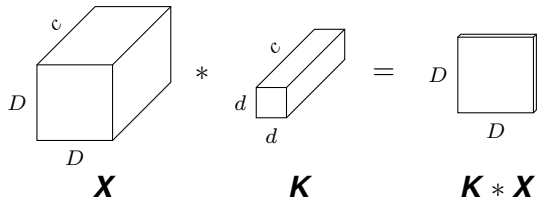


Figure 4.1: Schematics of the convolution between an input $\boldsymbol{X} \in \mathbb{R}^{D \times D \times c}$ and a kernel $\boldsymbol{K} \in \mathbb{R}^{d \times d \times c}$ resulting in a $D \times D$ matrix.

Lemma 4.2.1 asserts that, with high probability, we can prune this structure to approximate the output of a convolution with any given kernel as long as the amount of random kernels is large enough.

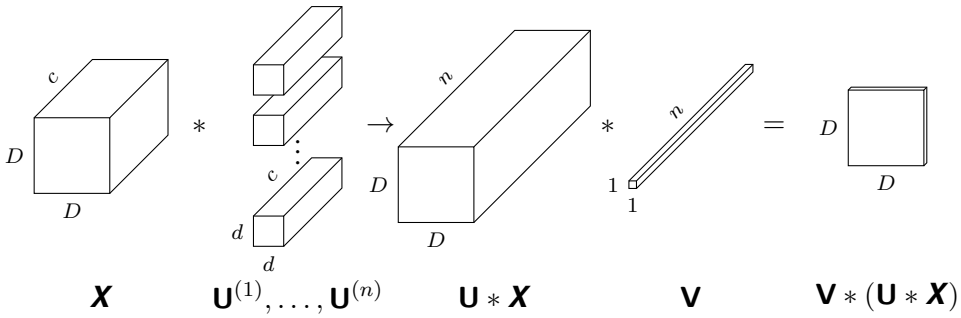


Figure 4.2: Schematics of the use of two convolutions to approximate the operation depicted in figure 4.1. The elements of the set $\mathbf{U} = \{\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(n)}\}$ and $\mathbf{V}$ are random tensors. Notice that the intermediate tensor $\mathbf{U} * \boldsymbol{X}$ has size $D \times D \times n$ and yet the final output is a $D \times D$ matrix.

Lemma 4.2.1 (Single kernel)**.** *Let $D, d, c, n \in \mathbb{N}$, and $\varepsilon, C \in \mathbb{R}_{>0}$, where $n \geq C \log \frac{d^2 c}{\varepsilon}$, $\mathbf{U} \in \mathbb{R}^{d \times d \times c \times n}$, $\mathbf{V} \in \mathbb{R}^{1 \times 1 \times n \times 1}$, and $\boldsymbol{S} \in \{0,1\}^{\operatorname{shape}(\mathbf{U})}$, where the entries of $\mathbf{U}$ and $\mathbf{V}$ are i.i.d.* $\operatorname{Uniform}([-1,1])$ *random variables. Moreover, define the random CNN $g\colon [0,1]^{D \times D \times c} \to \mathbb{R}^{D \times D \times 1}$ and its pruned version $g_{\boldsymbol{S}}$ by*

$$g(\boldsymbol{X}) = \mathbf{V} * \sigma(\mathbf{U} * \boldsymbol{X}) \quad \text{and} \quad g_{\boldsymbol{S}}(\boldsymbol{X}) = \mathbf{V} * \sigma((\mathbf{U} \odot \boldsymbol{S}) * \boldsymbol{X}).$$

Then, we can choose constant C independently from other parameters so that, with probability at least $1 - \varepsilon$, for all $\boldsymbol{K} \in [-1,1]^{d \times d \times c \times 1}$ with $\|\boldsymbol{K}\|_1 \leq 1$, there exists a pruning mask $\boldsymbol{S}$ such that

$$\sup_{\boldsymbol{X} \in [0,1]^{D \times D \times c}} \|\boldsymbol{K} * \boldsymbol{X} - g_{\boldsymbol{S}}(\boldsymbol{X})\|_\infty < \varepsilon.$$

[1] If biases are present, the structures used in the proofs also puts them in a RSS configuration. Thus the results can be readily adapted by replacing the d_i^2 terms by $d_i^2 + 1$.

Proof idea. We leverage the absence of negative entries in the input and an initial pruning of $\mathbf{U}$ to bypass the ReLU non-linearity. This allows us to virtually replace the operations in g by a single convolution with a random kernel obtained by combining $\mathbf{U}$ and $\mathbf{V}$. Each entry of this resulting kernel is the sum of n random variables, where we can choose to include/exclude each term in the sum by choosing to keep/prune the relevant weights. We finish the proof by applying Theorem H.2 to conclude that n variables suffice to approximate all entries, simultaneously, with enough precision to ensure the thesis. □

We now extend Lemma 4.2.1 to an entire layer. As before, a detailed proof is provided in section 4.4.2.

Lemma 4.2.2 (Convolutional Layer)**.** *Let $D, d, c_0, c_1, n \in \mathbb{N}$, and $\varepsilon, C \in \mathbb{R}_{>0}$, where $n \geq Cc_1 \log \frac{d^2 c_0 c_1}{\varepsilon}$, $\mathbf{U} \in \mathbb{R}^{d\times d\times c_0\times n}$, $\mathbf{V} \in \mathbb{R}^{1\times 1\times n\times c_1}$, $\boldsymbol{S} \in \{0,1\}^{\mathrm{shape}(\mathbf{U})}$ and $\boldsymbol{T} \in \{0,1\}^{\mathrm{shape}(\mathbf{V})}$, where the entries of $\mathbf{U}$ and $\mathbf{V}$ are i.i.d.* $\mathrm{Uniform}([-1,1])$ *random variables. Finally, define the random CNN $g\colon [0,1]^{D\times D\times c_0} \to \mathbb{R}^{D\times D\times c_1}$ and its pruned version $g_{\boldsymbol{T},\boldsymbol{S}}(\boldsymbol{X})$ by*

$$g(\boldsymbol{X}) = \mathbf{V} * \sigma(\mathbf{U} * \boldsymbol{X}) \quad \text{and} \quad g_{\boldsymbol{T},\boldsymbol{S}}(\boldsymbol{X}) = (\mathbf{V} \odot \boldsymbol{T}) * \sigma((\mathbf{U} \odot \boldsymbol{S}) * \boldsymbol{X}).$$

Then, we can choose constant C independently from other parameters so that, with probability at least $1-\varepsilon$, for all $\boldsymbol{K} \in [-1,1]^{d\times d\times c_0\times c_1}$ with $\|\boldsymbol{K}\|_1 \leq 1$, there exist masks $\boldsymbol{S}$ and $\boldsymbol{T}$ such that

$$\sup_{\boldsymbol{X}\in[0,1]^{D\times D\times c_0}} \|\boldsymbol{K} * \boldsymbol{X} - g_{\boldsymbol{T},\boldsymbol{S}}(\boldsymbol{X})\|_\infty < \varepsilon.$$

Proof Idea. The lemma follows by applying Lemma 4.2.1 to each kernel independently so that all of them are approximated by a factor of at most ε/c_1. Such approximation allows us to apply the union bound so that the desired approximation holds simultaneously for all c_1 output kernels with probability at least $1-\varepsilon$. □

Next, we extend Lemma 4.2.2 from a single layer to the entire network, thus proving our main result. A detailed proof is given in appendix G.

Theorem 4.2.3 (Convolutional Network)**.** *Let $D, d, c_0, \ell \in \mathbb{N}$, and $\varepsilon, C, \delta \in \mathbb{R}_{>0}$. For each $i \in [\ell]$, let $c_i, n_i \in \mathbb{N}$, where $n_i \geq Cc_i \log \frac{c_{i-1}c_i d_i^2 \ell}{\min\{\varepsilon,\delta\}}$, and $\boldsymbol{L}^{2i-1} \in \mathbb{R}^{d_i\times d_i\times c_{i-1}\times n_i}$, $\boldsymbol{L}^{2i} \in \mathbb{R}^{1\times 1\times n_i\times c_i}$, $\boldsymbol{S}^{2i-1} \in \{0,1\}^{\mathrm{shape}(\boldsymbol{L}^{2i-1})}$, $\boldsymbol{S}^{2i} \in \{0,1\}^{\mathrm{shape}(\boldsymbol{L}^{2i})}$, where the entries of $\boldsymbol{L}^1, \ldots, \boldsymbol{L}^{2\ell}$ are i.i.d.* $\mathrm{Uniform}([-1,1])$ *random variables and define the random 2ℓ-layer CNN $g\colon [0,1]^{D\times D\times c_0} \to \mathbb{R}^{D\times D\times c_\ell}$ and its pruned version $g_{\boldsymbol{S}^1,\ldots,\boldsymbol{S}^{2\ell}}(\boldsymbol{X})$ by*

$$g(\boldsymbol{X}) = \boldsymbol{L}^{2\ell} * \sigma(\cdots \sigma(\boldsymbol{L}^1 * \boldsymbol{X})) \quad \text{and} \quad g_{\boldsymbol{S}^1,\ldots,\boldsymbol{S}^{2\ell}}(\boldsymbol{X}) = (\boldsymbol{L}^{2\ell} \odot \boldsymbol{S}^{2\ell}) * \sigma[\cdots \sigma[(\boldsymbol{L}^1 \odot \boldsymbol{S}^1) * \boldsymbol{X}]].$$

Finally, let $\mathcal{F}$ be the class of functions from $[0,1]^{D\times D\times c_0}$ to $\mathbb{R}^{D\times D\times c_\ell}$ such that, for each $f \in \mathcal{F}$

$$f(\boldsymbol{X}) = \boldsymbol{K}^\ell * \sigma(\boldsymbol{K}^{\ell-1} * \cdots \sigma(\boldsymbol{K}^1 * \boldsymbol{X})),$$

where, for each $i \in [\ell]$, $\boldsymbol{K}^i \in [-1,1]^{d_i\times d_i\times c_{i-1}\times c_i}$ and $\|\boldsymbol{K}^i\|_1 \leq 1$.

Then, we can choose constant C independently from other parameters so that, with probability at least $1-\delta$, the following holds for every $f \in \mathcal{F}$:

$$\inf_{\forall i\in[2\ell],\, \boldsymbol{S}^i\in\{0,1\}^{\mathrm{shape}(\boldsymbol{L}^i)}} \sup_{\boldsymbol{X}\in[0,1]^{D\times D\times c_0}} \|f(\boldsymbol{X}) - g_{\boldsymbol{S}^1,\ldots,\boldsymbol{S}^{2\ell}}(\boldsymbol{X})\|_\infty < \varepsilon.$$

Proof Idea. The proof leverages Lemma 4.2.2 in an analogous way to how the latter relied on Lemma 4.2.1; namely, we apply Lemma 4.2.2 by requiring an approximation factor that guarantees, with sufficient probability, that a suitable approximation is reached across all layers simultaneously. The latter requirement is responsible for the ℓ factor which appears in the logarithms of the dimensions of each random tensor $\boldsymbol{L}_i$. □

4.2.1 Discussion on Theorem 4.2.3

Size analysis. For each layer, we emulate a 4-D kernel $\boldsymbol{K}$ with size $d_i \times d_i \times c_{i-1} \times c_i$ with two 4-D kernels $\mathbf{U}$ and $\mathbf{V}$ with size $d_i \times d_i \times c_{i-1} \times n$ and $1 \times 1 \times n \times c_i$ respectively with $n \geq C c_i \log \frac{c_{i-1} c_i d_i^2 \ell}{\min\{\varepsilon,\delta\}}$. Under the technical assumption $c_i = \mathcal{O}(d_i^2 c_{i-1})$ for $i \in [\ell]$, the size of $\mathbf{V}$ is within a constant factor of that of $\mathbf{U}$, and the whole random network we prune has size $\mathcal{O}\big(k \log \frac{k\ell}{\min\{\varepsilon,\delta\}}\big)$, where k is the size of the network we want to approximate. This technical assumption is met for all classical convolutional networks used for image processing with a reasonably small constant in the big O notation. We come back to this assumption below.

Limitations. The properties of convolutional layers require stronger hypotheses in Theorem 4.2.3 when compared with the results for dense layers Malach et al. (2020) or Pensia et al. (2020). First, we require non-negative inputs for all layers, however, since the output of the ReLU function is never negative, this restriction is only relevant for the input of the first layer. The mentioned works avoid this restriction by exploiting the identity $a = \sigma(a) - \sigma(-a)$ to deal separately with the positive and negative entries. The fact that each entry of the output of a convolution is affected by potentially multiple input entries prevents us from employing a similar strategy. Nonetheless, we remark that, while this is a relevant theoretical indication of the challenges imposed by the operation of convolution, in practice the inclusion of biases suffices to easily convert any CNN with a domain including negative values into an equivalent CNN that takes only non-negative inputs. Finally, the possibly multidimensional entries of convolutions also motivate the restriction on the norm of the target weight tensors in terms of the 1-norm.

Generalizations. For the sake of simplicity, we state and prove Theorem 4.2.3 in a restricted setting. It is worth remarking on a series of generalisations that can be obtained at the mere cost of making the proofs more technically involved. First, the proof could also consider other parameters, such as stride, padding, average pooling and other operations that can be seen as convolutions. Moreover, we could consider more general convolutions, not necessarily 2-D, operating on tensors of any sufficiently large dimension. In particular, it is not necessary to assume that $\mathbf{V}$ has size $1 \times 1 \times n \times c_i$ in the above analysis. Using a 5-D tensor with size $d \times d \times c_{i-1} \times (n/c_i) \times c_1$ for $\mathbf{U}$, an appropriate convolution $\mathbf{U} * \boldsymbol{X}$ would result in a $D \times D \times (n/c_i) \times c_i$ tensor, and we could use a $1 \times 1 \times (n/c_i) \times c_i$ tensor for $\mathbf{V}$ without the need for the mask $\boldsymbol{T}$ by performing in parallel c_i appropriate convolutions. Note that the size of $\mathbf{V}$ is then smaller than the size of $\mathbf{U}$. The technical assumption used in the above size analysis is thus not necessary to guarantee that logarithmic over-parametrization is sufficient. Finally, observe that our results generalize to any probability distribution for the weights that contains a b-scaled $\mathrm{Uniform}([-a, a])$ for some constant $a > 0$ (in the sense of Definition H.1 in appendix H), where the parameters a and b only impact the constants in the theorem.

4.3 Experiments

As networks with higher parameter count tend to be more robust to noise, we stick to the small CNN architecture used by Pensia et al. (2020), namely, LeNet5 (LeCun et al., 1989) with ReLU activations. We conduct our experiments by first training a the network to 98.99% test accuracy on MNIST dataset (Lecun, Bottou, Bengio, & Haffner, 1998). To avoid well-known limitations of the MNIST dataset (in particular its large number of zero entries), we also trained it on the Fashion-MNIST dataset (H. Xiao, Rasul, & Vollgraf, 2017) to 89.12% test accuracy. We adopted Kaiming Uniform (K. He, Zhang, Ren, & Sun, 2015) for weight initialization, a batch size of 64 and trained for 50 epochs using ADAM optimizer (Kingma & Ba, 2015) with learning rate of 0.001, exponential decay of 0.9 and momentum estimate of 0.999, the default values in Flux.jl (Innes et al., 2018) machine learning library.

Once the network is trained we change its weights for a random subset sum approximation of them. More precisely, for each weight w we sample $\boldsymbol{x}$ from $\mathrm{Uniform}([-1,1]^n)$ and use Gurobi optimization software (Gurobi Optimization, LLC, 2021) to solve the mixed-integer program

$$\min_{a_1,\dots,a_n} \left| w - \sum_{i=1}^{n} a_i \cdot x_i \right| \qquad \text{s.t.} \quad a_i \in \{0,1\} \quad \forall i \in [n],$$

where n is the sample size. Solving this subset sum problem with $n = 30$ for the 2572 parameters in the convolutional layers of LetNet takes around 1 hour on 32 cores of a Intel® Xeon® Gold 6240 CPU @ 2.60GHz.

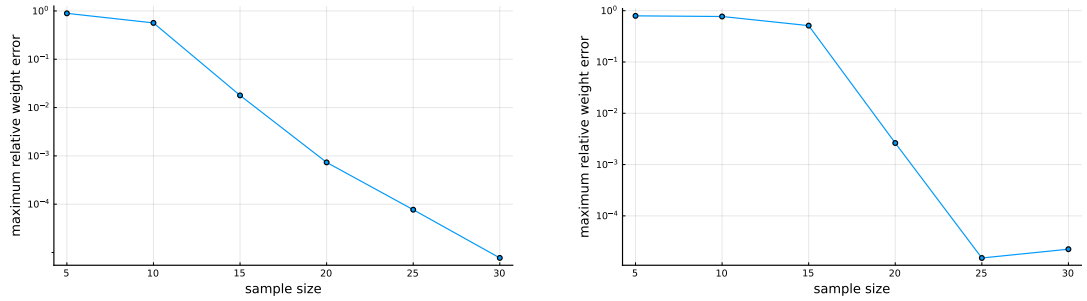


Figure 4.3: Relative error of random subset sum approximation of the convolution weights of a LeNet5 trained on MNIST (left) and on Fashion-MNIST (right). The error is given in logarithmic scale as the maximum distance between a weight and its approximation for different sample sizes.

Figure 4.3 shows the accuracy of the approximation for different sample sizes. We start to obtain good approximations (error smaller than 10^{-2}) from sample sizes around 15-20. Also, when comparing to the weights obtained for MNIST and for Fashion-MNIST, we have better approximations for the smaller sample sizes for MNIST. We believe this is due to the fact that the training on Fashion-MNIST resulted in filters with larger weights (up to a factor 2, roughly), since a larger sample size is necessary to approximate a larger interval of values (see Theorem H.2).

The high precision in the approximation of most weights leads to negligible change in the accuracy of the network. For this reason, we focus on studying the error at the output of the convolutional section of LeNet5, right before the flattening. Also, at this point the activation tensor has dimension $7 \times 7 \times 16$ as opposed to the vector of size 10 at the end of LeNet5.

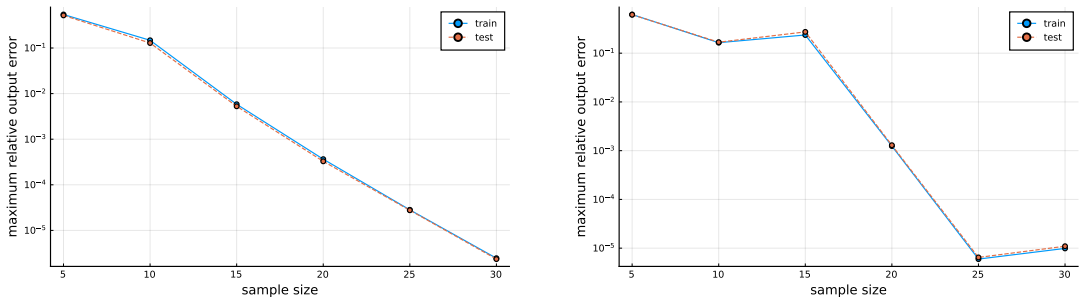


Figure 4.4: Maximum relative output error for the convolutional portion of LeNet5 trained on MNIST (left) and Fashion-MNIST (right) for different sample sizes. The maximum is computed over all images in the dataset.

Figure 4.4 shows the maximum relative error for all approximated outputs compared to original ones. The relative error of the output for an input image is computed as the maximum activation error divided by the maximum original activation (both maxima are taken over all $7 \times 7 \times 16$ activations). Once again, MNIST leads to better precision for the smaller sample sizes. This can be explained by the fact that weights are better approximated in that range with MNIST as seen in figure 4.3. In both cases, we get a relative error close to 10^{-3} with sample size 20, and even better with larger sample sizes. Within the settings of Theorem 4.2.3, this corresponds to expanding the convolutional portion of the network by a factor of, roughly, 30 if we take into account kernel sizes and number of channels. This high precision is achieved even though the trained weights do not satisfy the norm restrictions of Theorem 4.2.3. Indeed, as we do not use any explicit regularization, the 1-norms of the kernels obtained are quite high (from 50 to 15000 roughly for both datasets).

4.4 Technical Analyses

4.4.1 Single Kernel Approximation (Proof of Lemma 4.2.1)

Our first goal is to bypass the non-linearity so we can combine the two convolutions in $g(\boldsymbol{X}) = \mathbf{V} * \sigma(\mathbf{U} * \boldsymbol{X})$ into a single one. Given that the activation function under consideration is the ReLU, it suffices to ensure that its input has no negative entry. Hence, we prune all negative entries of $\mathbf{U}$, obtaining the tensor $\mathbf{U}^+ = \max\{\mathbf{0}, \mathbf{U}\}$, where the maximum is applied entry-wise. Since, by hypothesis, the entries of the input $\boldsymbol{X}$ are non-negative, it follows that the entries of the tensor $\mathbf{U}^+ * \boldsymbol{X}$ are also non-negative. Therefore,

$$\mathbf{V} * \sigma(\mathbf{U}^+ * \boldsymbol{X}) = \mathbf{V} * (\mathbf{U}^+ * \boldsymbol{X}). \tag{4.1}$$

We now look at the first convolution on the right side of equation (4.1). By Definition (1.3.1), we have

$$\begin{aligned}
[\mathbf{V} * (\mathbf{U}^+ * \boldsymbol{X})]_{r,s,1} &= \sum_{t=1}^{n} \mathsf{V}_{1,1,t,1} \cdot (\mathbf{U}^+ * \boldsymbol{X})_{r,s,t} \\
&= \sum_{t=1}^{n} \mathsf{V}_{1,1,t,1} \cdot \left(\sum_{i,j\in[d],k\in[c]} \mathsf{U}^+_{i,j,k,t} \cdot \boldsymbol{X}_{r-i+1,s-j+1,k} \right) \\
&= \sum_{t=1}^{n} \sum_{i,j\in[d],k\in[c]} \left(\mathsf{V}_{1,1,t,1} \cdot \mathsf{U}^+_{i,j,k,t} \right) \cdot \boldsymbol{X}_{r-i+1,s-j+1,k} \\
&= \sum_{i,j\in[d],k\in[c]} (\sum_{t=1}^{n} \mathsf{V}_{1,1,t,1} \cdot \mathsf{U}^+_{i,j,k,t}) \cdot \boldsymbol{X}_{r-i+1,s-j+1,k}.
\end{aligned}$$

The equation above shows that performing $\mathbf{V} * (\mathbf{U}^+ * \boldsymbol{X})$ is equivalent to performing a single convolution between $\boldsymbol{X}$ and a tensor $\boldsymbol{L} \in \mathbb{R}^{d\times d\times c\times 1}$ whose coordinates are given by

$$\boldsymbol{L}_{i,j,k,1} = \sum_{t=1}^{n} \mathsf{V}_{1,1,t,1} \cdot \mathsf{U}^+_{i,j,k,t}.$$

This reveals a RSS configuration where we can choose to include/exclude each value $\mathsf{V}_{1,1,t,1} \cdot \mathsf{U}^+_{i,j,k,t}$ in the sum by choosing to keep/prune $\mathsf{U}^+_{i,j,k,t}$. Since equation (4.1) continues to hold after further pruning $\mathbf{U}^+$, we finish our proof by doing exactly that: we leverage Theorem H.2 to ensure that, with high probability, we can solve this RSS problem for each entry of $\boldsymbol{L}$ to approximate the respective entry of $\boldsymbol{K}$.

To see that we can apply Theorem H.2 in this setting, for $\varepsilon' > 0$, $i, j \in [d]$, and $k \in [c]$, denote by $\mathcal{E}_{i,j,k,\varepsilon'}$ the event

$$\left\{ \forall z \in [-1,1],\ \exists \mathcal{S} \subseteq [n] : \left| z - \sum_{t\in\mathcal{S}} \mathsf{V}_{1,1,t,1} \cdot \mathsf{U}^+_{i,j,k,t} \right| < \varepsilon' \right\}.$$

We now use the RSS result, (Lueker, 1998, Corollary 3.3) (Theorem H.2 in appendix H), to show that there exists constants a, b such that

$$\mathbb{E}\left[\max_{z\in[-n/32,n/32]} \min_{\mathcal{S}\subseteq[n]} |z - \sum_{t\in\mathcal{S}} \mathsf{V}_{1,1,t,1} \cdot \mathsf{U}^+_{i,j,k,t}| \right] \leq a e^{-bn}.$$

It is not hard to show that, since $(\mathsf{V}_{1,1,t,1})_{1\leq t\leq n}$ and $(\mathsf{U}^+_{i,j,k,t})_{1\leq t\leq n}$ are i.i.d. $\mathrm{Uniform}([-1,1])$ random variables, then the value of the density of $(\mathsf{V}_{1,1,t,1} \cdot \mathsf{U}^+_{i,j,k,t})_{1\leq t\leq n}$ is at least $\frac{\log 2}{2}$ on $[-1/2, 1/2]$, and, thus, it *contains* a $\frac{\log 2}{2}$-scaled $\mathrm{Uniform}([-1/2,1/2])$ (see Lemma H.1 in appendix H for details). In particular, setting $\mathrm{X} = \mathsf{V}_{1,1,t,1} \cdot \mathsf{U}^+_{i,j,k,t}$, we have that $\mu_- = \mathbb{E}[\mathbf{1}_{\mathrm{X}\leq 0}X] \leq -\frac{\log 2}{8} < -1/16$ and $\mu_+ = \mathbb{E}[\mathbf{1}_{\mathrm{X}>0}\mathrm{X}] \geq \frac{\log 2}{8} > 1/16$. Therefore, we can apply Theorem H.2 with $\xi = 1/32$: there exist constants $a, b > 0$ such that the expected value of the $[-n/32, n/32]$-subset-sum gap for $(\mathsf{V}_{1,1,t,1} \cdot \mathsf{U}^+_{i,j,k,t})_{1\leq t\leq n}$ is at most ae^{-bn}. That is,

$$\mathbb{E}\left[\max_{z\in[-n/32,n/32]} \min_{\mathcal{S}\subseteq[n]} |z - \sum_{t\in\mathcal{S}} \mathsf{V}_{1,1,t,1} \cdot \mathsf{U}^+_{i,j,k,t}| \right] \leq a e^{-bn}.$$

Assuming $n \geq 32$, Markov's inequality yields $\Pr[\overline{\mathcal{E}_{i,j,k,\varepsilon'}}] \leq \frac{ae^{-bn}}{\varepsilon'}$. Setting $\varepsilon' = \frac{\varepsilon}{d^2c}$ and $C = \frac{2}{b} + \frac{\log a}{b}$, and supposing without loss of generality that $\varepsilon' < e^{-1}$, the condition $n \geq C \log \frac{1}{\varepsilon'}$ implies $bn \geq 2\log\varepsilon' + \log a$ and $\frac{ae^{-bn}}{\varepsilon'} < \varepsilon'$, and we get

$$\Pr\left[\mathcal{E}_{i,j,k,\frac{\varepsilon}{d^2c}}\right] \geq 1 - \frac{\varepsilon}{d^2c}.$$

Now define the simultaneous event $\mathcal{E}_{\varepsilon'} = \bigcap_{i,j,k} \mathcal{E}_{i,j,k,\varepsilon'}$. By a union bound over the inequality above for $i, j \in [d]$, $k \in [c]$, we have

$$\Pr\left[\mathcal{E}_{\frac{\varepsilon}{d^2c}}\right] \geq 1 - \varepsilon.$$

Finally, conditioning on $\mathcal{E}_{\frac{\varepsilon}{d^2c}}$, it holds that

$$\begin{aligned}
&\sup_{\boldsymbol{K}\in[0,1]^{d\times d\times 1\times 1}} \inf_{\boldsymbol{S}\in\{0,1\}^{\mathrm{shape}(\mathbf{U})}} \sup_{\boldsymbol{X}\in[0,1]^{D\times D\times 1}} \|\boldsymbol{K} * \boldsymbol{X} - \mathbf{V} * \sigma[(\mathbf{U} \odot \boldsymbol{S}) * \boldsymbol{X}]\|_\infty \\
&\overset{(a)}{=} \sup_{\boldsymbol{K}} \inf_{\boldsymbol{S}} \sup_{\boldsymbol{X}} \|\boldsymbol{K} * \boldsymbol{X} - \mathbf{V} * (\mathbf{U}^+ \odot \boldsymbol{S}) * \boldsymbol{X}\|_\infty \\
&\overset{(b)}{=} \sup_{\boldsymbol{K}} \inf_{\boldsymbol{S}} \sup_{\boldsymbol{X}} \|[\boldsymbol{K} - \mathbf{V} * (\mathbf{U}^+ \odot \boldsymbol{S})] * \boldsymbol{X}\|_\infty \\
&\overset{(c)}{\leq} \sup_{\boldsymbol{K}} \inf_{\boldsymbol{S}} \sup_{\boldsymbol{X}} (\|\boldsymbol{K} - \mathbf{V} * (\mathbf{U}^+ \odot \boldsymbol{S})\|_1 \cdot \|\boldsymbol{X}\|_\infty) \\
&\overset{(d)}{\leq} \sup_{\boldsymbol{K}} \inf_{\boldsymbol{S}} \|\boldsymbol{K} - \mathbf{V} * (\mathbf{U}^+ \odot \boldsymbol{S})\|_1 \\
&\overset{(e)}{\leq} d^2c \cdot \sup_{\boldsymbol{K}} \inf_{\boldsymbol{S}} \|\boldsymbol{K} - \mathbf{V} * (\mathbf{U}^+ \odot \boldsymbol{S})\|_\infty \\
&\overset{(f)}{\leq} d^2c\frac{\varepsilon}{d^2c} = \varepsilon,
\end{aligned}$$

where (a) follows from equation (4.1), (b) from the distributivity of the convolution operation, (c) from proposition F.1, (d) from the fact that $\boldsymbol{X} \in [0,1]^{D\times D\times 1}$, (e) from the inequality $\|\boldsymbol{x}\|_1 \leq m\|\boldsymbol{x}\|_\infty$ for $\boldsymbol{x} \in \mathbb{R}^m$, and (f) from the definition of $\mathcal{E}_{\frac{\varepsilon}{cd^2}}$.

4.4.2 Convolutional Layer Approximation (Proof of Lemma 4.2.2)

The general goal of this argument is to choose binary masks $\boldsymbol{T}$ and $\boldsymbol{S}$ so that $(\mathbf{V}\odot\boldsymbol{T})*\sigma[(\mathbf{U}\odot\boldsymbol{S})*\boldsymbol{X}]$ is a sufficiently close approximation of $\boldsymbol{K} * \boldsymbol{X}$.

For $\ell \in [c_1]$ let $\boldsymbol{K}^{(\ell)}$ be $\boldsymbol{K}$'s ℓ-th kernel. That is,

$$\boldsymbol{K}^{(\ell)} = \boldsymbol{K}_{:,:,:,\ell}.$$

Notice that $\boldsymbol{K} * \boldsymbol{X}$ is the concatenation along the third dimension of each $\boldsymbol{K}^{(\ell)} * \boldsymbol{X}$, i.e., for $\ell \in [c_1]$, we have $(\boldsymbol{K} * \boldsymbol{X})_{:,:,\ell} = \boldsymbol{K}^{(\ell)} * \boldsymbol{X}$.

We fix $\boldsymbol{T}$ a priori to be the block diagonal matrix $\boldsymbol{B}$ with entries given by $B_{1,1,t,\ell} = \mathbf{1}_{(\ell-1)n' < t \leq \ell n'}$ for $t \in [n], \ell \in [c_1]$, where $n' = n/c_1$. In the rest of the proof, we show how to choose $\boldsymbol{S}$, based on $\mathbf{U}$ and $\mathbf{V}$, in order to approximate the kernels $\boldsymbol{K}^{(\ell)}$.

We perform the approximation of each $\boldsymbol{K}^{(\ell)}$ using different sections of the tensors. To this end, for $\ell \in [c_1]$, let

$$\mathbf{U}^{\ell} = \mathbf{U}_{:,:,:,(\ell-1)n'<t\leq \ell n'}, \qquad \boldsymbol{S}^{\ell} = \boldsymbol{S}_{:,:,:,(\ell-1)n'<t\leq \ell n'}, \quad \text{and} \quad \mathbf{V}^{\ell} = \mathbf{V}_{:,:,(\ell-1)n'<t\leq \ell n',:}.$$

As we did in the proof of Lemma 4.2.1, we perform an initial pruning on $\mathbf{U}$ by restricting $\boldsymbol{S}$ to the space of masks that prune all of its negative entries. This allows us to ignore the ReLU activation and conclude that

$$\begin{aligned}
&(\mathbf{V} \odot \boldsymbol{B}) * \sigma[(\mathbf{U} \odot \boldsymbol{S}) * \boldsymbol{X}])_{r,s,\ell} \\
&\quad = \sum_{(\ell-1)n'<t\leq \ell n'} \mathsf{V}_{1,1,t,\ell} \sum_{i,j\in[d],k\in[c]} (\mathbf{U} \odot \boldsymbol{S})_{i,j,k,t} \cdot \mathsf{X}_{r-i+1,s-j+1,k} \\
&\quad = \left(\mathbf{V}^{\ell} * [(\mathbf{U}^{\ell} \odot \boldsymbol{S}^{\ell}) * \boldsymbol{X}]\right)_{r,s} \\
&\quad = \left(\mathbf{V}^{\ell} * \sigma[(\mathbf{U}^{\ell} \odot \boldsymbol{S}^{\ell}) * \boldsymbol{X}]\right)_{r,s}.
\end{aligned}$$

For $\ell \in [c_1]$ and $\varepsilon' > 0$, denote by $\mathcal{E}_{\ell,\varepsilon'}$ the event

$$\left\{ \sup_{\boldsymbol{K}^{(\ell)}\in[-1,1]^{d\times d\times c_0\times 1}} \inf_{\boldsymbol{S}^{\ell}\in\{0,1\}^{\mathrm{shape}(\mathbf{U}^{\ell})}} \sup_{\boldsymbol{X}\in[0,1]^{D\times D\times c_0}} \left\| \boldsymbol{K}^{(\ell)} * \boldsymbol{X} - \mathbf{V}^{\ell} * \sigma[(\mathbf{U}^{\ell} \odot \boldsymbol{S}^{\ell}) * \boldsymbol{X}] \right\|_{\infty} < \varepsilon' \right\}.$$

Consider the event $\mathcal{E}_{\varepsilon/c_1} = \bigcap_{\ell} \mathcal{E}_{\ell,\varepsilon/c_1}$. Since $n' = n/c_1 = C \log \frac{d^2 c_0}{\varepsilon/c_1}$, for each $\ell \in [c_1]$, Lemma 4.2.1 ensures that $\Pr[\mathcal{E}_{\ell,\varepsilon/c_1}] \geq 1 - \varepsilon/c_1$, which implies that $\Pr[\mathcal{E}_{\varepsilon/c_1}] \geq 1 - \varepsilon$.

Finally, conditioning on $\mathcal{E}_{\varepsilon/c_1}$ and using the fact that the output channels of a convolutional layer are calculated independently, we conclude

$$\begin{aligned}
&\sup_{\boldsymbol{K}\in[-1,1]^{d\times d\times c_0\times c_1}} \inf_{\substack{\boldsymbol{S}\in\{0,1\}^{\mathrm{shape}(\mathbf{U})} \\ \boldsymbol{T}\in\{0,1\}^{\mathrm{shape}(\mathbf{V})}}} \sup_{\boldsymbol{X}\in[0,1]^{D\times D\times c_0}} \|\boldsymbol{K} * \boldsymbol{X} - (\mathbf{V} \odot \boldsymbol{T}) * \sigma[(\mathbf{U} \odot \boldsymbol{S}) * \boldsymbol{X}]\|_{\infty} \\
&\leq \sup_{\boldsymbol{K}\in[-1,1]^{d\times d\times c_0\times c_1}} \inf_{\boldsymbol{S}\in\{0,1\}^{\mathrm{shape}(\mathbf{U})}} \sup_{\boldsymbol{X}\in[0,1]^{D\times D\times c_0}} \|\boldsymbol{K} * \boldsymbol{X} - (\mathbf{V} \odot \boldsymbol{B}) * \sigma[(\mathbf{U} \odot \boldsymbol{S}) * \boldsymbol{X}]\|_{\infty} \\
&= \max_{\ell\in[c_1]} \sup_{\boldsymbol{K}^{(\ell)}\in[-1,1]^{d\times d\times c_0}} \inf_{\boldsymbol{S}^{\ell}\in\{0,1\}^{\mathrm{shape}(\mathbf{U}^{\ell})}} \sup_{\boldsymbol{X}\in[0,1]^{D\times D\times c_0}} \left\| \boldsymbol{K}^{(\ell)} * \boldsymbol{X} - \mathbf{V}^{\ell} * \sigma[(\mathbf{U}^{\ell} \odot \boldsymbol{S}^{\ell}) * \boldsymbol{X}] \right\|_{\infty} \\
&< \varepsilon.
\end{aligned}$$

CHAPTER 5

Structured pruning

The Strong Lottery Ticket Hypothesis (SLTH) states that randomly-initialised neural networks contain subnetworks that can perform well without any training. Although unstructured pruning has been extensively studied in this context, its structured counterpart, which can deliver significant computational and memory efficiency gains, has been largely unexplored. One of the main reasons for this gap is the limitations of the underlying mathematical tools used in formal analyses of the SLTH. In this chapter, we overcome these limitations: we leverage recent advances in the multidimensional generalisation of the Random Subset-Sum Problem and obtain a variant that admits the stochastic dependencies that arise when addressing structured pruning in the SLTH. We apply this result to prove, for a wide class of random Convolutional Neural Networks, the existence of structured subnetworks that can approximate any sufficiently smaller network.

This is the first work to address the SLTH for structured pruning, opening up new avenues for further research on the hypothesis and contributing to the understanding of the role of over-parameterization in deep learning.

5.1 Introduction

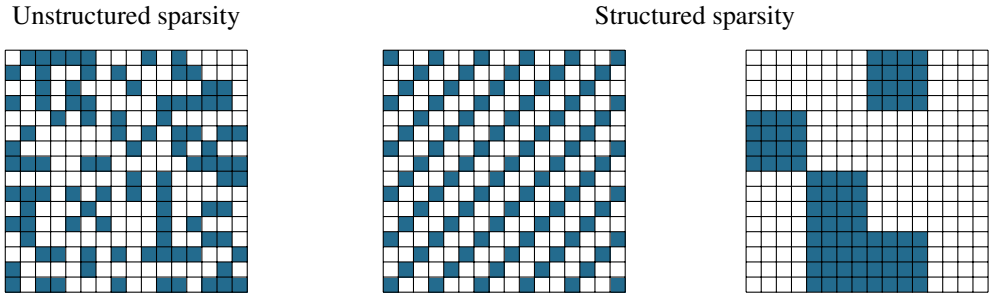
Much of the success of deep learning techniques relies on extreme over-parameterization. While such excess of parameters has allowed neural networks to become the state of the art in many tasks, the associated computational cost limits both the progress of those techniques and their deployment in real-world applications. This limitation motivated the development of methods for reducing the number of parameters of neural networks; both in the past (Reed, 1993) and in the present (Blalock et al., 2020; Hoefler et al., 2021).

Although pruning methods have traditionally targeted reducing the size of networks for inference purposes, recent works have indicated that they can also be used to reduce parameter counts during training or even at initialization without sacrificing model accuracy. In particular, Frankle and Carbin (2019) proposed the *Lottery Ticket Hypothesis (LTH)*, which conjectures that randomly initialised networks contain sparse subnetworks that can be trained and reach the performance of the fully-trained original network. Empirical investigations on the LTH (Zhou et al., 2019; Ramanujan et al., 2020; Y. Wang et al., 2020) pointed towards an even more impressive phenomenon: the existence of subnetworks that perform well without any training. This conjecture was named the *Strong Lottery Ticket Hypothesis (SLTH)* by Pensia et al. (2020).

While the SLTH has been proved for many different classes of neural networks (see section 5.2), those works are restricted to unstructured pruning, where the subnetworks are obtained by freely removing individual parameters from the original network. However, this lack of structure can significantly reduce the main gains of pruning, both in terms of memory and computational efficiency. Removing parameters at arbitrary points of the network implies the need to store the indices of the remaining non-zero parameters, which can become a significant overhead with its own research challenges (Pooch & Nieder, 1973). Moreover, the theoretical computational gains of unstructured sparsity can also be difficult to realize in standard hardware, which is optimized for dense operations. Most notably, the irregularity of the memory access patterns can lead to both data and instruction cache misses, significantly reducing the performance of the pruned network.

The limitations of parameter-level pruning have motivated extensive research on *structured pruning*, which constrain the sparsity patterns to reduce the complexity of parameter indexation and, more generally, to make the processing of the pruned network more efficient. A simple example of structured pruning is *neuron pruning* of fully-connected layers: deletions in the weight matrix are constrained to the level of whole rows/columns. With this constraint, pruning results in a smaller dense network, directly reducing the computational costs without any need for extra memory to store indices. Similarly, deleting entire filters in Convolutional Neural Networks (CNNs) (Polyak & Wolf, 2015) or "heads" in attention-based architectures (Michel, Levy, & Neubig, 2019) also produces direct reductions in computational costs.

It is important to note that structured pruning is a restriction of unstructured pruning so, theoretically, the former is bound to perform at most as well as the latter. For example, by deleting whole neurons one can remove about 70% of the weights in dense networks without significantly affecting its performance. Through unstructured pruning, on the other hand, one can usually reach 95% sparsity without accuracy loss (Alvarez & Salzmann, 2016). In practice, however, the computational advantage of structured pruning can offset this difference. This trade-off between sparsity and actual efficiency has motivated the study of less coarse sparsity patterns. Weaker structural constraints such as strided sparsity (Anwar et al., 2017) (figure 5.1b) or block sparsity



(a) No pattern. (b) Strided pattern. (c) Block pattern.

Figure 5.1: Examples of different pruning patterns.

(Siswanto, 2021) (figure 5.1b) are already sufficient to deliver the bulk of the computational gains that structured can offer.

Despite its benefits, there have been no results on structured pruning in the context of the SLTH. We believe this gap can be attributed to the limitations of a central result underlying almost all of the theoretical works on the SLTH: a theorem by Lueker on the *Random Subset-Sum Problem (RSSP)*.

Theorem 5.1.1 ((Lueker, 1998; da Cunha, d'Amore, et al., 2022)). *Let $X_1, \dots, X_n$ be independent uniform random variables over $[-1, 1]$, and let $\varepsilon \in (0, 1/3)$. There exists a universal constant $C > 0$ such that, if $n \geq C \log(1/\varepsilon)$, then, with probability at least $1 - \varepsilon$, for all $z \in [-1, 1]$ there exists $S_z \subseteq [n]$ for which*

$$\left| z - \sum_{i \in S_z} X_i \right| \leq \varepsilon.$$

In general terms, the theorem states that given a rather small number of random variables, there is a high probability that any target value (within an interval of interest) can be approximated as a sum of a subset of the random variables. An important remark is that even though Theorem 5.1.1 is stated in terms of uniform random variables, it is not hard to extend it to a wide class of distributions.[1]

While Theorem 5.1.1 closely matches the setup of the SLTH, it only concerns individual random variables, so it does not apply to entire random structures directly. Borst, Dadush, Huiberts, and Tiwari (2023); Becchetti et al. (2022) reduced this gap by proving multidimensional versions of Theorem 5.1.1. Still, the intricate manipulation of the network parameters in proofs around the SLTH imposes restrictions that are not covered by those results.

Contributions

In this chapter, we close this gap by providing a version of Theorem 5.1.1 that allows us to prove that networks in a wide class of CNNs are likely to contain structured subnetworks that approximate any sufficiently smaller CNN in the class. To the best of our knowledge, this is the first result around the SLTH for structured pruning of neural networks of any kind. More precisely,

[1] Distributions whose probability density function φ satisfies $\varphi(x) \geq b$ for all $x \in [-a, a]$, for some constants $a, b > 0$ (see Lueker (1998, Corollary 3.3)).

We prove a multidimensional version Theorem 5.1.1 that is robust to some dependencies between coordinates, which is crucial for structured pruning (Theorem 5.3.2);

We use this result to show that, with high probability, a rather wide class of random CNNs can be pruned (in a structured manner) to approximate any sufficiently smaller CNN in this class (Theorem 5.3.1);

Additionally, our pruning scheme focuses on filter pruning, which, like neuron pruning, allows for a direct reduction of the size of the original CNN.

5.2 Related Work

SLTH Put roughly, research on the SLTH revolves around the following question:

Question – Given an error margin $\varepsilon > 0$ and a target network architecture f_{target}, how large must an architecture f_{random} be to ensure that, with high probability on the sampling of parameters of f_{random}, one can prune f_{random} to obtain a subnetwork that approximates f_{target} up to output error ε?

Malach et al. (2020) first proved that, for dense networks with ReLU activations, it was sufficient for f_{random} to be twice as deep and polynomially wider than f_{target}. Orseau et al. (2020) showed that the width overhead could be greatly reduced by sampling parameters from a hyperbolic distribution. Pensia et al. (2020) improved the original result for a wide class of weight distribution, requiring only a logarithmic width overhead, which they proved to be asymptotically optimal. da Cunha, Natale, and Viennot (2022) generalised those results with optimal bounds to CNNs with non-negative inputs, which Burkholz (2022a) extended to general inputs and to residual architectures. Burkholz (2022a) also reduced the depth overhead to a single extra layer and provided results that include a whole class of activation functions. Burkholz (2022b) obtained similar improvements to dense architectures. Fischer and Burkholz (2021) modified many of the previous arguments to take into consideration networks with non-zero biases. Ferbach et al. (2022) further generalise previous results on CNNs to general equivariant networks. Diffenderfer and Kailkhura (2021) obtained similar SLTH results for binary dense neural networks within polynomial depth and width overhead, which Sreenivasan, Rajput, Sohn, and Papailiopoulos (2022) improved to logarithmic overhead.

Structured pruning Works on structured pruning date back to the early days of the field of neural network sparsification with works such as M. Mozer and Smolensky (1988) and M. C. Mozer and Smolensky (1989). Since then, a vast literature was built around the topic, particularly for the pruning of CNNs. For a survey of structured pruning in general, we refer the reader to the associated sections of Hoefler et al. (2021), and to Y. He and Xiao (2023) for a survey on structured pruning of CNNs.

RSSP Pensia et al. (2020) introduced the use of theoretical results on the RSSP in arguments around the SLTH, namely Lueker (1998, Corollary 3.3). da Cunha, d'Amore, et al. (2022) provides an alternative, simpler proof of this result. Borst, Dadush, Huiberts, and Tiwari (2023) and Becchetti et al. (2022) prove multidimensional versions of the theorem. Theorem 5.3.2 diverges from those results in that it supports some dependencies between the entries of random vectors.

5.3 Preliminaries and contribution

Given $n \in \mathbb{N}$, we denote the set $\{1, \dots, n\}$ by $[n]$. The symbol $*$ represents the convolution operation, $\odot$ represents the element-wise (Hadamard) product, and relu represents the ReLU activation function. The notation $\|\cdot\|_1$ refers to the sum of the absolute values of each entry in a tensor. Similarly, $\|\cdot\|_2$ refers to the square root of the sum of the squares of each entry in a tensor. $\|\cdot\|_\infty$ denotes the maximum norm: the maximum among the absolute value of each entry. Sometimes we represent a tensor $\boldsymbol{X} \in \mathbb{R}^{d_1 \times \cdots \times d_n}$ by the notation $(\boldsymbol{X}_{i_1,\dots,i_n})_{i_1 \in [d_1],\dots,i_n \in [d_n]}$. We denote the normal probability distribution with mean μ and variance σ^2 by $\mathcal{N}\,\mu, \sigma^2$. We write $\mathbf{U} \sim \mathcal{N}^{d_1 \times \cdots \times d_n}$ to denote that $\mathbf{U}$ is a random tensor of size $d_1 \times \cdots \times d_n$ with entries independent and identically distributed (i.i.d.), each following $\mathcal{N}\,0, 1$. We refer to such random tensors as *normal tensors*. Finally, we refer to the axis of a 4-D tensor as *rows*, *columns*, *channels*, and *kernels* (a.k.a., filters), in this order.

For the sake of simplicity, we assume CNNs to be of the form $N\colon [-1,1]^{D \times D \times c_0} \to \mathbb{R}^{D \times D \times c_\ell}$ given by

$$N(\boldsymbol{X}) = \boldsymbol{K}^{(\ell)} * \mathrm{relu}(\boldsymbol{K}^{(\ell-1)} * \cdots \mathrm{relu}(\boldsymbol{K}^{(1)} * \boldsymbol{X})),$$

where $\boldsymbol{K}^{(i)} \in \mathbb{R}^{d_i \times d_i \times c_{i-1} \times c_i}$ for $i \in [\ell]$, and the convolutions have no bias and are suitably padded with zeros. Moreover, when the kernels $\boldsymbol{K}^{(i)}$ are normal tensors, we say that N is a *random CNN*.

Before we proceed to our main theorem, we introduce a definition that encompasses the sparsity structure underlying our proofs.

Definition 5.3.1 (n-channel-blocked mask)**.** Given a positive integer n, a binary tensor $\boldsymbol{S} \in \{0,1\}^{d \times d \times c \times cn}$ is called n-channel-blocked if and only if

$$\boldsymbol{S}_{i,j,k,l} = \begin{cases} 1 & \text{if } \left\lceil \frac{l}{n} \right\rceil = k, \\ 0 & \text{otherwise,} \end{cases}$$

for all $i, j \in [d]$, $k \in [c]$, and $l \in [cn]$.

Theorem 5.3.1 (SLTH for kernel pruning)**.** *Let D, d, c_0, c_1 and ℓ be positive integers and let ε and C be positive real numbers. For each $i \in [\ell]$, let $\boldsymbol{L}^{(2i-1)} \sim \mathcal{N}^{1 \times 1 \times c_{i-1} \times 2c_{i-1}n_i}$ and $\boldsymbol{L}^{(2i)} \sim \mathcal{N}^{d_i \times d_i \times 2c_{i-1}n_i \times c_i}$ with $n_i \geq C d^{12} c_i^6 \log^3 \frac{d^2 c_i c_{i-1} \ell}{\varepsilon}$ for some positive integers n_i and c_i. Let then $N_0\colon [-1,1]^{D \times D \times c_0} \to^{D \times D \times c_\ell}$ be a random CNN of the form*

$$N_0(X) = \boldsymbol{L}^{(2\ell)} * \cdots \mathrm{relu}(\boldsymbol{L}^{(1)} * \boldsymbol{X}).$$

Given $2n_i$-channel-blocked masks $\boldsymbol{S}^{(2i-1)} \in \{0,1\}^{1 \times 1 \times n_i \times c_i}$ for each tensor $\boldsymbol{L}^{(2i-1)}$, for $i \in [\ell]$; let

$$N_0^{(\boldsymbol{S}^{(1)},\dots,\boldsymbol{S}^{(2\ell-1)})} = \boldsymbol{L}^{(2\ell)} * \mathrm{relu}(\cdots(\boldsymbol{L}^{(2)} * \mathrm{relu}((\boldsymbol{S}^{(1)} \odot \boldsymbol{L}^{(1)}) * X))).$$

Finally, let $\boldsymbol{\mathcal{F}}$ be the class of functions $f\colon [-1,1]^{D \times D \times c_0} \to \mathbb{R}^{D \times D \times c_\ell}$ of the form

$$f(\boldsymbol{X}) = \boldsymbol{K}^{(\ell)} * \cdots \mathrm{relu}(\boldsymbol{K}^{(1)} * \boldsymbol{X}),$$

where $\boldsymbol{K}^{(i)} \in \mathbb{R}^{d_i \times d_i \times c_{i-1} \times c_i}$ with $\|\boldsymbol{K}^{(i)}\|_1 \leq 1$, for $i \in [\ell]$.

There exists a universal value of C such that, with probability $1 - \varepsilon$, for every $f \in \boldsymbol{\mathcal{F}}$ it is possible to remove filters from $N_0^{(\boldsymbol{S}^{(1)},\dots,\boldsymbol{S}^{(2\ell-1)})}$ to obtain a CNN $\tilde{N}_0$ for which

$$\sup_{\boldsymbol{X} \in [-1,1]^{D \times D \times c_0}} \|f(\boldsymbol{X}) - \tilde{N}_0(\boldsymbol{X})\|_\infty \leq \varepsilon.$$

The filter removals ensured by Theorem 5.3.1 take place at layers $1, 3, \ldots, 2\ell - 1$ and imply the removal of the corresponding channels in the next layer. The overall modification yields a CNN with kernels $\tilde{\boldsymbol{L}}^{(1)}, \ldots, \tilde{\boldsymbol{L}}^{(2\ell)}$ such that, for $i \in [\ell]$, $\tilde{\boldsymbol{L}}^{(2i-1)} \in \mathbb{R}^{1\times 1\times c_{i-1}\times 2c_{i-1}m_i}$ and $\tilde{\boldsymbol{L}}^{(2i)} \in \mathbb{R}^{d_i\times d_i\times 2c_{i-1}m_i\times c_i}$, where $m_i = \sqrt{n_i/(C_1 \log \frac{1}{\varepsilon})}$ for a universal constant C_1. Moreover, the kernels $\tilde{\boldsymbol{L}}^{(2i-1)}$ are structured as if pruned by $2m_i$-channel-blocked masks.

We remark that, from a broader perspective, the central aspect of Theorem 5.3.1 is that the lower bound on the size of the random CNN depends only on the kernel sizes of the CNNs being approximated.

In section 5.4.2 we discuss the proof of Theorem 5.3.1. It requires handling subset-sum problems on multiple random variables at once (random vectors). Furthermore, the inherent parameter-sharing of CNNs creates a specific type of stochastic dependency between coordinates of the random vectors, which we capture with the following definition.

Definition 5.3.2 (NSN vector)**.** A d-dimensional random vector $\mathbf{Y}$ follows a *normally-scaled normal* (NSN) distribution if, for each $i \in [d]$, $\mathrm{Y}_i = \mathrm{Z} \cdot \mathrm{Z}^{(i)}$ where $\mathrm{Z}, \mathrm{Z}^{(1)}, \ldots, \mathrm{Z}^{(d)}$ are i.i.d. random variables following a standard normal distribution.

A key technical contribution of ours is a Multidimensional Random Subset Sum (MRSS) result that supports NSN vectors. In section 5.4.1 we discuss the proof of the next theorem, which follows a strategy similar to that of (Borst, Dadush, Huiberts, & Tiwari, 2023, Lemmas 1, 15).

Theorem 5.3.2 (Normally-scaled MRSS)**.** *Let $0 < \varepsilon \leq 1/4$, and let d, k, and n be positive integers such that $n \geq k^2$ and $k \geq Cd^3 \log \frac{d}{\varepsilon}$ for some universal constant $C \in \mathbb{R}_{>0}$. Furthermore, let $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(n)}$ be d-dimensional i.i.d. NSN random vectors. For any $\boldsymbol{z} \in \mathbb{R}^d$ with $\|\boldsymbol{z}\|_1 \leq \sqrt{k}$, there exists with constant probability a subset $\mathcal{S} \subseteq [n]$ of size k such that $\|(\sum_{i\in\mathcal{S}} \mathbf{X}^{(i)}) - \boldsymbol{z}\|_\infty \leq \varepsilon$.*

While it is possible to naïvely apply Theorem 5.1.1 to obtain a version of Theorem 5.3.1, doing so would lead to an exponential lower bound on the required number of random vectors.

5.4 Analysis

In this section, after proving our MRSS result (Theorem 5.3.2), we discuss how to use it to obtain our main result on structured pruning (Theorem 5.3.1). Full proofs are deferred to the supplementary material (SM).

5.4.1 Multidimensional Random Subset Sum for normally-scaled normal vectors

Notation. Given a set $\mathcal{S}$ and a positive integer n, the notation $\binom{\mathcal{S}}{n}$ denotes the family of subsets of $\mathcal{S}$ containing exactly n elements of $\mathcal{S}$. Given $\varepsilon \in \mathbb{R}_{>0}$, we define the interval $I_\varepsilon(z) = [z-\varepsilon, z+\varepsilon]$ and the multi-interval $I_\varepsilon(\boldsymbol{z}) = [\boldsymbol{z} - \varepsilon\mathbf{1}, \boldsymbol{z} + \varepsilon\mathbf{1}]$, where $\mathbf{1} = (1, 1, \ldots, 1) \in \mathbb{R}^d$. Moreover, for any event $\mathcal{E}$, we denote its complementary event by $\overline{\mathcal{E}}$.

In this subsection, we estimate the probability that a set of n random vectors contains a subset that sums up to a value that is ε-close to a given target. The following definition formalizes this notion.

Definition 5.4.1 (Subset-sum number)**.** Given (possibly random) vectors $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(n)}$ and a vector $\boldsymbol{z}$, we define the *ε-subset-sum number* of $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(n)}$ for $\boldsymbol{z}$ as

$$\mathrm{T}^{k}_{\mathbf{X}^{(1)},\dots,\mathbf{X}^{(n)}}(\boldsymbol{z}) = \sum_{\mathcal{S}\in\binom{[n]}{k}} \mathbf{1}_{\mathcal{E}^{(\boldsymbol{z})}_{\mathcal{S}}},$$

where $\mathcal{E}^{(\boldsymbol{z})}_{\mathcal{S}}$ denotes the event $\|(\sum_{i\in\mathcal{S}} \mathbf{X}^{(i)}) - \boldsymbol{z}\|_\infty \leq \varepsilon$. We write simply $\mathrm{T}_{n,k}$ when $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(n)}$ and $\boldsymbol{z}$ are clear from the context.

To prove Theorem 5.3.2 we use the second moment method to provide a lower bound on the probability that the subset-sum number $\mathrm{T}_{n,k}$ is strictly positive, which implies that at least one subset of the random vectors can approximate the target value $\boldsymbol{z}$. Hence, we seek a lower bound on $\mathbb{E}[\mathrm{T}_{n,k}]^2/\mathbb{E}[\mathrm{T}^2_{n,k}]$.

Our first lemma provides a lower bound on the probability that a sum of NSN vectors is ε-close to a target vector, through which one can infer a lower bound on $\mathbb{E}[\mathrm{T}_{n,k}]$.

Lemma 5.4.1 (Sum of NSN vectors)**.** *Let $k \in \mathbb{N}$, $\varepsilon \in (0, 1/4)$, $\boldsymbol{z} \in \mathbb{R}^d$ such that $\|\boldsymbol{z}\|_1 \leq \sqrt{k}$ and $k \geq 16$. Furthermore, let $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(k)}$ be d-dimensional i.i.d. NSN random vectors with $d \leq k$, and let $c_d = \min\{\frac{1}{d^2}, \frac{1}{16}\}$. It holds that*

$$\Pr\left[\sum_{i=1}^{k} \mathbf{X}^{(i)} \in I_\varepsilon(\boldsymbol{z})\right] \geq \frac{1}{16}\left(\frac{2\varepsilon}{\sqrt{\pi\left(1 + 2\sqrt{c_d} + 2c_d\right)k}}\right)^d.$$

Overview of the proof. The main technical difficulty lies in the fact that the random vectors $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(k)}$ are NSN vectors. Bounds can be easily derived for the case where the $\mathbf{X}^{(i)}$ are i.i.d. normal random vectors by observing that the sum of normal random variables is also normal.

For $i \in [k]$, each entry of $\mathbf{X}^{(i)}$ can be written as $\mathrm{Z}^{(i)} \cdot \mathrm{Z}i, j$ where $\mathrm{Z}^{(i)}$ and $\mathrm{Z}^{(i,j)}$ are i.i.d. normal random variables. Conditional on $\mathrm{Z}(1), \dots, \mathrm{Z}^{(k)}$, the d entries of $\mathbf{X} = \sum_{i=1}^{k} \mathbf{X}^{(i)}$ are independent and distributed as $\mathcal{N}(0, \sum_{i=1}^{k} Z_i^2)$. By noticing that $(\mathrm{Z}^{(i)})^2$ is a chi-squared random variable and employing standard concentration inequalities (Lemma I.4 in appendix I) combined with the law of total probability, we can proceed as if the entries of $\mathbf{X}$ were normal, up to some correction factors. □

Bounding $\mathbb{E}[(\mathrm{T}^{(n,k)})^2]$ requires handling stochastic dependencies. Thus, we estimate the joint probability that two subsets of k elements of $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(n)}$ sum ε-close to the same target, taking into account that the intersection of the subsets might not be empty. The next lemma provides an upper bound on this joint probability that depends only on the size of the symmetric difference between the two subsets.

Lemma 5.4.2 (Sum of NSN vectors)**.** *Let $k, j \in \mathbb{N}_0$ with $1 \leq j \leq k$ Furthermore, let $\mathbf{X}^{(1)}, \dots, \mathbf{X}(k+j)$ be i.i.d. d-dimensional NSN random vectors with $k \geq Cd^3 \log \frac{d}{\varepsilon}$. Let $c_d = \min[\frac{1}{d^2}, \frac{1}{16}]$, $\mathbf{A} = \sum_{i=1}^{j} \mathbf{X}^{(i)}$, $\mathbf{B} = \sum_{i=j+1}^{k} \mathbf{X}(i)$, and $\mathbf{C} = \sum_{i=k+1}^{k+j} \mathbf{X}(i)$.*[2] *Then, it holds that*

$$\Pr[\mathbf{A} + \mathbf{B} \in I_\varepsilon(\boldsymbol{z}), \mathbf{B} + \mathbf{C} \in I_\varepsilon(\boldsymbol{z})] \leq 3\left(\frac{4\varepsilon^2}{\pi\left(1 - 2\sqrt{c_d}\right)j}\right)^d.$$

[2] We adopt the convention that $\sum_{i=1}^{0} \mathbf{X}^{(i)} = 0$.

Overview of the proof. We exploit once more the fact that, for all $i \in [k]$, each entry $\mathbf{X}^{(i)}$ can be written as $\mathrm{Z}^{(i)} \cdot \mathrm{Z}^{(i,j)}$ where $\mathrm{Z}^{(i)}$ and $\mathrm{Z}^{(i,j)}$ are i.i.d. normal random variables. Conditional on $\mathrm{Z}^{(1)}, \dots, \mathrm{Z}^{(k)}$, the d entries of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are independent and distributed as $\mathcal{N}(0, \sum_{i=1}^{j} (\mathrm{Z}^{(i)})^2)$, $\mathcal{N}(0, \sum_{i=j+1}^{k} (\mathrm{Z}^{(i)})^2)$, and $\mathcal{N}(0, \sum_{i=k+1}^{k+j} (\mathrm{Z}^{(i)})^2)$, respectively. Hence, by the concentration inequalities for the sum of chi-squared random variables (Lemma I.4 in appendix I) and by the law of total probability, we can focus on the term

$$\Pr\Big[\mathrm{A}_i + \mathrm{B}_i \in I_\varepsilon(z_i), \mathrm{B}_i + \mathrm{C}_i \in I_\varepsilon(z_i) \;\Big|\; \mathrm{Z}^{(1)}, \dots, \mathrm{Z}^{(n)}\Big],$$

where A_i, B_i, and C_i indicate the i-th entries of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, respectively.

Another concentration argument for normal random variables (Lemma I.1 in appendix I), allow us to show that

$$\begin{aligned}
&\Pr\Big[\mathrm{A}_i + \mathrm{B}_i \in I_\varepsilon(\mathrm{Z}^{(i)}), \mathrm{B}_i + \mathrm{C}_i \in I_\varepsilon(z_i) \;\Big|\; \mathrm{Z}^{(1)}, \dots \mathrm{Z}^{(n)}\Big] \\
&\quad = \mathbb{E}_{\mathrm{B}_i}\Big[\Pr\Big[\mathrm{A}_i \in I_\varepsilon(z_i - \mathrm{B}_i), \mathrm{C}_i \in I_\varepsilon(z_i - \mathrm{B}_i) \;\Big|\; \mathrm{Z}^{(1)}, \dots, \mathrm{Z}^{(n)}, \mathrm{B}_i\Big]\Big] \\
&\quad = \mathbb{E}_{\mathrm{B}_i}\Big[\Pr\Big[\mathrm{A}_i \in I_\varepsilon(z_i - \mathrm{B}_i) \;\Big|\; \mathrm{Z}^{(1)}, \dots, \mathrm{Z}^{(n)}, \mathrm{B}_i\Big] \Pr\Big[\mathrm{C}_i \in I_\varepsilon(z_i - \mathrm{B}_i) \;\Big|\; \mathrm{Z}^{(1)}, \dots, \mathrm{Z}^{(n)}, \mathrm{B}_i\Big]\Big] \\
&\quad \le \mathbb{E}_{\mathrm{B}_i}\Big[\Pr\Big[\mathrm{A}_i \in I_\varepsilon(0) \;\Big|\; \mathrm{Z}^{(1)}, \dots, \mathrm{Z}^{(n)}, \mathrm{B}_i\Big] \Pr\Big[\mathrm{C}_i \in I_\varepsilon(0) \;\Big|\; \mathrm{Z}^{(1)}, \dots, \mathrm{Z}^{(n)}, \mathrm{B}_i\Big]\Big] \\
&\quad = \Pr\Big[\mathrm{A}_i \in I_\varepsilon(0) \;\Big|\; \mathrm{Z}^{1}, \dots, \mathrm{Z}^{(n)}\Big] \Pr\Big[\mathrm{C}_i \in I_\varepsilon(0) \;\Big|\; \mathrm{Z}^{(1)}, \dots, \mathrm{Z}^{(n)}\Big].
\end{aligned}$$

Thus, we have reduced our argument to the estimation of probabilities of independent normal random variables being close to zero. □

The following lemma provides an explicit expression for the variance of the ε-subset-sum number.

Lemma 5.4.3 (Second moment of $\mathrm{T}_{n,k}$)**.** *Let k, n be positive integers. Let $\mathcal{S}_0, \mathcal{S}_1, \dots, \mathcal{S}_k$ be subsets of $[n]$ such that $|\mathcal{S}_0 \cap \mathcal{S}_j| = k - j$ for $j = 0, 1, \dots, k$. Let $\mathcal{S}, \mathcal{S}'$ be two random variables yielding two subsets of $[n]$ drawn independently and uniformly at random. Let $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(n)}$ be d-dimensional i.i.d. NSN random vectors. For any $\varepsilon > 0$ and $\boldsymbol{z} \in \mathbb{R}^d$, the second moment of the ε-subset sum number is*

$$\mathbb{E}[\mathrm{T}_{n,k}^2] = \binom{n}{k}^2 \sum_{j=0}^{k} \Pr[|\mathcal{S} \cap \mathcal{S}'| = k - j] \Pr\Big[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})} \cap \mathcal{E}_{\mathcal{S}_j}^{(\boldsymbol{v})}\Big],$$

where $\mathcal{E}_{\mathcal{S}}^{(\boldsymbol{z})}$ denotes the event $\|(\sum_{i \in \mathcal{S}} \mathbf{X}^{(i)}) - \boldsymbol{z}\|_\infty \le \varepsilon$.

Proof. Let $\mathcal{S}$ and $\mathcal{S}'$ be random variables yielding elements of $\binom{[n]}{k}$ drawn independently and uniformly at random. By the definition of $\mathrm{T}_{n,k}$, we have that

$$\begin{aligned}
\mathbb{E}[\mathrm{T}_{n,k}^2] &= \mathbb{E}\left[\left(\sum_{\mathcal{S}\in\binom{[n]}{k}} \mathbf{1}_{\mathcal{E}_{\mathcal{S}}^{(\boldsymbol{z})}}\right)\left(\sum_{\mathcal{S}'\in\binom{[n]}{k}} \mathbf{1}_{\mathcal{E}_{\mathcal{S}'}^{(\boldsymbol{v})}}\right)\right] \\
&= \mathbb{E}\left[\sum_{\mathcal{S},\mathcal{S}'\in\binom{[n]}{k}} \mathbf{1}_{\mathcal{E}_{\mathcal{S}}^{(\boldsymbol{z})}}\mathbf{1}_{\mathcal{E}_{\mathcal{S}'}^{(\boldsymbol{v})}}\right] \\
&= \sum_{\mathcal{S},\mathcal{S}'\in\binom{[n]}{k}} \Pr\left[\mathcal{E}_{\mathcal{S}}^{(\boldsymbol{z})} \cap \mathcal{E}_{\mathcal{S}'}^{(\boldsymbol{v})}\right] \\
&= \sum_{\mathcal{S},\mathcal{S}'\in\binom{[n]}{k}} \Pr\left[\mathcal{E}_{\mathcal{S}}^{(\boldsymbol{v})} \cap \mathcal{E}_{\mathcal{S}'}^{(\boldsymbol{v})} \,\middle|\, \mathcal{S}=S,\mathcal{S}'=\mathcal{S}'\right]\Pr[\mathcal{S}=S,\mathcal{S}'=\mathcal{S}'] \\
&= \binom{n}{k}^2 \sum_{j=0}^{k} \Pr\left[\mathcal{E}_{\mathcal{S}}^{(\boldsymbol{v})} \cap \mathcal{E}_{\mathcal{S}'}^{(\boldsymbol{v})} \,\middle|\, |\mathcal{S}\cap\mathcal{S}'|=k-j\right]\Pr[|\mathcal{S}\cap\mathcal{S}'|=k-j],
\end{aligned}$$

as $\Pr[\mathcal{E}_{\mathcal{S}}^{(\boldsymbol{v})} \cap \mathcal{E}_{\mathcal{S}'}^{(\boldsymbol{v})}]$ depends only on the size of $\mathcal{S}\cap\mathcal{S}'$. □

Overview of the proof of Theorem 5.3.2

We use the second moment method (Lemma I.2 in appendix I) on the ε-subset-sum number $\mathrm{T}_{n,k}$ of $\mathbf{X}^{(1)},\dots,\mathbf{X}^{(n)}$. Thus, we want to lower bound the right-hand side of

$$\Pr[\mathrm{T}>0] \geq \frac{\mathbb{E}[\mathrm{T}_{n,k}]^2}{\mathbb{E}[\mathrm{T}_{n,k}^2]}.$$

Equivalently, we can provide an upper bound on the inverse, $\frac{\mathbb{E}[\mathrm{T}_{n,k}^2]}{\mathbb{E}[\mathrm{T}_{n,k}]^2}$. By Lemma 5.4.3,

$$\mathbb{E}[\mathrm{T}_{n,k}^2] = \binom{n}{k}^2 \sum_{j=0}^{k} \Pr[|\mathcal{S}\cap\mathcal{S}'|=k-j]\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})} \cap \mathcal{E}_{\mathcal{S}_j}^{(\boldsymbol{v})}\right]$$

where $\mathcal{S},\mathcal{S}',\mathcal{S}_i$ and $\mathcal{E}_{\mathcal{S}}^{(\boldsymbol{z})}$ are defined as in the statement of the lemma. Observe also that

$$\mathbb{E}[T_{n,k}] = \sum_{\mathcal{S}\in\binom{[n]}{k}} \mathbb{E}[\mathbf{1}_{\mathcal{E}_{\mathcal{S}}^{(\boldsymbol{z})}}] = \sum_{\mathcal{S}\in\binom{[n]}{k}} \Pr\left[\mathcal{E}_{\mathcal{S}}^{(\boldsymbol{z})}\right] = \binom{n}{k}\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})}\right].$$

By using the two above observations, we have

$$\begin{aligned}
\frac{\mathbb{E}[\mathrm{T}_{n,k}]^2}{\mathbb{E}[\mathrm{T}_{n,k}^2]} &= \frac{\binom{n}{k}^2}{\mathbb{E}[\mathrm{T}_{n,k}]^2} \sum_{j=0}^{k} \Pr[|\mathcal{S}\cap\mathcal{S}'|=k-j]\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})} \cap \mathcal{E}_{\mathcal{S}_j}^{(\boldsymbol{v})}\right] \\
&= \sum_{j=0}^{k} \Pr[|\mathcal{S}\cap\mathcal{S}'|=k-j] \frac{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})} \cap \mathcal{E}_{\mathcal{S}_j}^{(\boldsymbol{v})}\right]}{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})}\right]^2}.
\end{aligned}$$

Lemma 5.4.1 provides a lower bound on the term $\Pr[\mathcal{E}^{(\boldsymbol{v})}_{\mathcal{S}_0}]$ while Lemma 5.4.2 gives an upper bound on the term $\Pr[\mathcal{E}^{(\boldsymbol{v})}_{\mathcal{S}_0} \cap \mathcal{E}^{(\boldsymbol{v})}_{\mathcal{S}_j}]$.

In the full proof, we then show that $\Pr[|\mathcal{S} \cap \mathcal{S}'| \geq k/d]$ can be bounded using the Chernoff bound (Lemma I.3 in appendix I) even if we do not deal directly with Binomial random variables. This allows us to discard the indices j for which $\Pr[\mathcal{E}^{(\boldsymbol{v})}_{\mathcal{S}_0} \cap \mathcal{E}^{(\boldsymbol{v})}_{\mathcal{S}_j}]$ is large, which leads to the result after some technical manipulations.

5.4.2 Proving SLTH for structured pruning

To prove Theorem 5.3.1, we first show how to obtain the same approximation result for a single-layer CNN. Then, we iteratively apply the same argument for all layers of a larger CNN and show that the approximation error keeps small.

We define the *positive* and *negative* parts of a tensor.

Definition 5.4.2. Given a tensor $\boldsymbol{X} \in \mathbb{R}^{d_1 \times \cdots \times d_n}$, the *positive* and *negative* parts of $\boldsymbol{X}$ are respectively defined as $\boldsymbol{X}^+_{\vec{i}} = \boldsymbol{X}_{\vec{i}} \cdot \mathbf{1}_{\boldsymbol{X}_{\vec{i}} > 0}$ and $\boldsymbol{X}^-_{\vec{i}} = -\boldsymbol{X}_{\vec{i}} \cdot \mathbf{1}_{\boldsymbol{X}_{\vec{i}} < 0}$, where $\vec{i} \in [d_1] \times \cdots \times [d_n]$ points at a generic entry of $\boldsymbol{X}$.

Approximating a single-layer CNN

We first present a preliminary lemma that shows how to prune a single-layer convolution $\mathrm{relu}(\boldsymbol{V} * \boldsymbol{X})$ in a way that dispenses us from dealing with the ReLU relu.

Lemma 5.4.4. *Let $D, d, c, n \in \mathbb{N}$ be positive integers, $\boldsymbol{V} \in \mathbb{R}^{1\times 1\times c\times 2nc}$, and $\boldsymbol{X} \in \mathbb{R}^{D\times D\times c}$. If $\boldsymbol{S} \in \{0,1\}^{\mathrm{shape}(\boldsymbol{V})}$ is a $2n$-channel blocked mask, then, for each $(i,j,k) \in [D] \times [D] \times [2nc]$,*

$$\Big(\mathrm{relu}((\boldsymbol{V} \odot \boldsymbol{S}) * \boldsymbol{X})\Big)_{i,j,k} = \Big((\boldsymbol{V} \odot \boldsymbol{S})^+ * \boldsymbol{X}^+ + (\boldsymbol{V} \odot \boldsymbol{S})^- * \boldsymbol{X}^-\Big)_{i,j,k}.$$

Overview of the proof. $\boldsymbol{S} \in \{0,1\}^{\mathrm{shape}(\boldsymbol{V})}$ is such that $\mathcal{V} = \boldsymbol{V} \odot \boldsymbol{S}$ contains only non-negative edges going from each input channel t to the output channels $(t-1)n+1, \ldots, tn$, and only non-positive edges going from each input channel t to the output channels $tn+1, \ldots, 2tn$, while all remaining edges are set to zero. □

We approximate a single convolution $\boldsymbol{K} * \boldsymbol{X}$ by pruning a polynomially larger neural network of the form $\boldsymbol{U} * \mathrm{relu}(\boldsymbol{V} * \boldsymbol{X})$ exploiting only a channel blocked mask and filter removal: this is achieved using the MRSS result (Theorem 5.3.2).

Lemma 5.4.5 (Kernel pruning)**.** *Let $D, d, c_0, c_1, n \in \mathbb{N}$ be positive integers, $\varepsilon \in (0, \frac{1}{4})$, $M \in \mathbb{R}_{>0}$, and $C \in \mathbb{R}_{>0}$ be a universal constant with*

$$n \geq C d^{12} c_1^6 \log^3 \frac{d^2 c_1 c_0}{\varepsilon}.$$

Let $\mathbf{U} \sim \mathcal{N}^{d\times d\times 2nc_0\times c_1}$, $\mathbf{V} \sim \mathcal{N}^{1\times 1\times c_0\times 2nc_0}$ and $tS \in \{0,1\}^{\mathrm{shape}(\boldsymbol{V})}$, with $\boldsymbol{S}$ being a $2n$-channel-blocked mask. We define $N_0(\boldsymbol{X}) = \mathbf{U} * \mathrm{relu}(\mathbf{V} * \boldsymbol{X})$ where $\boldsymbol{X} \in \mathbb{R}^{D\times D\times c_0}$, and its pruned version $N_0^{(\boldsymbol{S})}(\boldsymbol{X}) = \mathbf{U} * \mathrm{relu}((\mathbf{V} \odot \boldsymbol{S}) * \boldsymbol{X})$. With probability $1-\varepsilon$, for all $\boldsymbol{K} \in \mathbb{R}^{d\times d\times c_0\times c_1}$ with $\|\boldsymbol{K}_{:,:,t_0,:}\|_1 \leq 1$ for each $t_0 \in [c_0]$, it is possible to remove filters from $N_0^{(\boldsymbol{S})}$ to obtain a CNN $\tilde{N}_0^{(\boldsymbol{S})}$ for which

$$\sup_{\boldsymbol{X}:\|\boldsymbol{X}\|_\infty \leq M} \|\boldsymbol{K} * \boldsymbol{X} - \tilde{N}_0^{(\boldsymbol{S})}(\boldsymbol{X})\|_\infty < \varepsilon M.$$

Overview of the proof. Exploiting Lemma 5.4.4, for each $(r, s, t_1) \in [d] \times [d] \times [c_1]$, one can show that

$$(\mathbf{U} * \mathrm{relu}((\mathbf{V} \odot \boldsymbol{S}) * \boldsymbol{X}))_{r,s,t_1} = \sum_{i,j \in [d], t_0 \in [c_0]} \left(\sum_{k \in [nc_0]} \mathsf{U}_{i,j,k,t_1} \cdot \tilde{\mathsf{V}}^{+}_{1,1,t_0,k} \right) \cdot \boldsymbol{X}^{+}_{r-i+1,s-j+1,t_0} + \sum_{i,j \in [d], t_0 \in [c_0]} \left(\sum_{k \in [nc_0]} \mathsf{U}_{i,j,k,t_1} \cdot \tilde{\mathsf{V}}^{-}_{1,1,t_0,k} \right) \cdot \boldsymbol{X}^{-}_{r-i+1,s-j+1,t_0}.$$

Through a Chernoff bound, we show that $\tilde{\mathbf{V}}^{+}_{1,1,t_0,:}$ has at least $n/3$ non-zero entries. Up to reshaping the tensor as a one-dimensional vector, we observe that $\mathbf{U}_{:,:,k,:} \tilde{\mathsf{V}}^{+}_{1,1,t_0,k}$ is an NSN vector (Lemma I.5 in appendix I). Hence, we can apply a boosted version of the MRSS result (Corollary I.6 in appendix I) and show we can prune all but roughly $\sqrt{n/(C_1 \log \frac{1}{\varepsilon})}$ positive entries of $\tilde{\mathsf{V}}^{+}_{1,1,t_0,k}$, with C_1 being a universal constant, such that $\sum_{k \in [nc_0]} \mathbf{U}_{:,:,k,:} \tilde{\mathbf{V}}^{+}_{1,1,t_0,:}$ approximates the channels $\boldsymbol{K}_{:,:,t_0,:}$ up to error $\varepsilon/(2d^2 c_0 c_1)$. The same holds for $\sum_{k \in [nc_0]} \mathbf{U}_{:,:,k,:} \tilde{\mathbf{V}}^{-}_{1,1,t_0,:}$. This pruning can be achieved by further zeroing the entries of the mask $\tilde{\boldsymbol{S}}$. Through some non-trivial calculations and by applying the Tensor Convolution Inequality (Lemma I.7 in appendix I), one can combine the above results to get the thesis. □

Remark 5.4.1 – From the proof of Lemma 5.4.5, we can see that the overall modification yields a pruned CNN $\hat{\mathbf{U}} * \mathrm{relu}(\hat{\mathbf{V}} * \boldsymbol{X})$ with $\hat{\mathbf{V}} \in \mathbb{R}^{1 \times 1 \times c_0 \times 2mc_0}$ and $\hat{\mathbf{U}} \in \mathbb{R}^{d \times d \times 2mc_0 \times c_1}$, where $m = \sqrt{n/(C_1 \log \frac{1}{\varepsilon})}$ for a universal constant C_1. Moreover, the kernel $\hat{\mathbf{V}}$ is structured as if pruned by a $2m$-channel-blocked mask.

Overview of the proof of Theorem 5.3.1

We iteratively apply Lemma 5.4.5 to each layer while carefully controlling the approximation error via tools such as the Lipschitz property of ReLU and the Tensor Convolution Inequality (Lemma I.7). More precisely, we show that (i) the approximation error does not increase too much at each layer; and (ii) all layer approximations can be combined to approximate the entire target network.

5.5 Limitations and future work

In previous works (da Cunha, Natale, & Viennot, 2022; Burkholz, 2022a) the assumption that the kernel of every second layer has shape $1 \times 1 \times \ldots$ is only an artifact of the proof since one can readily prune entries of an arbitrarily shaped tensor to enforce the desired shape. In our case, however, the concept of structured pruning can be quite broad, and such reshaping via pruning might not fit some sparsity patterns, depending on the context. The hypothesis on the shape can be a relevant limitation for such use cases. The constructions proposed by Burkholz (2022a, 2022b) appear as a promising direction to overcome this limitation, with the added benefit of reducing the depth overhead.

The convolution operation commonly employed in CNNs can be cumbersome at many points of our analysis. Exploring different concepts of convolution can be an interesting path for future

work as it could lead to tidier proofs and more general results. For instance, employing a 3D convolution would spare a factor c in Theorem 5.3.1.

Another limitation of our results is the restriction to ReLU as the activation function. Many previous works on the SLTH exploit the fact that ReLU satisfies the identity $x = \mathrm{relu}(x) - \mathrm{relu}(-x)$. Burkholz (2022a) leverages that to obtain an SLTH result for CNNs with activation functions f for which $f(x) - f(-x) \approx x$ around the origin. Our analysis, on the other hand, does not rely on such property, so adapting the approach of Burkholz (2022a) to our setting is not straightforward.

Finally, we remark that the assumption of normally distributed weights might be relaxed. Borst, Dadush, Huiberts, and Tiwari (2023) provided an MRSSP result for independent random variables whose distribution converges "fast enough" to a Gaussian one.[3] We believe our arguments can serve well as baselines to generalise our results to support random weights distributed as such.

[3]The required convergence rate is higher than that ensured by the Berry-Esseen theorem.

PART

Application

CHAPTER 6

Randomised Circuits

```
! pgfkeys Error: I do not know the key
'/tikz/align' and I am going to ignore it.
Perhaps you misspelled it.
! circuitikz Error: Giving up on this
path. Did you forget a semicolon?
! Missing number, treated as zero. <to
be read again> \let 1.10 \draw (1,1)
to[resistor] (2,
! Dimension too large. <recently read>
\pgf@ya 1.11 \draw (0,0) to[short,o-]
(0.2,0);
```

— CircuiTi*k*Z, Figure 6.2.

The high energy demands of modern Artificial Intelligence not only imply huge costs to run it at scale, but also constrains its deployment on edge devices. While analog computing offers a way to run those algorithms with orders of magnitude more efficiency, most proposals for its actual implementation require very high precision components and would depend on non-standard manufacturing. We propose a new method that uses a few inaccurate components to build accurate and programmable resistors, allowing analog neuromorphic devices to be manufactured with standard processes. It leverages the possibility of approximating any target value by summing a subset of given random values.

6.1 Problem Solved

Integrated resistors suffer from poor accuracy, with variations in resistance as large as 20% (Talebbeydokhti, Hanumolu, Kurahashi, & Moon, 2006). The need for better accuracy motivates the inclusion of additional trimming bits, post-fabrication testing, and circuitry resulting in higher cost, larger silicon area, and longer test times (Talebbeydokhti et al., 2006; McLaren & Martin, 2001). Moreover, resistors are a non-programmable component: their resistance is essentially fixed, having no controllable and precise way to change its value.

This chapter draws inspiration from theoretical results on the Random Subset-Sum Problem and Random Number Partitioning Problem. In analogy to those setups, we embrace the inaccuracy of components to obtain not only programmable properties but also higher degrees of accuracy in a stable manner.

6.2 Prior Solutions

The use of memristors as programmable resistive elements has features similar to those of the invention here disclosed. Nonetheless, our proposal distinguishes from memristors in many ways, as we leverage more classical components to achieve programmability and accuracy. Moreover, memristors suffer from innate instability in the sense that the current flow it influences also changes its resistivity. This is not the case for the circuits we present. We also remark that many memristor technologies suffer from low endurance (Merced-Grafals, Dávila, Ge, Williams, & Strachan, 2016), allowing for a limited amount of rewriting, whilst the basic blocks of the device proposed here have no such limitations. Further comparisons would be highly dependent on the physical implementation of the devices.

X. Zhang, Ni, Mukhopadhyay, and Apsel (2012) uses optimization methods to reduce inaccuracy in integrated resistors. It proposes combining different types of resistors to obtain an equivalent resistance with smaller expected variability. Among others, X. Zhang et al. (2012) differs from our approach in that we leverage component variability instead of avoiding it.

Forgoing the programmability of the circuit reduces our discussion to known techniques for programming (statically) accurate resistances. One way of realizing this approach would be to replace the transistors in the circuits by fuses/anti-fuses and selectively blowing them in a post-processing step (after solving MIP (6.1)). We stress that, whilst our proposal offer indefinite programmability, fuse/anti-fuse blowing can be done only once.

6.3 Description

In the following, we discuss the details of a possible implementation of our idea, as depicted in figure 6.1.

The device has an analog input V (as voltage relative to the output), n digital inputs $s_1, s_2, \ldots, s_n$ and an analog output y (as current). By sustaining a voltage V at the analog input, each transistor t_i determines whether current can flow through the correspondent vertical connection in the schematic. Thus, the i-th binary input s_i controls the contribution of the respective resistor to the final current. If s_i is set to 1, that is, if it provides sufficient current to saturate t_i,

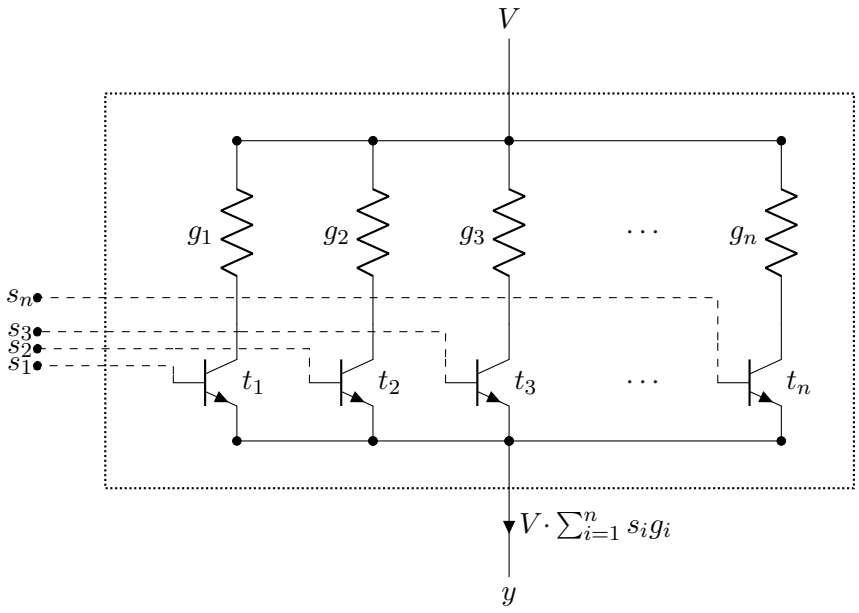


Figure 6.1: An implementation of the device proposed. The resistors shown have conductances $g_1, g_2, \ldots, g_n$ that can be inaccurate. Given a target conductance, one can set the transistors $t_1, t_2, \ldots, t_n$ through a binary signal $\{s_1, s_2, \ldots, s_n\}$ to get an equivalent conductance that approximates the target value.

by Ohm's law[1] the current flowing through the resistor with conductance g_i is $g_i V$. On the other hand, if s_i is set to 0, that is, the current is low enough to cutoff t_i, the current flow is blocked.

The binary nature of the variables $s_1, s_2, \ldots, s_n$ allows us to express the flow of current through the resistor with conductance g_i as

$$s_i g_i V.$$

Therefore, by Kirchhoff's current law, the total current in the analog output y is given by

$$\sum_{i=1}^{n} s_i g_i V = V \cdot \sum_{i=1}^{n} s_i g_i.$$

That is, the circuit has an equivalent conductance

$$\sum_{i=1}^{n} s_i g_i.$$

Hence, its value can be controlled by setting the binary inputs s_i.

This configuration is an instance of a Subset Sum Problem. Given a target conductance G, we want to solve the mixed-integer program (MIP)

$$\begin{aligned} \min_{s_1,\ldots,s_n} \quad & \Big| G - \sum_{i=1}^{n} s_i g_i \Big|, \\ \text{s.t.} \quad & s_i \in \{0,1\} \quad \forall i \in \{1,2,\ldots,n\}. \end{aligned} \tag{6.1}$$

[1] Here we ignore the transistor's influence on the resulting current as those can be set to be negligible in comparison to the contributions of the resistors.

This is a well known problem, sometimes discussed in the literature under the equivalent version of Number Partitioning Problem. We highlight the following result on the random version of the problem.

Theorem 6.3.1 (Lueker (1998)). *Given $n \in \mathbb{N}$, let $x_1, x_2, \ldots, x_n$ be independently and identically distributed random variables from a uniform distribution over $[-1, 1]$. Given $\varepsilon, \delta > 0$, there exists a constant C for which, if*

$$n \geq C \log\left(\frac{1}{\min\{\varepsilon, \delta\}}\right),$$

then, with probability at least $1 - \delta$,

$$\forall z \in [-1, 1],\ \exists S \subseteq \{1, 2, \ldots, n\} : \left| z - \sum_{i \in S} x_i \right| < \varepsilon.$$

While Theorem 6.3.1 can be enunciated in more general terms, we state it here without aiming to satisfy its hypothesis, but rather to illustrate the good behaviour of the problem. Given a target accuracy, one can expect that a small resistor count (relative to the accuracy) suffices to approximate any value within a range up to a tolerable error. Furthermore, this should be expected even if the resistors have inaccurate conductances.

Once the conductance values $g_1, \ldots, g_n$ are known, one can solve MIP (6.1) using generic mathematical optimization software like Gurobi (Gurobi Optimization, LLC, 2021). In our experiments, this software managed to do so tenths of thousands of times in just a few minutes.

6.4 Possible Applications

More accurate resistors can benefit many types of integrated circuits, such as bias networks (Wu & Chou, 2001; Talebbeydokhti et al., 2006), references (Sengupta, Carastro, & Allen, 2005; E. K. F. Lee, 2010), and filters (Vasilopoulos, Vitzilaios, Theodoratos, & Papananos, 2006). However, the combination of stability, accuracy, and programmability of the proposed solution makes it most promising when applied to "in-memory computing". In particular, as resistance units of resistive crossbars, such as the one depicted in figure 6.2.

While the diagram in figure 6.2 shows actual resistors as the sources of conductance for resistive crossbars, in practice other components or circuits are usually employed. Chakraborty et al. (2020) provides a deep review of the current status of this technology and discusses many of its limitations. Regarding the properties offered, the use of memristors as resistance units in crossbars resembles the most our proposal. Ankit et al. (2019) illustrates well such use.

6.5 Design Around

One can discuss the same ideas presented in this document in terms of resistances instead of conductances. The analogous circuit is a version of the one represented in figure 6.1 with connections in series rather than in parallel. Figure 6.3 represents an implementation of this analogy.

Including negative values in MIP (6.1) might be of interest. While resistances and conductances cannot be negative, one can achieve similar effect by feeding the opposite of the input voltage to a second instance of the proposed circuits and connecting the outputs of both instances.

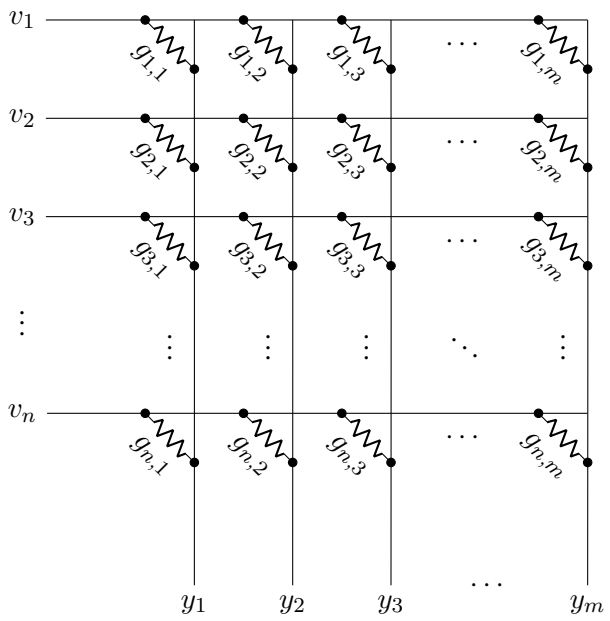


Figure 6.2: A resistive crossbar. Given voltage inputs $v_1, \ldots, v_n$, by Ohm's law, the resistor with conductance $g_{i,j}$ contributes with a current of $v_i g_{i,j}$ to the j-th vertical connection. By Kirchhoff's current law, those currents are added and amount to $\sum_{i=1}^{n} v_i g_{i,j}$ at y_j. In effect, this accomplishes an analog calculation of the matrix-vector product between a matrix with entries $g_{i,j}$ and a vector with entries v_i resulting in a vector with entries y_j.

Muralimanohar, Feinberg, and Shafiee-Ardestani (2018) discusses the same idea in terms of resistive crossbars.

If we keep restricted to positive conductance values (that is, not using negative voltage copies as we just mentioned), one can benefit from extra constraints on MIP (6.1). For example, if the variation in conductance values is small relative to the mean, one might benefit from solving the problem restricted to subsets of a fixed size.

We remark the ideas presented here in terms of resistors have immediate analogous based in other types of components, such as capacitors.

Finally, the circuit could integrate a map of conductance values and the respective binary inputs.

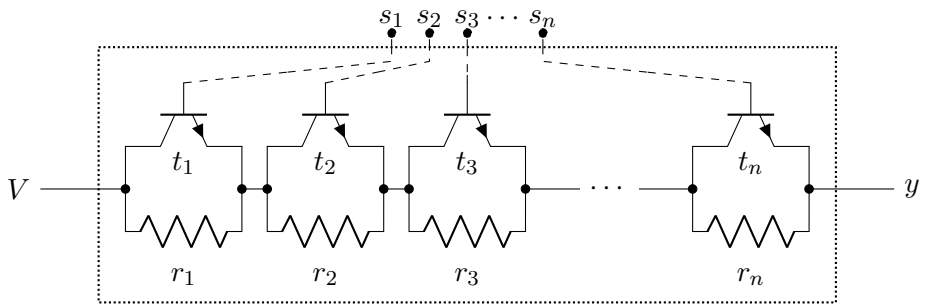


Figure 6.3: An analog of the circuit in figure 6.1 in terms of resistances $r_1, \ldots, r_n$. An input voltage V (relative to the output) results in a current $V \Big/ \sum_{i=1}^{n} s_i r_i$ at the output y.

References

Adams, G. S., Converse, B. A., Hales, A. H., & Klotz, L. E. (2021, Apr 01). People systematically overlook subtractive changes. *Nature*, *592*(7853), 258-261. Retrieved from https://doi.org/10.1038/s41586-021-03380-y doi: 10.1038/s41586-021-03380-y

Aladago, M. M., & Torresani, L. (2021). Slot machines: Discovering winning combinations of random weights in neural networks. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning, ICML 2021, 18-24 july 2021, virtual event* (Vol. 139, pp. 163–174). PMLR. Retrieved from http://proceedings.mlr.press/v139/aladago21a.html

Alvarez, J. M., & Salzmann, M. (2016). Learning the number of neurons in deep networks. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems 29: Annual conference on neural information processing systems 2016, december 5-10, 2016, barcelona, spain* (pp. 2262–2270). Retrieved from https://proceedings.neurips.cc/paper/2016/hash/6e7d2da6d3953058db75714ac400b584-Abstract.html

Amodei, D., Hernandez, D., Sastry, G., Clark, J., Brockman, G., & Sutskever, I. (2018, May). *Ai and compute*. OpenAI. Retrieved 2023/06/13, from https://openai.com/research/ai-and-compute

Ankit, A., Hajj, I. E., Chalamalasetti, S. R., Ndu, G., Foltin, M., Williams, R. S., ... Milojicic, D. S. (2019). PUMA: A programmable ultra-efficient memristor-based accelerator for machine learning inference. In I. Bahar, M. Herlihy, E. Witchel, & A. R. Lebeck (Eds.), *Proceedings of the twenty-fourth international conference on architectural support for programming languages and operating systems, ASPLOS 2019, providence, ri, usa, april 13-17, 2019* (pp. 715–731). ACM. Retrieved from https://doi.org/10.1145/3297858.3304049 doi: 10.1145/3297858.3304049

Anwar, S., Hwang, K., & Sung, W. (2017). Structured pruning of deep convolutional neural networks. *ACM J. Emerg. Technol. Comput. Syst.*, *13*(3), 32:1–32:18. Retrieved from https://doi.org/10.1145/3005348 doi: 10.1145/3005348

Azuma, K. (1967). Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, *19*(3), 357 – 367. Retrieved from https://doi.org/10.2748/tmj/1178243286 doi: 10.2748/tmj/1178243286

Balas, V. E., Roy, S. S., Sharma, D., & Samui, P. (Eds.). (2019). *Handbook of deep learning applications*. Springer. Retrieved from https://doi.org/10.1007/978-3-030-11479-4 doi: 10.1007/978-3-030-11479-4

Becchetti, L., da Cunha, A. C. W., Clementi, A., d' Amore, F., Lesfari, H., Natale, E., & Trevisan, L. (2022, 7). *On the Multidimensional Random Subset Sum Problem* (report). https://hal.science/hal-03738204: Inria & Université Cote d'Azur, CNRS, I3S, Sophia Antipolis,

France ; Sapienza Università di Roma, Rome, Italy ; Università Bocconi, Milan, Italy ; Università di Roma Tor Vergata, Rome, Italy.

Beier, R., & Vöcking, B. (2003). Random knapsack in expected polynomial time. In L. L. Larmore & M. X. Goemans (Eds.), *Proceedings of the 35th annual ACM symposium on theory of computing, june 9-11, 2003, san diego, ca, USA* (pp. 232–241). ACM. Retrieved from https://doi.org/10.1145/780542.780578 doi: 10.1145/780542.780578

Beier, R., & Vöcking, B. (2004). Probabilistic analysis of knapsack core algorithms. In J. I. Munro (Ed.), *Proceedings of the fifteenth annual ACM-SIAM symposium on discrete algorithms, SODA 2004, new orleans, louisiana, usa, january 11-14, 2004* (pp. 468–477). SIAM. Retrieved from http://dl.acm.org/citation.cfm?id=982792.982859

Bellman, R. (1966). Dynamic programming. *Science*, *153*(3731), 34–37.

Bengio, Y., Léonard, N., & Courville, A. C. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, *abs/1308.3432*. Retrieved from http://arxiv.org/abs/1308.3432

Blalock, D. W., Ortiz, J. J. G., Frankle, J., & Guttag, J. V. (2020). What is the state of neural network pruning? In I. S. Dhillon, D. S. Papailiopoulos, & V. Sze (Eds.), *Proceedings of machine learning and systems 2020, mlsys 2020, austin, tx, usa, march 2-4, 2020*. mlsys.org. Retrieved from https://proceedings.mlsys.org/book/296.pdf

Borgs, C., Chayes, J. T., Mertens, S., & Pittel, B. G. (2004). Phase diagram for the constrained integer partitioning problem. *Random Struct. Algorithms*, *24*(3), 315–380. Retrieved from https://doi.org/10.1002/rsa.20001 doi: 10.1002/rsa.20001

Borgs, C., Chayes, J. T., & Pittel, B. G. (2001). Phase transition and finite-size scaling for the integer partitioning problem. *Random Struct. Algorithms*, *19*(3-4), 247–288. Retrieved from https://doi.org/10.1002/rsa.10004 doi: 10.1002/rsa.10004

Borst, S., Dadush, D., Huiberts, S., & Kashaev, D. (2023). A nearly optimal randomized algorithm for explorable heap selection. In A. D. Pia & V. Kaibel (Eds.), *Integer programming and combinatorial optimization - 24th international conference, IPCO 2023, madison, wi, usa, june 21-23, 2023, proceedings* (Vol. 13904, pp. 29–43). Springer. Retrieved from https://doi.org/10.1007/978-3-031-32726-1_3 doi: 10.1007/978-3-031-32726-1_3

Borst, S., Dadush, D., Huiberts, S., & Tiwari, S. (2023). On the integrality gap of binary integer programs with gaussian data. *Mathematical Programming*, *197*(2), 1221–1263. Retrieved from https://doi.org/10.1007/s10107-022-01828-1 doi: 10.1007/s10107-022-01828-1

Borst, S., Dadush, D., & Mikulincer, D. (2023). Integrality gaps for random integer programs via discrepancy. In N. Bansal & V. Nagarajan (Eds.), *Proceedings of the 2023 ACM-SIAM symposium on discrete algorithms, SODA 2023, florence, italy, january 22-25, 2023* (pp. 1692–1733). SIAM. Retrieved from https://doi.org/10.1137/1.9781611977554.ch65 doi: 10.1137/1.9781611977554.ch65

Boyd, S. P., Ghosh, A., Prabhakar, B., & Shah, D. (2006). Randomized gossip algorithms. *IEEE Trans. Inf. Theory*, *52*(6), 2508–2530. Retrieved from https://doi.org/10.1109/TIT.2006.874516 doi: 10.1109/TIT.2006.874516

Bringmann, K., & Wellnitz, P. (2021). On near-linear-time algorithms for dense subset sum. In D. Marx (Ed.), *Proceedings of the 2021 ACM-SIAM symposium on discrete algorithms, SODA 2021, virtual conference, january 10 - 13, 2021* (pp. 1777–1796). SIAM. Retrieved from https://doi.org/10.1137/1.9781611976465.107 doi: 10.1137/1.9781611976465.107

Bürger, R. (2000). *The mathematical theory of selection, recombination, and mutation*. John Wiley & Sons.

Burkholz, R. (2022a). Convolutional and residual networks provably contain lottery tickets. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, & S. Sabato (Eds.), *International conference on machine learning, ICML 2022, 17-23 july 2022, baltimore, maryland, USA* (Vol. 162, pp. 2414–2433). PMLR. Retrieved from https://proceedings.mlr.press/v162/burkholz22a.html

Burkholz, R. (2022b). Most activation functions can win the lottery without excessive depth. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (Vol. 35, pp. 18707–18720). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2022/file/76bf7786d311217077bc8bb021946cd9-Paper-Conference.pdf

Burkholz, R., Laha, N., Mukherjee, R., & Gotovos, A. (2022). On the existence of universal lottery tickets. In *International conference on learning representations*. Retrieved from https://openreview.net/forum?id=SYB4WrJql1n

Chakraborty, I., Ali, M. F., Ankit, A., Jain, S., Roy, S., Sridharan, S., ... Roy, K. (2020). Resistive crossbars as approximate hardware building blocks for machine learning: Opportunities and challenges. *Proc. IEEE*, *108*(12), 2276–2310. Retrieved from https://doi.org/10.1109/JPROC.2020.3003007 doi: 10.1109/JPROC.2020.3003007

Chastain, E., Livnat, A., Papadimitriou, C., & Vazirani, U. (2014). Algorithms, games, and evolution. *Proceedings of the National Academy of Sciences*, *111*(29), 10620–10623.

Chellapilla, K., Puri, S., & Simard, P. (2006). High performance convolutional neural networks for document processing. In *Tenth international workshop on frontiers in handwriting recognition*.

Chen, X., Jin, Y., Randolph, T., & Servedio, R. A. (2022). Average-case subset balancing problems. In J. S. Naor & N. Buchbinder (Eds.), *Proceedings of the 2022 ACM-SIAM symposium on discrete algorithms, SODA 2022, virtual conference / alexandria, va, usa, january 9 - 12, 2022* (pp. 743–778). SIAM. Retrieved from https://doi.org/10.1137/1.9781611977073.33 doi: 10.1137/1.9781611977073.33

Chen, X., Zhang, J., & Wang, Z. (2022). Peek-a-boo: What (more) is disguised in a randomly weighted neural network, and how to find it efficiently. In *The tenth international conference*

on learning representations, ICLR 2022, virtual event, april 25-29, 2022. OpenReview.net. Retrieved from https://openreview.net/forum?id=moHCzz6D5H3

Cheng, H., Zhao, P., Li, Y., Lin, X., Diffenderfer, J., Goldhahn, R. A., & Kailkhura, B. (2022). Efficient multi-prize lottery tickets: Enhanced accuracy, training, and inference speed. *CoRR*, *abs/2209.12839*. Retrieved from https://doi.org/10.48550/arXiv.2209.12839 doi: 10.48550/arXiv.2209.12839

Chijiwa, D., Yamaguchi, S., Ida, Y., Umakoshi, K., & Inoue, T. (2021). Pruning randomly initialized neural networks with iterative randomization. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in neural information processing systems 34: Annual conference on neural information processing systems 2021, neurips 2021, december 6-14, 2021, virtual* (pp. 4503–4513). Retrieved from https://proceedings.neurips.cc/paper/2021/hash/23e582ad8087f2c03a5a31c125123f9a-Abstract.html

Cosentino, J., Zaiter, F., Pei, D., & Zhu, J. (2019). The search for sparse, robust neural networks. *CoRR*, *abs/1912.02386*. Retrieved from http://arxiv.org/abs/1912.02386

da Cunha, A., d'Amore, F., Giroire, F., Lesfari, H., Natale, E., & Viennot, L. (2022, April). *Revisiting the Random Subset Sum problem* (Research Report). Inria Sophia Antipolis - Méditerranée, Université Côte d'Azur ; Inria Paris. Retrieved 2022-10-11, from https://hal.archives-ouvertes.fr/hal-03654720

da Cunha, A., d'Amore, F., & Natale, E. (2023, June). *Convolutional neural networks contain structured strong lottery tickets.* Retrieved from https://hal.science/hal-04143024 (Preprint)

da Cunha, A., Natale, E., & Viennot, L. (2022). Proving the strong lottery ticket hypothesis for convolutional neural networks. In *The tenth international conference on learning representations, ICLR 2022, virtual event, april 25-29, 2022*. OpenReview.net. Retrieved from https://openreview.net/forum?id=Vjki79-619-

Da Cunha, A. C. W., Natale, E., & Viennot, L. (2022, October 5). *Résistance équivalente modulable à partir de résistances imprécises.* France Patent FR2210217. (Institut national de recherche en informatique et en automatique)

da Cunha, A. C. W., Natale, E., & Viennot, L. (2023). Neural network information leakage through hidden learning. In B. Dorronsoro, F. Chicano, G. Danoy, & E. Talbi (Eds.), *Optimization and learning - 6th international conference, OLA 2023, malaga, spain, may 3-5, 2023, proceedings* (Vol. 1824, pp. 117–128). Springer. Retrieved from https://doi.org/10.1007/978-3-031-34020-8_8 doi: 10.1007/978-3-031-34020-8_8

Davis, T. A., & Hu, Y. (2011). The university of florida sparse matrix collection. *ACM Trans. Math. Softw.*, *38*(1), 1:1–1:25. Retrieved from https://doi.org/10.1145/2049662.2049663 doi: 10.1145/2049662.2049663

Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE computer society conference on computer vision and*

pattern recognition (CVPR 2009), 20-25 june 2009, miami, florida, USA (pp. 248–255). IEEE Computer Society. Retrieved from https://doi.org/10.1109/CVPR.2009.5206848 doi: 10.1109/CVPR.2009.5206848

Denil, M., Shakibi, B., Dinh, L., Ranzato, M., & de Freitas, N. (2013). Predicting parameters in deep learning. In C. J. C. Burges, L. Bottou, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 26: 27th annual conference on neural information processing systems 2013. proceedings of a meeting held december 5-8, 2013, lake tahoe, nevada, united states* (pp. 2148–2156). Retrieved from https://proceedings.neurips.cc/paper/2013/hash/7fec306d1e665bc9c748b5d2b99a6e97-Abstract.html

Diffenderfer, J., & Kailkhura, B. (2021). Multi-prize lottery ticket hypothesis: Finding accurate binary neural networks by pruning A randomly weighted network. In *9th international conference on learning representations, ICLR 2021, virtual event, austria, may 3-7, 2021*. OpenReview.net. Retrieved from https://openreview.net/forum?id=U_mat0b9iv

Doerr, B. (2011). Analyzing randomized search heuristics: Tools from probability theory. In A. Auger & B. Doerr (Eds.), *Theory of randomized search heuristics: Foundations and recent developments* (Vol. 1, pp. 1–20). World Scientific. Retrieved from https://doi.org/10.1142/9789814282673_0001 doi: 10.1142/9789814282673_0001

Doerr, B. (2020). Probabilistic tools for the analysis of randomized optimization heuristics. In B. Doerr & F. Neumann (Eds.), *Theory of evolutionary computation - recent developments in discrete optimization* (pp. 1–87). Springer. Retrieved from https://doi.org/10.1007/978-3-030-29414-4_1 doi: 10.1007/978-3-030-29414-4_1

Doerr, B., & Kostrygin, A. (2017). Randomized rumor spreading revisited. In I. Chatzigiannakis, P. Indyk, F. Kuhn, & A. Muscholl (Eds.), *44th international colloquium on automata, languages, and programming, ICALP 2017, july 10-14, 2017, warsaw, poland* (Vol. 80, pp. 138:1–138:14). Schloss Dagstuhl - Leibniz-Zentrum für Informatik. Retrieved from https://doi.org/10.4230/LIPIcs.ICALP.2017.138 doi: 10.4230/LIPIcs.ICALP.2017.138

Dubhashi, D. P., & Panconesi, A. (2009). *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press. Retrieved from http://www.cambridge.org/gb/knowledge/isbn/item2327542/

Duff, I. S., Heroux, M. A., & Pozo, R. (2002). An overview of the sparse basic linear algebra subprograms: The new standard from the BLAS technical forum. *ACM Trans. Math. Softw.*, *28*(2), 239–267. Retrieved from https://doi.org/10.1145/567806.567810 doi: 10.1145/567806.567810

Dyer, M. E., & Frieze, A. M. (1989). Probabilistic analysis of the multidimensional knapsack problem. *Math. Oper. Res.*, *14*(1), 162–176. Retrieved from https://doi.org/10.1287/moor.14.1.162 doi: 10.1287/moor.14.1.162

Elsen, E., Dukhan, M., Gale, T., & Simonyan, K. (2020). Fast sparse convnets. In *2020 IEEE/CVF conference on computer vision and pattern recognition, CVPR 2020, seattle, wa, usa, june 13-19, 2020* (pp. 14617–14626). Computer Vision Foundation /

IEEE. Retrieved from https://openaccess.thecvf.com/content_CVPR_2020/html/Elsen_Fast_Sparse_ConvNets_CVPR_2020_paper.html doi: 10.1109/CVPR42600.2020.01464

Ferbach, D., Tsirigotis, C., Gidel, G., & Bose, A. J. (2022). A general framework for proving the equivariant strong lottery ticket hypothesis. *CoRR*, *abs/2206.04270*. Retrieved from https://doi.org/10.48550/arXiv.2206.04270 doi: 10.48550/arXiv.2206.04270

Fischer, J., & Burkholz, R. (2021). Towards strong pruning for lottery tickets with non-zero biases. *CoRR*, *abs/2110.11150*. Retrieved from https://arxiv.org/abs/2110.11150

Frankle, J. (2023). *The lottery ticket hypothesis: On sparse, trainable neural networks* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Frankle, J., & Carbin, M. (2019). The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th international conference on learning representations, ICLR 2019, new orleans, la, usa, may 6-9, 2019*. OpenReview.net. Retrieved from https://openreview.net/forum?id=rJl-b3RcF7

Frankle, J., Dziugaite, G. K., Roy, D. M., & Carbin, M. (2020). Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th international conference on machine learning, ICML 2020, 13-18 july 2020, virtual event* (Vol. 119, pp. 3259–3269). PMLR. Retrieved from http://proceedings.mlr.press/v119/frankle20a.html

Gaier, A., & Ha, D. (2019). Weight agnostic neural networks. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019, neurips 2019, december 8-14, 2019, vancouver, bc, canada* (pp. 5365–5379). Retrieved from https://proceedings.neurips.cc/paper/2019/hash/e98741479a7b998f88b8f8c9f0b6b6f1-Abstract.html

Gale, T., Zaharia, M., Young, C., & Elsen, E. (2020). Sparse GPU kernels for deep learning. In C. Cuicchi, I. Qualters, & W. T. Kramer (Eds.), *Proceedings of the international conference for high performance computing, networking, storage and analysis, SC 2020, virtual event / atlanta, georgia, usa, november 9-19, 2020* (p. 17). IEEE/ACM. Retrieved from https://doi.org/10.1109/SC41405.2020.00021 doi: 10.1109/SC41405.2020.00021

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of np-completeness*. W. H. Freeman.

Gemmell, P., & Johnston, A. M. (2001). Analysis of a subset sum randomizer. *IACR Cryptol. ePrint Arch.*, 18. Retrieved from http://eprint.iacr.org/2001/018

Goodfellow, I. J., Bengio, Y., & Courville, A. C. (2016). *Deep learning*. MIT Press. Retrieved from http://www.deeplearningbook.org/

Gorantla, S., Louis, A., Papadimitriou, C. H., Vempala, S., & Yadati, N. (2019). Biologically Plausible Neural Networks via Evolutionary Dynamics and Dopaminergic Plasticity. In *International conference on learning representations (iclr)*. Retrieved 2022-06-03, from https://openreview.net/forum?id=B1eh4mYIUB&referrer=[the%20profile%20of%20Santosh%20Vempala](/profile?id=~Santosh_Vempala1)

Gorbett, M., & Whitley, D. (2023). Randomly initialized subnetworks with iterative weight recycling. *CoRR*, *abs/2303.15953*. Retrieved from https://doi.org/10.48550/arXiv.2303.15953 doi: 10.48550/arXiv.2303.15953

Gregor, K., & LeCun, Y. (2010). Emergence of complex-like cells in a temporal product network with local receptive fields. *CoRR*, *abs/1006.0448*. Retrieved from http://arxiv.org/abs/1006.0448

Grönquist, P., Yao, C., Ben-Nun, T., Dryden, N., Dueben, P., Li, S., & Hoefler, T. (2021). Deep learning for post-processing ensemble weather forecasts. *Philosophical Transactions of the Royal Society A*, *379*(2194), 20200092.

Gurobi Optimization, LLC. (2021). *Gurobi Optimizer Reference Manual*. Retrieved from https://www.gurobi.com

Han, S., Kang, J., Mao, H., Hu, Y., Li, X., Li, Y., ... Dally, W. B. J. (2017). ESE: efficient speech recognition engine with sparse LSTM on FPGA. In J. W. Greene & J. H. Anderson (Eds.), *Proceedings of the 2017 ACM/SIGDA international symposium on field-programmable gate arrays, FPGA 2017, monterey, ca, usa, february 22-24, 2017* (pp. 75–84). ACM. Retrieved from http://dl.acm.org/citation.cfm?id=3021745

Han, S., Pool, J., Tran, J., & Dally, W. J. (2015). Learning both weights and connections for efficient neural network. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems 28: Annual conference on neural information processing systems 2015, december 7-12, 2015, montreal, quebec, canada* (pp. 1135–1143). Retrieved from https://proceedings.neurips.cc/paper/2015/hash/ae0eb3eed39d2bcef4622b2499a05fe6-Abstract.html

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction, 2nd edition*. Springer. Retrieved from https://doi.org/10.1007/978-0-387-84858-7 doi: 10.1007/978-0-387-84858-7

Hayes, B. (2002). The easiest hard problem. *American Scientist*, *90*, 113-117.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE international conference on computer vision, ICCV 2015, santiago, chile, december 7-13, 2015* (pp. 1026–1034). IEEE Computer Society. Retrieved from https://doi.org/10.1109/ICCV.2015.123 doi: 10.1109/ICCV.2015.123

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE conference on computer vision and pattern recognition, CVPR 2016, las vegas, nv, usa, june 27-30, 2016* (pp. 770–778). IEEE Computer Society. Retrieved from https://doi.org/10.1109/CVPR.2016.90 doi: 10.1109/CVPR.2016.90

He, Y., & Xiao, L. (2023). Structured pruning for deep convolutional neural networks: A survey. *CoRR*, *abs/2303.00566*. Retrieved from https://doi.org/10.48550/arXiv.2303.00566 doi: 10.48550/arXiv.2303.00566

Helm, A., & May, A. (2018). Subset sum quantumly in 1.17^n. In S. Jeffery (Ed.), *13th conference on the theory of quantum computation, communication and cryptography, TQC 2018, july 16-18, 2018, sydney, australia* (Vol. 111, pp. 5:1–5:15). Schloss Dagstuhl - Leibniz-Zentrum für Informatik. Retrieved from https://doi.org/10.4230/LIPIcs.TQC.2018.5 doi: 10.4230/LIPIcs.TQC.2018.5

Hoefler, T., Alistarh, D., Ben-Nun, T., Dryden, N., & Peste, A. (2021). Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *J. Mach. Learn. Res.*, *22*, 241:1–241:124. Retrieved from http://jmlr.org/papers/v22/21-0366.html

Hooker, S. (2021). The hardware lottery. *Commun. ACM*, *64*(12), 58–65. Retrieved from https://doi.org/10.1145/3467017 doi: 10.1145/3467017

Huang, G. B., Lee, H., & Learned-Miller, E. G. (2012). Learning hierarchical representations for face verification with convolutional deep belief networks. In *2012 IEEE conference on computer vision and pattern recognition, providence, ri, usa, june 16-21, 2012* (pp. 2518–2525). IEEE Computer Society. Retrieved from https://doi.org/10.1109/CVPR.2012.6247968 doi: 10.1109/CVPR.2012.6247968

Innes, M., Saba, E., Fischer, K., Gandhi, D., Rudilosso, M. C., Joy, N. M., ... Shah, V. (2018). Fashionable modelling with flux. *CoRR*, *abs/1811.01457*. Retrieved from https://arxiv.org/abs/1811.01457

Janowsky, S. A. (1989, Jun). Pruning versus clipping in neural networks. *Phys. Rev. A*, *39*, 6600–6603. Retrieved from https://link.aps.org/doi/10.1103/PhysRevA.39.6600 doi: 10.1103/PhysRevA.39.6600

Janson, S. (2004). Large deviations for sums of partly dependent random variables. *Random Struct. Algorithms*, *24*(3), 234–248. Retrieved from https://doi.org/10.1002/rsa.20008 doi: 10.1002/rsa.20008

Jin, C., Vyas, N., & Williams, R. (2021). Fast low-space algorithms for subset sum. In D. Marx (Ed.), *Proceedings of the 2021 ACM-SIAM symposium on discrete algorithms, SODA 2021, virtual conference, january 10 - 13, 2021* (pp. 1757–1776). SIAM. Retrieved from https://doi.org/10.1137/1.9781611976465.106 doi: 10.1137/1.9781611976465.106

Jin, C., & Wu, H. (2019). A simple near-linear pseudopolynomial time randomized algorithm for subset sum. In J. T. Fineman & M. Mitzenmacher (Eds.), *2nd symposium on simplicity in algorithms, SOSA 2019, january 8-9, 2019, san diego, ca, USA* (Vol. 69, pp. 17:1–17:6). Schloss Dagstuhl - Leibniz-Zentrum für Informatik. Retrieved from https://doi.org/10.4230/OASIcs.SOSA.2019.17 doi: 10.4230/OASIcs.SOSA.2019.17

Johnson, D. S., Aragon, C. R., McGeoch, L. A., & Schevon, C. (1991). Optimization by simulated annealing: An experimental evaluation; part ii, graph coloring and number partitioning. *Oper. Res.*, *39*(3), 378–406. Retrieved from https://doi.org/10.1287/opre.39.3.378 doi: 10.1287/opre.39.3.378

Kang, H., Mina, R. J. L., Madjid, S. R. H., Yoon, J., Hasegawa-Johnson, M., Hwang, S. J., & Yoo, C. D. (2022). Forget-free continual learning with winning subnetworks. In K. Chaudhuri,

S. Jegelka, L. Song, C. Szepesvári, G. Niu, & S. Sabato (Eds.), *International conference on machine learning, ICML 2022, 17-23 july 2022, baltimore, maryland, USA* (Vol. 162, pp. 10734–10750). PMLR. Retrieved from https://proceedings.mlr.press/v162/kang22b.html

Karmarkar, N., Karp, R. M., Lueker, G. S., & Odlyzko, A. M. (1986). Probabilistic analysis of optimum partitioning. *Journal of Applied probability*, *23*(3), 626–645.

Kate, A., & Goldberg, I. (2011). Generalizing cryptosystems based on the subset sum problem. *Int. J. Inf. Sec.*, *10*(3), 189–199. Retrieved from https://doi.org/10.1007/s10207-011-0129-2 doi: 10.1007/s10207-011-0129-2

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Y. Bengio & Y. LeCun (Eds.), *3rd international conference on learning representations, ICLR 2015, san diego, ca, usa, may 7-9, 2015, conference track proceedings*. Retrieved from http://arxiv.org/abs/1412.6980

Koster, N., Grothe, O., & Rettinger, A. (2022). Signing the supermask: Keep, hide, invert. In *The tenth international conference on learning representations, ICLR 2022, virtual event, april 25-29, 2022*. OpenReview.net. Retrieved from https://openreview.net/forum?id=e0jtGTfPihs

Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images. *Master's thesis, University of Tronto*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 25: 26th annual conference on neural information processing systems 2012. proceedings of a meeting held december 3-6, 2012, lake tahoe, nevada, united states* (pp. 1106–1114). Retrieved from https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

Kuzmin, A., Nagel, M., Pitre, S., Pendyam, S., Blankevoort, T., & Welling, M. (2019). Taxonomy and evaluation of structured compression of convolutional neural networks. *CoRR*, *abs/1912.09802*. Retrieved from http://arxiv.org/abs/1912.09802

Lange, R. T. (2020). The lottery ticket hypothesis: A survey. *https://roberttlange.github.io/year-archive/posts/2020/06/lottery-ticket-hypothesis/*. Retrieved from https://roberttlange.github.io/posts/2020/06/lottery-ticket-hypothesis/

Laurent, B., & Massart, P. (2000). Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, *28*(5), 1302 – 1338. Retrieved from https://doi.org/10.1214/aos/1015957395 doi: 10.1214/aos/1015957395

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324. Retrieved from https://ieeexplore.ieee.org/document/726791/ doi: 10.1109/5.726791

LeCun, Y., Cortes, C., & Burges, C. (2010). Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, *2*.

LeCun, Y., Denker, J. S., & Solla, S. A. (1989). Optimal brain damage. In D. S. Touretzky (Ed.), *Advances in neural information processing systems 2, [NIPS conference, denver, colorado, usa, november 27-30, 1989]* (pp. 598–605). Morgan Kaufmann. Retrieved from http://papers.nips.cc/paper/250-optimal-brain-damage

Lee, E. K. F. (2010). Low voltage CMOS bandgap references with temperature compensated reference current output. In *International symposium on circuits and systems (ISCAS 2010), may 30 - june 2, 2010, paris, france* (pp. 1643–1646). IEEE. Retrieved from https://doi.org/10.1109/ISCAS.2010.5537472 doi: 10.1109/ISCAS.2010.5537472

Lee, N., Ajanthan, T., & Torr, P. H. S. (2019). Snip: single-shot network pruning based on connection sensitivity. In *7th international conference on learning representations, ICLR 2019, new orleans, la, usa, may 6-9, 2019*. OpenReview.net. Retrieved from https://openreview.net/forum?id=B1VZqjAcYX

Li, A., Sun, J., Zeng, X., Zhang, M., Li, H., & Chen, Y. (2021). Fedmask: Joint computation and communication-efficient personalized federated learning via heterogeneous masking. In J. S. Silva, F. Boavida, A. Rodrigues, A. Markham, & R. Zheng (Eds.), *Sensys '21: The 19th ACM conference on embedded networked sensor systems, coimbra, portugal, november 15 - 17, 2021* (pp. 42–55). ACM. Retrieved from https://doi.org/10.1145/3485730.3485929 doi: 10.1145/3485730.3485929

Li, H., Kadav, A., Durdanovic, I., Samet, H., & Graf, H. P. (2017). Pruning filters for efficient convnets. In *5th international conference on learning representations, ICLR 2017, toulon, france, april 24-26, 2017, conference track proceedings*. OpenReview.net. Retrieved from https://openreview.net/forum?id=rJqFGTslg

Lin, T., Stich, S. U., Barba, L., Dmitriev, D., & Jaggi, M. (2020). Dynamic model pruning with feedback. In *8th international conference on learning representations, ICLR 2020, addis ababa, ethiopia, april 26-30, 2020*. OpenReview.net. Retrieved from https://openreview.net/forum?id=SJem8lSFwB

Lueker, G. S. (1982). On the average difference between the solutions to linear and integer knapsack problems. In R. L. Disney & T. J. Ott (Eds.), *Applied probability-computer science: The interface volume 1* (pp. 489–504). Boston, MA: Birkhäuser Boston. Retrieved from https://doi.org/10.1007/978-1-4612-5791-2_22 doi: 10.1007/978-1-4612-5791-2_22

Lueker, G. S. (1998). Exponentially small bounds on the expected optimum of the partition and subset sum problems. *Random Structures & Algorithms*, *12*(1), 51–62. Retrieved 2021-06-30, from https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291098-2418%28199801%2912%3A1%3C51%3A%3AAID-RSA3%3E3.0.CO%3B2-S (_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/%28SICI%291098-2418%28199801%2912%3A1%3C51%3A%3AAID-RSA3%3E3.0.CO%3B2-S) doi: 10.1002/(SICI)1098-2418(199801)12:1<51::AID-RSA3>3.0.CO;2-S

Luo, Q., Xu, X., Liu, H., Lv, H., Gong, T., Long, S., ... others (2016). Super non-linear rram with ultra-low power for 3d vertical nano-crossbar arrays. *Nanoscale*, *8*(34), 15629–15636.

Malach, E., Yehudai, G., Shalev-Shwartz, S., & Shamir, O. (2020). Proving the lottery ticket hypothesis: Pruning is all you need. In *Proceedings of the 37th international conference on machine learning, ICML 2020, 13-18 july 2020, virtual event* (Vol. 119, pp. 6682–6691). PMLR. Retrieved from http://proceedings.mlr.press/v119/malach20a.html

McLaren, A., & Martin, K. (2001). Generation of accurate on-chip time constants and stable transconductances. *IEEE Journal of Solid-State Circuits*, *36*(4), 691–695.

Merced-Grafals, E. J., Dávila, N., Ge, N., Williams, R. S., & Strachan, J. P. (2016). Repeatable, accurate, and high speed multi-level programming of memristor 1t1r arrays for power efficient analog computing applications. *Nanotechnology*, *27 36*, 365202.

Mertens, S. (1998). Phase transition in the number partitioning problem. *Physical Review Letters*, *81*, 4281-4284.

Mertens, S. (2001). A physicist's approach to number partitioning. *Theor. Comput. Sci.*, *265*(1-2), 79–108. Retrieved from https://doi.org/10.1016/S0304-3975(01)00153-0 doi: 10.1016/S0304-3975(01)00153-0

Mezard, M., & Montanari, A. (2009). *Information, physics, and computation*. Oxford ; New York: Oxford University Press. (OCLC: ocn234430714)

Michel, P., Levy, O., & Neubig, G. (2019). Are sixteen heads really better than one? In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019, neurips 2019, december 8-14, 2019, vancouver, bc, canada* (pp. 14014–14024). Retrieved from https://proceedings.neurips.cc/paper/2019/hash/2c601ad9d2ff9bc8b282670cdd54f69f-Abstract.html

Midya, R., Wang, Z., Zhang, J., Savel'ev, S. E., Li, C., Rao, M., ... others (2017). Anatomy of ag/hafnia-based selectors with 1010 nonlinearity. *Advanced Materials*, *29*(12), 1604457.

Mörters, P., & Peres, Y. (2010). *Brownian motion* (Vol. 30). Cambridge University Press.

Mozaffari, H., Shejwalkar, V., & Houmansadr, A. (2023). Every vote counts: Ranking-based training of federated learning to resist poisoning attacks. In *32nd usenix security symposium (usenix security 23)*.

Mozer, M., & Smolensky, P. (1988). Skeletonization: A technique for trimming the fat from a network via relevance assessment. In D. S. Touretzky (Ed.), *Advances in neural information processing systems 1, [NIPS conference, denver, colorado, usa, 1988]* (pp. 107–115). Morgan Kaufmann. Retrieved from http://papers.nips.cc/paper/119-skeletonization-a-technique-for-trimming-the-fat-from-a-network-via-relevance-assessment

Mozer, M. C., & Smolensky, P. (1989). Using relevance to reduce network size automatically. *Connection Science*, *1*(1), 3–16.

Muralimanohar, N., Feinberg, B., & Shafiee-Ardestani, A. (2018, March). *Vector-matrix multiplications involving negative values.* Google Patents. (US Patent 9,910,827)

Naumov, M., Chien, L., Vandermersch, P., & Kapasi, U. (2010). Cusparse library. In *Gpu technology conference.*

Orseau, L., Hutter, M., & Rivasplata, O. (2020). Logarithmic pruning is all you need. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems 33: Annual conference on neural information processing systems 2020, neurips 2020, december 6-12, 2020, virtual.* Retrieved from https://proceedings.neurips.cc/paper/2020/hash/1e9491470749d5b0e361ce4f0b24d037-Abstract.html

Panafieu, É. d., Lamali, M. L., & Wallner, M. (2019). Combinatorics of nondeterministic walks of the dyck and motzkin type. In *2019 proceedings of the sixteenth workshop on analytic algorithmics and combinatorics (analco)* (pp. 1–12).

Pase, F., Isik, B., Gunduz, D., Weissman, T., & Zorzi, M. (2022). Efficient federated random subnetwork training. In *Workshop on federated learning: Recent advances and new challenges (in conjunction with neurips 2022).* Retrieved from https://openreview.net/forum?id=YZIVv_37y2z

Patterson, D. A., Gonzalez, J., Hölzle, U., Le, Q. V., Liang, C., Munguia, L., ... Dean, J. (2022). The carbon footprint of machine learning training will plateau, then shrink. *Computer*, *55*(7), 18–28. Retrieved from https://doi.org/10.1109/MC.2022.3148714 doi: 10.1109/MC.2022.3148714

Pensia, A., Rajput, S., Nagle, A., Vishwakarma, H., & Papailiopoulos, D. S. (2020). Optimal lottery tickets via subset sum: Logarithmic over-parameterization is sufficient. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems 33: Annual conference on neural information processing systems 2020, neurips 2020, december 6-12, 2020, virtual.* Retrieved from https://proceedings.neurips.cc/paper/2020/hash/1b742ae215adf18b75449c6e272fd92d-Abstract.html

Polyak, A., & Wolf, L. (2015). Channel-level acceleration of deep face representations. *IEEE Access*, *3*, 2163–2175. Retrieved from https://doi.org/10.1109/ACCESS.2015.2494536 doi: 10.1109/ACCESS.2015.2494536

Pooch, U. W., & Nieder, A. (1973). A survey of indexing techniques for sparse matrices. *ACM Comput. Surv.*, *5*(2), 109–133. Retrieved from https://doi.org/10.1145/356616.356618 doi: 10.1145/356616.356618

Qin, H., Gong, R., Liu, X., Bai, X., Song, J., & Sebe, N. (2020). Binary neural networks: A survey. *Pattern Recognit.*, *105*, 107281. Retrieved from https://doi.org/10.1016/j.patcog.2020.107281 doi: 10.1016/j.patcog.2020.107281

Ramanujan, V., Wortsman, M., Kembhavi, A., Farhadi, A., & Rastegari, M. (2020). What's hidden in a randomly weighted neural network? In *2020 IEEE/CVF conference on computer vision and pattern recognition, CVPR 2020, seattle, wa, usa, june 13-19, 2020* (pp. 11890–11899). Computer Vision Foundation / IEEE. Retrieved from https://openaccess.thecvf.com/content_CVPR_2020/html/Ramanujan_Whats_Hidden_in_a_Randomly_Weighted_Neural_Network_CVPR_2020_paper.html doi: 10.1109/CVPR42600.2020.01191

Reed, R. (1993). Pruning algorithms-a survey. *IEEE Trans. Neural Networks*, *4*(5), 740–747. Retrieved from https://doi.org/10.1109/72.248452 doi: 10.1109/72.248452

Renda, A., Frankle, J., & Carbin, M. (2020). Comparing rewinding and fine-tuning in neural network pruning. In *8th international conference on learning representations, ICLR 2020, addis ababa, ethiopia, april 26-30, 2020*. OpenReview.net. Retrieved from https://openreview.net/forum?id=S1gSj0NKvB

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986a). Learning representations by back-propagating errors. *nature*, *323*(6088), 533–536.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986b). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Ruml, W., Ngo, J. T., Marks, J., & Shieber, S. M. (1996). Easily searched encodings for number partitioning. *Journal of Optimization Theory and Applications*, *89*(2), 251–291.

Sabatelli, M., Kestemont, M., & Geurts, P. (2021). On the transferability of winning tickets in non-natural image datasets. In G. M. Farinella, P. Radeva, J. Braz, & K. Bouatouch (Eds.), *Proceedings of the 16th international joint conference on computer vision, imaging and computer graphics theory and applications, VISIGRAPP 2021, volume 5: Visapp, online streaming, february 8-10, 2021* (pp. 59–69). SCITEPRESS. Retrieved from https://doi.org/10.5220/0010196300590069 doi: 10.5220/0010196300590069

Saxe, A. M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B. D., & Cox, D. D. (2018). On the information bottleneck theory of deep learning. In *6th international conference on learning representations, ICLR 2018, vancouver, bc, canada, april 30 - may 3, 2018, conference track proceedings*. OpenReview.net. Retrieved from https://openreview.net/forum?id=ry_WPG-A-

See, A., Luong, M., & Manning, C. D. (2016). Compression of neural machine translation models via pruning. In Y. Goldberg & S. Riezler (Eds.), *Proceedings of the 20th SIGNLL conference on computational natural language learning, conll 2016, berlin, germany, august 11-12, 2016* (pp. 291–301). ACL. Retrieved from https://doi.org/10.18653/v1/k16-1029 doi: 10.18653/v1/k16-1029

Sengupta, S., Carastro, L., & Allen, P. E. (2005). Design considerations in bandgap references over process variations. In *International symposium on circuits and systems (ISCAS 2005), 23-26 may 2005, kobe, japan* (pp. 3869–3872). IEEE. Retrieved from https://doi.org/10.1109/ISCAS.2005.1465475 doi: 10.1109/ISCAS.2005.1465475

Shwartz-Ziv, R., & Tishby, N. (2017). Opening the black box of deep neural networks via information. *CoRR*, *abs/1703.00810*. Retrieved from http://arxiv.org/abs/1703.00810

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nat.*, *550*(7676), 354–359. Retrieved from https://doi.org/10.1038/nature24270 doi: 10.1038/nature24270

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In Y. Bengio & Y. LeCun (Eds.), *3rd international conference on learning representations, ICLR 2015, san diego, ca, usa, may 7-9, 2015, conference track proceedings*. Retrieved from http://arxiv.org/abs/1409.1556

Siswanto, A. E. (2021). *Block sparsity and weight initialization in neural network pruning* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Soelen, R. V., & Sheppard, J. W. (2019). Using winning lottery tickets in transfer learning for convolutional neural networks. In *International joint conference on neural networks, IJCNN 2019 budapest, hungary, july 14-19, 2019* (pp. 1–8). IEEE. Retrieved from https://doi.org/10.1109/IJCNN.2019.8852405 doi: 10.1109/IJCNN.2019.8852405

Sreenivasan, K., Rajput, S., Sohn, J.-Y., & Papailiopoulos, D. (2022, 28–30 Mar). Finding nearly everything within random binary networks. In G. Camps-Valls, F. J. R. Ruiz, & I. Valera (Eds.), *Proceedings of the 25th international conference on artificial intelligence and statistics* (Vol. 151, pp. 3531–3541). PMLR. Retrieved from https://proceedings.mlr.press/v151/sreenivasan22a.html

Sun, Y., Liang, D., Wang, X., & Tang, X. (2015). Deepid3: Face recognition with very deep neural networks. *CoRR*, *abs/1502.00873*. Retrieved from http://arxiv.org/abs/1502.00873

Sun, Z., & Huang, R. (2021). Time complexity of in-memory matrix-vector multiplication. *IEEE Trans. Circuits Syst. II Express Briefs*, *68*(8), 2785–2789. Retrieved from https://doi.org/10.1109/TCSII.2021.3068764 doi: 10.1109/TCSII.2021.3068764

Sun, Z.-W. (2003). Unification of zero-sum problems, subset sums and covers of z. *Electronic Research Announcements of The American Mathematical Society*, *9*, 51-60.

Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE conference on computer vision and pattern recognition, CVPR 2014, columbus, oh, usa, june 23-28, 2014* (pp. 1701–1708). IEEE Computer Society. Retrieved from https://doi.org/10.1109/CVPR.2014.220 doi: 10.1109/CVPR.2014.220

Talebbeydokhti, N., Hanumolu, P. K., Kurahashi, P., & Moon, U. (2006). Constant transconductance bias circuit with an on-chip resistor. In *International symposium on circuits and systems (ISCAS 2006), 21-24 may 2006, island of kos, greece*. IEEE. Retrieved from https://doi.org/10.1109/ISCAS.2006.1693220 doi: 10.1109/ISCAS.2006.1693220

Tishby, N., & Zaslavsky, N. (2015). Deep learning and the information bottleneck principle. In *2015 IEEE information theory workshop, ITW 2015, jerusalem, israel, april 26 - may 1, 2015* (pp. 1–5). IEEE. Retrieved from https://doi.org/10.1109/ITW.2015.7133169 doi: 10.1109/ITW.2015.7133169

Ulyanov, D., Vedaldi, A., & Lempitsky, V. S. (2020). Deep image prior. *Int. J. Comput. Vis.*, *128*(7), 1867–1888. Retrieved from https://doi.org/10.1007/s11263-020-01303-4 doi: 10.1007/s11263-020-01303-4

Vallapuram, A. K., Zhou, P., Kwon, Y. D., Lee, L. H., Xu, H., & Hui, P. (2022). Hidenseek: Federated lottery ticket via server-side pruning and sign supermask. *CoRR*, *abs/2206.04385*. Retrieved from https://doi.org/10.48550/arXiv.2206.04385 doi: 10.48550/arXiv.2206.04385

Vasilopoulos, A., Vitzilaios, G., Theodoratos, G., & Papananos, Y. (2006). A low-power wideband reconfigurable integrated active-rc filter with 73 db sfdr. *IEEE Journal of Solid-State Circuits*, *41*(9), 1997–2008.

Wang, C., Deng, J., Meng, X., Wang, Y., Li, J., Lin, S., ... Ding, C. (2021). A secure and efficient federated learning framework for NLP. In M. Moens, X. Huang, L. Specia, & S. W. Yih (Eds.), *Proceedings of the 2021 conference on empirical methods in natural language processing, EMNLP 2021, virtual event / punta cana, dominican republic, 7-11 november, 2021* (pp. 7676–7682). Association for Computational Linguistics. Retrieved from https://doi.org/10.18653/v1/2021.emnlp-main.606 doi: 10.18653/v1/2021.emnlp-main.606

Wang, Y., Ramon, J., & Guo, Z.-C. (2017, June). Learning from networked examples. *arXiv:1405.2600 [cs, stat]*. Retrieved 2022-04-26, from http://arxiv.org/abs/1405.2600 (arXiv: 1405.2600)

Wang, Y., Zhang, X., Xie, L., Zhou, J., Su, H., Zhang, B., & Hu, X. (2020). Pruning from scratch. In *The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, the thirty-second innovative applications of artificial intelligence conference, IAAI 2020, the tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020, new york, ny, usa, february 7-12, 2020* (pp. 12273–12280). AAAI Press. Retrieved from https://aaai.org/ojs/index.php/AAAI/article/view/6910

Warnke, L. (2016). On the method of typical bounded differences. *Combinatorics, Probability and Computing*, *25*(2), 269–299.

Wilkinson, J. H., & Reinsch, C. H. (1971). *Linear algebra* (Vol. 2). Springer. Retrieved from https://www.worldcat.org/oclc/00243328

Wortsman, M., Ramanujan, V., Liu, R., Kembhavi, A., Rastegari, M., Yosinski, J., & Farhadi, A. (2020). Supermasks in superposition. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems 33: Annual conference on neural information processing systems 2020, neurips 2020, december 6-12, 2020, virtual*. Retrieved from https://proceedings.neurips.cc/paper/2020/hash/ad1f8bb9b51f023cdc80cf94bb615aa9-Abstract.html

Wu, C., & Chou, C. (2001). The design of a CMOS IF bandpass amplifier with low sensitivity to process and temperature variations. In *Proceedings of the 2001 international symposium on circuits and systems, ISCAS 2001, sydney, australia, may 6-9, 2001* (pp. 121–124). IEEE. Retrieved from https://doi.org/10.1109/ISCAS.2001.921803 doi: 10.1109/ISCAS.2001.921803

Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, *abs/1708.07747*. Retrieved from http://arxiv.org/abs/1708.07747

Xiao, T. P., Bennett, C. H., Feinberg, B., Agarwal, S., & Marinella, M. J. (2020). Analog architectures for neural network acceleration based on non-volatile memory. *Applied Physics Reviews*, *7*(3).

Xiong, Z., Liao, F., & Kyrillidis, A. (2023). Strong lottery ticket hypothesis with ϵ-perturbation. In F. J. R. Ruiz, J. G. Dy, & J. van de Meent (Eds.), *International conference on artificial intelligence and statistics, 25-27 april 2023, palau de congressos, valencia, spain* (Vol. 206, pp. 6879–6902). PMLR. Retrieved from https://proceedings.mlr.press/v206/xiong23a.html

Yang, T., Chen, Y., & Sze, V. (2017). Designing energy-efficient convolutional neural networks using energy-aware pruning. In *2017 IEEE conference on computer vision and pattern recognition, CVPR 2017, honolulu, hi, usa, july 21-26, 2017* (pp. 6071–6079). IEEE Computer Society. Retrieved from https://doi.org/10.1109/CVPR.2017.643 doi: 10.1109/CVPR.2017.643

Yu, H., Edunov, S., Tian, Y., & Morcos, A. S. (2020). Playing the lottery with rewards and multiple languages: lottery tickets in RL and NLP. In *8th international conference on learning representations, ICLR 2020, addis ababa, ethiopia, april 26-30, 2020*. OpenReview.net. Retrieved from https://openreview.net/forum?id=S1xnXRVFwH

Yu, J., Lukefahr, A., Palframan, D. J., Dasika, G. S., Das, R., & Mahlke, S. A. (2017). Scalpel: Customizing DNN pruning to the underlying hardware parallelism. In *Proceedings of the 44th annual international symposium on computer architecture, ISCA 2017, toronto, on, canada, june 24-28, 2017* (pp. 548–560). ACM. Retrieved from https://dl.acm.org/citation.cfm?id=3080215

Zagoruyko, S., & Komodakis, N. (2016). Wide residual networks. In R. C. Wilson, E. R. Hancock, & W. A. P. Smith (Eds.), *Proceedings of the british machine vision conference 2016, BMVC 2016, york, uk, september 19-22, 2016*. BMVA Press. Retrieved from http://www.bmva.org/bmvc/2016/papers/paper087/index.html

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. In *5th international conference on learning representations, ICLR 2017, toulon, france, april 24-26, 2017, conference track proceedings*. OpenReview.net. Retrieved from https://openreview.net/forum?id=Sy8gdB9xx

Zhang, X., Ni, B., Mukhopadhyay, I., & Apsel, A. B. (2012). Improving absolute accuracy of integrated resistors with device diversification. *IEEE Trans. Circuits Syst. II Express Briefs*, *59-II*(6), 346–350. Retrieved from https://doi.org/10.1109/TCSII.2012.2195057 doi: 10.1109/TCSII.2012.2195057

Zhang, Y., Lin, M., Chao, F., Wang, Y., Wu, Y., Huang, F., ... Ji, R. (2021). Lottery jackpots exist in pre-trained models. *CoRR*, *abs/2104.08700*. Retrieved from https://arxiv.org/abs/2104.08700

Zhou, H., Lan, J., Liu, R., & Yosinski, J. (2019). Deconstructing lottery tickets: Zeros, signs, and the supermask. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32: Annual conference*

on neural information processing systems 2019, neurips 2019, december 8-14, 2019, vancouver, bc, canada (pp. 3592–3602). Retrieved from https://proceedings.neurips.cc/paper/2019/hash/1113d7a76ffceca1bb350bfe145467c6-Abstract.html

Symbols

$[n]$	$\{1, 2, \ldots, n\}$
$2^{\mathcal{S}}$	Power set: $\{\mathcal{T} \mid \mathcal{T} \subseteq \mathcal{S}\}$
$\binom{\mathcal{S}}{n}$	$\{\mathcal{T} \subseteq \mathcal{S} \mid \lvert\mathcal{T}\rvert = n\rvert\}$
$\mathrm{relu}(x)$	Rectified Linear Unit: $x \mapsto \max\{0, x\}$
$\mathrm{relu}(\boldsymbol{T})$	Entry-wise relu
$\mathrm{shape}(\boldsymbol{T})$	Shape of a tensor: given $\boldsymbol{T} \in \mathbb{R}^{d_1 \times \cdots \times d_n}$, we have $\mathrm{shape}(\boldsymbol{T}) = d_1 \times \cdots \times d_n$
$\odot$	Entry-wise (Hadamard) product
$*$	Convolution (see Definition 1.3.1)
$\boldsymbol{I}_d$	$d \times d$ identity matrix
$\boldsymbol{I}$	Identity matrix with dimensionality implied by the context
$\lVert\boldsymbol{T}\rVert_p$	p-norm: $\left(\sum_{\boldsymbol{i}} \lvert T_{\boldsymbol{i}}\rvert^p\right)^{\frac{1}{p}}$
$\lVert\boldsymbol{T}\rVert_\infty$	Maximum norm: $\max_{\boldsymbol{i}} \lvert T_{\boldsymbol{i}}\rvert$
$\lVert\boldsymbol{M}\rVert_{\text{spectral}}$	Spectral norm: $\sup_{\boldsymbol{x} \in \mathbb{R}^d : \lVert\boldsymbol{x}\rVert_2 \leq 1} \lVert\boldsymbol{M}\boldsymbol{x}\rVert_2$
$\mathfrak{B}_\infty(\boldsymbol{x}, r)$	d-dimensional hypercube of radius r centred at $\boldsymbol{x}$: $\{\boldsymbol{y} \in \mathbb{R}^d : \lVert\boldsymbol{y} - \boldsymbol{x}\rVert_\infty \leq r\}$
$\mathfrak{B}_\infty(r)$	$\mathfrak{B}_\infty(\boldsymbol{0}, r)$
$\mathrm{Unif}(\mathcal{D})$	Uniform distribution over a set $\mathcal{D}$
$\mathrm{Bern}(p)$	Bernoulli distribution with probability of success equal to p
$\mathcal{N}(\mu, \sigma^2)$	Normal distribution with mean μ and variance σ^2
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
φ_{X}	Probability density function of X
$f(\,\cdot\,;\,\cdot\,)$	Neural network f without specific parameterization (architecture)
$f(\,\cdot\,;\boldsymbol{\theta})$	Neural network f with parameters $\boldsymbol{\theta}$
$f(\,\cdot\,)$	Neural network f with parameters implied by the context
$\mathfrak{Prune}(f)$	Class of all subnetworks of f (see page 5)
$\mathfrak{FilterPrune}(f)$	Class of all neuron/filter-subnetworks of f (see page 24)

List of Figures

Appendix

Appendices of Chapter 3

A Tools

Below we list some standard tools we use, and prove some inequalities.

A.1 Concentration bounds

Theorem A.1 (Chebyshev's inequality)**.** *Let* X *be a random variable with finite expected value* μ *and finite non-zero variance* σ^2*. Then for any real number* $k > 0$*, it holds that*

$$\Pr[|\mathrm{X} - \mu| \geq k] \leq \frac{\sigma^2}{k^2}.$$

Lemma A.2 (Chernoff-Hoeffding bounds (Doerr, 2011))**.** *Let* $\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_n$ *be independent random variables such that*

$$\Pr[0 \leq \mathrm{X}_i \leq 1] = 1$$

for all $i \in [n]$*. Let* $\mathrm{X} = \sum_{i=1}^{n} \mathrm{X}_i$ *and* $\mathbb{E}[\mathrm{X}] = \mu$*. Then, for any* $\delta \in (0, 1]$ *the following holds:*

1. $\Pr[\mathrm{X} \geq (1+\delta)\mu] \leq \exp\left(-\frac{\delta^2 \mu_+}{3}\right)$;
2. $\Pr[\mathrm{X} \leq (1-\delta)\mu] \leq \exp\left(-\frac{\delta^2 \mu_+}{2}\right)$.

A.2 Claims

Claim A.3. *Let* $d, n \in \mathbb{N}$*,* $\varepsilon \in (0, 1)$*, and* $\alpha \in (0, \frac{1}{2}]$*. If* $n \geq \frac{4d}{\alpha \log \frac{1}{\alpha}} \left[\log \frac{1}{\varepsilon} + \log 2\pi d\right] + \frac{8}{\alpha}$*, then*

$$\frac{\alpha^{\alpha n} \cdot \exp\left[\frac{4d}{\alpha n}\right] \cdot (2\pi\alpha n)^{\frac{d}{2}}}{(2\varepsilon)^d} \leq \varepsilon.$$

Proof. Consider the function

$$f(n) = \alpha^{\frac{\alpha n}{2}} \cdot (2\pi\alpha n)^{\frac{d}{2}}.$$

We have that

$$\begin{aligned} f'(n) &= \alpha^{\frac{\alpha n}{2}} \cdot \frac{\alpha \ln \alpha}{2} \cdot (2\pi\alpha n)^{\frac{d}{2}} + \alpha^{\frac{\alpha n}{2}} \cdot (2\pi\alpha n)^{\frac{d}{2}-1} \cdot \pi\alpha d \\ &= \alpha^{\frac{\alpha n}{2}} \cdot (2\pi\alpha n)^{\frac{d}{2}-1} \cdot (\pi\alpha) \cdot [\alpha n \ln \alpha + d], \end{aligned}$$

which is non-positive for $n \geq \frac{d}{\alpha \ln \frac{1}{\alpha}}$. Let $\bar{n} = \frac{d}{\alpha \ln \frac{1}{\alpha}}$; it holds that $f(n) \leq f(\bar{n})$ for $n \geq \bar{n}$. Hence, for $n \geq \bar{n}$,

$$\begin{aligned}\frac{\alpha^{\frac{\alpha n}{2}} \cdot \exp\left[\frac{4d}{\alpha n}\right] f(n)}{(2\varepsilon)^d} &\leq \frac{\alpha^{\frac{\alpha n}{2}} \cdot \exp\left[\frac{4d}{\alpha n}\right] f(\bar{n})}{(2\varepsilon)^d} \\ &= \frac{\alpha^{\frac{\alpha n}{2}} \cdot \exp\left[\frac{4d}{\alpha n} - \frac{d}{2}\right]}{(2\varepsilon)^d} \cdot \left[\frac{2\pi d}{\ln \frac{1}{\alpha}}\right]^{\frac{d}{2}} \\ &\leq \frac{\alpha^{\frac{\alpha n}{2}}}{(2\varepsilon)^d \cdot \alpha^4} \cdot (4\pi d)^{\frac{d}{2}},\end{aligned}$$

where the latter inequality holds since $\ln \frac{1}{\alpha} \geq \frac{1}{2}$. Now, it holds that $\frac{\alpha^{\frac{\alpha n}{2}}}{(2\varepsilon)^d \cdot \alpha^4} \cdot (4\pi d)^{\frac{d}{2}} \leq \varepsilon$ whenever

$$\begin{aligned}n &\geq \frac{2 \ln \frac{(2\varepsilon)^d \varepsilon \cdot \alpha^4}{(2\pi d)^{\frac{d}{2}}}}{-\alpha \ln \frac{1}{\alpha}} \\ &= \frac{2\left(-d \ln 2 + (d+1) \ln \frac{1}{\varepsilon} + 4 \ln \frac{1}{\alpha} + \frac{d}{2} \ln(2\pi) + \frac{d}{2} \ln d\right)}{\alpha \ln \frac{1}{\alpha}}.\end{aligned}$$

The latter condition is achieved for

$$n \geq \frac{4d}{\alpha \log \frac{1}{\alpha}} \left[\log \frac{1}{\varepsilon} + \log 2\pi d\right] + \frac{8}{\alpha}.$$

□

Claim A.4. *Let $d, n \in \mathbb{N}$ and $\alpha \in \mathbb{R}_{>0}$. If $n \geq \frac{68d}{\alpha}$ and $\alpha \leq \frac{1}{6\sqrt{d}}$, then*

$$e^{\frac{4d}{\alpha n}} \cdot \frac{1}{(1-4\alpha^2)^{\frac{d}{2}}} \leq 1 + \frac{1}{8}.$$

Proof. Since $e^x \leq (1-x)^{-1}$ for $x \leq 1$, for $n \geq \frac{4d}{\alpha}$, it holds that

$$e^{\frac{4d}{\alpha n}} \leq \frac{1}{1 - \frac{4d}{\alpha n}} = 1 + \frac{4d}{\alpha n - 4d}.$$

Thus, having $n \geq \frac{68d}{\alpha}$ implies that

$$e^{\frac{4d}{\alpha n}} \leq 1 + \frac{1}{16}.$$

Moreover, by Bernoulli's inequality, since $\alpha < \frac{1}{2}$, it holds that,

$$\frac{1}{(1-4\alpha^2)^{\frac{d}{2}}} \leq \frac{1}{1-2d\alpha^2}.$$

Altogether, we need that

$$\frac{1 + \frac{1}{16}}{1 - 2d\alpha^2} \leq 1 + \frac{1}{8},$$

which holds for $\alpha \leq \frac{1}{6\sqrt{d}}$. □

Claim A.5. *Let* A, B *be two centred normal random variables, and let* $\varphi_{\mathrm{B}}(x)$ *be the density function of* B*. Then, for any* $z \in \mathbb{R}$*, for any* $\varepsilon > 0$*, it holds that*

$$\int_{\mathbb{R}} \varphi_{\mathrm{B}}(x) \left[\Pr[\mathrm{A} \in (z-x-\varepsilon, z-x+\varepsilon)]\right]^2 \,\mathrm{d}x \leq \int_{\mathbb{R}} \varphi_{\mathrm{B}}(x) \left[\Pr[\mathrm{A} \in (-x-\varepsilon, -x+\varepsilon)]\right]^2 \,\mathrm{d}x.$$

Proof. For any $x, z \in \mathbb{R}$, let

$$h(x,z) = \varphi_{\mathrm{B}}(x) \left[\Pr[\mathrm{A} \in (z-x-\varepsilon, z-x+\varepsilon)]\right]^2 \,\mathrm{d}x,$$

and let

$$H(z) = \int_{\mathbb{R}} h(x,z) \,\mathrm{d}x.$$

Let $\varphi_{\mathrm{A}}(x)$ be the density function of a. Since

$$\begin{aligned} \left|\frac{\partial h(x,z)}{\partial z}\right| &= 2\left|\varphi_{\mathrm{B}}(x) \Pr[\mathrm{A} \in (z-x-\varepsilon, z-x+\varepsilon)] \left(\varphi_{\mathrm{A}}(z-x+\varepsilon) - \varphi_{\mathrm{A}}(z-x-\varepsilon)\right)\right| \\ &\leq 2\varphi_{\mathrm{B}}(x) \Pr[\mathrm{A} \in (z-x-\varepsilon, z-x+\varepsilon)] \left(\varphi_{\mathrm{A}}(z-x+\varepsilon) + \varphi_{\mathrm{A}}(z-x-\varepsilon)\right), \end{aligned}$$

$h(x,z)$ meets the hypothesis of the Leibniz integral rule and we can write

$$\begin{aligned} \frac{\mathrm{d}H(z)}{\mathrm{d}z} &= \int_{\mathbb{R}} \frac{\partial h(x,z)}{\partial z} \,\mathrm{d}x \\ &= 2\int_{\mathbb{R}} \varphi_{\mathrm{B}}(x) \Pr[\mathrm{A} \in (z-x-\varepsilon, z-x+\varepsilon)] \left(\varphi_{\mathrm{A}}(z-x+\varepsilon) - \varphi_{\mathrm{A}}(z-x-\varepsilon)\right) \,\mathrm{d}x \end{aligned}$$

If we prove that such a function is zero in $z = 0$, positive for $z < 0$ and negative for $z > 0$, then we have that the maximum of H is reached in $z = 0$.

First case: $z = 0$**.** Then

$$\begin{aligned} \frac{\mathrm{d}H(0)}{\mathrm{d}z} &= 2\int_{\mathbb{R}} \varphi_{\mathrm{B}}(x) \Pr[\mathrm{A} \in (x-\varepsilon, x+\varepsilon)] \left(\varphi_{\mathrm{A}}(x-\varepsilon) - \varphi_{\mathrm{A}}(x+\varepsilon)\right) \,\mathrm{d}x \qquad \text{(A.2)} \\ &= 2\int_{\mathbb{R}} \varphi_{\mathrm{B}}(x) \Pr[\mathrm{A} \in (x-\varepsilon, x+\varepsilon)] \varphi_{\mathrm{A}}(x-\varepsilon) \,\mathrm{d}x \\ &\quad - 2\int_{\mathbb{R}} \varphi_{\mathrm{B}}(x) \Pr[\mathrm{A} \in (x-\varepsilon, x+\varepsilon)] \varphi_{\mathrm{A}}(x+\varepsilon) \,\mathrm{d}x \\ &= 2\int_{\mathbb{R}} \varphi_{\mathrm{B}}(x) \Pr[\mathrm{A} \in (x-\varepsilon, x+\varepsilon)] \varphi_{\mathrm{A}}(x-\varepsilon) \,\mathrm{d}x \\ &\quad - 2\int_{\mathbb{R}} \varphi_{\mathrm{B}}(y) \Pr[\mathrm{A} \in (y-\varepsilon, y+\varepsilon)] \varphi_{\mathrm{A}}(y-\varepsilon) \,\mathrm{d}x \qquad \text{(A.3)} \\ &= 0, \end{aligned}$$

where in equation (A.2) we exploited the symmetry of the integrand functions, equation (A.3) we substituted in the second integral $y = -x$ and used again symmetry.

Second case: $z > 0$. Then

$$
\begin{aligned}
&\frac{\mathrm{d}H(z)}{\mathrm{d}z} \\
&= 2\int_{\mathbb{R}} \varphi_{\mathrm{B}}(x)\Pr[\mathrm{A}\in(z-x-\varepsilon,z-x+\varepsilon)]\left(\varphi_{\mathrm{A}}(z-x+\varepsilon)-\varphi_{\mathrm{A}}(z-x-\varepsilon)\right)\mathrm{d}x \\
&= 2\int_{-\infty}^{-z} \varphi_{\mathrm{B}}(x)\Pr[\mathrm{A}\in(z-x-\varepsilon,z-x+\varepsilon)]\left(\varphi_{\mathrm{A}}(z-x+\varepsilon)-\varphi_{\mathrm{A}}(z-x-\varepsilon)\right)\mathrm{d}x \\
&\quad + 2\int_{-z}^{+z} \varphi_{\mathrm{B}}(x)\Pr[\mathrm{A}\in(z-x-\varepsilon,z-x+\varepsilon)]\left(\varphi_{\mathrm{A}}(z-x+\varepsilon)-\varphi_{\mathrm{A}}(z-x-\varepsilon)\right)\mathrm{d}x \\
&\quad + 2\int_{+z}^{+\infty} \varphi_{\mathrm{B}}(x)\Pr[\mathrm{A}\in(z-x-\varepsilon,z-x+\varepsilon)]\left(\varphi_{\mathrm{A}}(z-x+\varepsilon)-\varphi_{\mathrm{A}}(z-x-\varepsilon)\right)\mathrm{d}x \\
&= 2\int_{+z}^{+\infty} \varphi_{\mathrm{B}}(x)\Pr[\mathrm{A}\in(z+x-\varepsilon,z+x+\varepsilon)]\left(\varphi_{\mathrm{A}}(z+x+\varepsilon)-\varphi_{\mathrm{A}}(z+x-\varepsilon)\right)\mathrm{d}x \qquad \text{(A.4)} \\
&\quad + 2\int_{+3z}^{+\infty} \varphi_{\mathrm{B}}(x)\Pr[\mathrm{A}\in(z-x-\varepsilon,z-x+\varepsilon)]\left(\varphi_{\mathrm{A}}(z-x+\varepsilon)-\varphi_{\mathrm{A}}(z-x-\varepsilon)\right)\mathrm{d}x \\
&\quad + 2\int_{+z}^{+3z} \varphi_{\mathrm{B}}(x)\Pr[\mathrm{A}\in(z-x-\varepsilon,z-x+\varepsilon)]\left(\varphi_{\mathrm{A}}(z-x+\varepsilon)-\varphi_{\mathrm{A}}(z-x-\varepsilon)\right)\mathrm{d}x \\
&\quad + 2\int_{-z}^{+z} \varphi_{\mathrm{B}}(x)\Pr[\mathrm{A}\in(z-x-\varepsilon,z-x+\varepsilon)]\left(\varphi_{\mathrm{A}}(z-x+\varepsilon)-\varphi_{\mathrm{A}}(z-x-\varepsilon)\right)\mathrm{d}x \\
&= 2\int_{+z}^{+\infty} \varphi_{\mathrm{B}}(x)\Pr[\mathrm{A}\in(z+x-\varepsilon,z+x+\varepsilon)]\left(\varphi_{\mathrm{A}}(z+x+\varepsilon)-\varphi_{\mathrm{A}}(z+x-\varepsilon)\right)\mathrm{d}x \\
&\quad - 2\int_{+z}^{+\infty} \varphi_{\mathrm{B}}(2z+x)\Pr[\mathrm{A}\in(z+x-\varepsilon,z+x+\varepsilon)]\left(\varphi_{\mathrm{A}}(z+x+\varepsilon)-\varphi_{\mathrm{A}}(z+x-\varepsilon)\right)\mathrm{d}x \qquad \text{(A.5)} \\
&\quad - 2\int_{-z}^{+z} \varphi_{\mathrm{B}}(x-2z)\Pr[\mathrm{A}\in(z-x-\varepsilon,z-x+\varepsilon)]\left(\varphi_{\mathrm{A}}(z-x+\varepsilon)-\varphi_{\mathrm{A}}(z-x-\varepsilon)\right)\mathrm{d}x \qquad \text{(A.6)} \\
&\quad + 2\int_{-z}^{+z} \varphi_{\mathrm{B}}(x)\Pr[\mathrm{A}\in(z-x-\varepsilon,z-x+\varepsilon)]\left(\varphi_{\mathrm{A}}(z-x+\varepsilon)-\varphi_{\mathrm{A}}(z-x-\varepsilon)\right)\mathrm{d}x \\
&= 2\int_{+z}^{+\infty} \left(\varphi_{\mathrm{B}}(x)-\varphi_{\mathrm{B}}(2z+x)\right)\Pr[\mathrm{A}\in(z+x-\varepsilon,z+x+\varepsilon)]\left(\varphi_{\mathrm{A}}(z+x+\varepsilon)-\varphi_{\mathrm{A}}(z+x-\varepsilon)\right)\mathrm{d}x \qquad \text{(A.7)} \\
&\quad + 2\int_{-z}^{+z} \left(\varphi_{\mathrm{B}}(x)-\varphi_{\mathrm{B}}(x-2z)\right)\Pr[\mathrm{A}\in(z-x-\varepsilon,z-x+\varepsilon)]\left(\varphi_{\mathrm{A}}(z-x+\varepsilon)-\varphi_{\mathrm{A}}(z-x-\varepsilon)\right)\mathrm{d}x, \qquad \text{(A.8)}
\end{aligned}
$$

where in equation (A.4) we substituted $x' = -x$ and used the symmetry of the integrand functions, in equations (A.5) and (A.6) we substituted $x' = x - 2z$ and $x' = 2z - x$, respectively, and used again the symmetry. The expression in equation (A.7) is negative as $\varphi_{\mathrm{B}}(x) > \varphi_{\mathrm{B}}(2z+x)$ and $\varphi_{\mathrm{A}}(z+x+\varepsilon) < \varphi_{\mathrm{A}}(z+x-\varepsilon)$ for $x \geq z$; the expression in equation (A.8) is negative as $\varphi_{\mathrm{B}}(x) > \varphi_{\mathrm{B}}(x-2z)$ and $\varphi_{\mathrm{A}}(z-x+\varepsilon) < \varphi_{\mathrm{A}}(z-x-\varepsilon)$ for $x \in (-z, z)$.

Third case: $z < 0$**.** This case is similar to the previous one: with the same arguments, we obtain

$$\begin{aligned}
&\frac{\mathrm{d}H(z)}{\mathrm{d}z} \\
&= 2\int_{-\infty}^{+z} (\varphi_{\mathrm{B}}(x) - \varphi_{\mathrm{B}}(2z+x))\Pr[\mathrm{A} \in (z+x-\varepsilon, z+x+\varepsilon)]\,(\varphi_{\mathrm{A}}(z+x+\varepsilon) - \varphi_{\mathrm{A}}(z+x-\varepsilon))\,\mathrm{d}x \qquad \text{(A.9)} \\
&\quad + 2\int_{+z}^{-z} (\varphi_{\mathrm{B}}(x) - \varphi_{\mathrm{B}}(x-2z))\Pr[\mathrm{A} \in (z-x-\varepsilon, z-x+\varepsilon)]\,(\varphi_{\mathrm{A}}(z-x+\varepsilon) - \varphi_{\mathrm{A}}(z-x-\varepsilon))\,\mathrm{d}x. \qquad \text{(A.10)}
\end{aligned}$$

The expression in equation (A.9) is positive as $\varphi_{\mathrm{B}}(x) > \varphi_{\mathrm{B}}(2z+x)$ and $\varphi_{\mathrm{A}}(z+x+\varepsilon) > \varphi_{\mathrm{A}}(z+x-\varepsilon)$ for $x \le z$; the expression in equation (A.10) is positive as $\varphi_{\mathrm{B}}(x) > \varphi_{\mathrm{B}}(x-2z)$ and $\varphi_{\mathrm{A}}(z-x+\varepsilon) < \varphi_{\mathrm{A}}(z-x-\varepsilon)$ for $x \in (z,-z)$. □

Claim A.6. *For all* $x \in \mathbb{R}$, $c \in \left(0, \frac{1}{162}\right)$, *and* $\varepsilon \in (0,1)$, *it holds that*

$$\left(\int_{-\varepsilon}^{\varepsilon} e^{-c(x+s)^2}\,\mathrm{d}s\right)^2 \le \left(\int_{-\varepsilon}^{\varepsilon} \frac{e^{-c(x+\varepsilon)^2} + e^{-c(x-\varepsilon)^2}}{2} e^{c\varepsilon^2}\,\mathrm{d}s\right)^2.$$

Proof. Let

$$f_x(s) = e^{-c(x+s)^2}.$$

Since

$$\int_{-\varepsilon}^{\varepsilon} \frac{e^{-c(x+\varepsilon)^2} + e^{-c(x-\varepsilon)^2}}{2} e^{c\varepsilon^2}\,\mathrm{d}s = \int_{-\varepsilon}^{\varepsilon} ms + \frac{e^{-c(x+\varepsilon)^2} + e^{-c(x-\varepsilon)^2}}{2} e^{c\varepsilon^2}\,\mathrm{d}s$$

for any $m \in \mathbb{R}$, we choose it to be the angular coefficient of the line passing through $f_x(-\varepsilon)$ and $f_x(\varepsilon)$, and prove the stronger result

$$e^{-c(x+s)^2} \le \frac{e^{-c(x+\varepsilon)^2} - e^{-c(x-\varepsilon)^2}}{2\varepsilon} s + \frac{e^{-c(x+\varepsilon)^2} + e^{-c(x-\varepsilon)^2}}{2} e^{c\varepsilon^2} \qquad \text{(A.11)}$$

for all $s \in (-\varepsilon, \varepsilon)$. In fact, the right-hand side of equation (A.11) is the equation for the line passing by the extrema of f_x in $(-\varepsilon, \varepsilon)$ lifted by a factor of $e^{c\varepsilon^2}$. Therefore, the results holds trivially if f_x is convex in the entire range $(-\varepsilon, \varepsilon)$, which is true when $|x| > 1 + \frac{1}{\sqrt{2c}}$. Moreover, the factor $e^{c\varepsilon^2}$ ensures the result for $x = 0$, so, we follow with the analysis of the case $x \in \left(0, 1 + \frac{1}{\sqrt{2c}}\right]$ and the remaining case $x \in \left[-1 - \frac{1}{\sqrt{2c}}, 0\right)$ follows by symmetry.

Dividing both sides of equation (A.11) by $e^{-c(x+s)^2}$, we obtain

$$\begin{aligned}
1 &\le e^{2csx+cs^2}\left[\frac{e^{-c\varepsilon^2}s}{\varepsilon} \cdot \frac{e^{-2c\varepsilon x} - e^{2c\varepsilon x}}{2} + \frac{e^{-2c\varepsilon x} + e^{2c\varepsilon x}}{2}\right] \qquad \text{(A.12)} \\
&= e^{2csx+cs^2}\left[-\frac{e^{-c\varepsilon^2}s}{\varepsilon}\sinh(2c\varepsilon x) + \cosh(2c\varepsilon x)\right].
\end{aligned}$$

Let $g(x)$ be the right-hand side of this inequality. Then

$$\begin{aligned} g'(x) &= 2csg(x) + 2c\varepsilon e^{2csx+cs^2}\left[-\frac{e^{-c\varepsilon^2}s}{\varepsilon}\cosh(2c\varepsilon x) + \sinh(2c\varepsilon x)\right] \\ &= 2ce^{2csx+cs^2}\left[\cosh(2c\varepsilon x)\left(s - se^{-c\varepsilon^2}\right) + \sinh(2c\varepsilon x)\left(\varepsilon - \frac{s^2}{\varepsilon}e^{-c\varepsilon^2}\right)\right]. \end{aligned}$$

If $s \in [0, \varepsilon)$, then $s \geq se^{-c\varepsilon^2}$ and $\varepsilon \geq \frac{\varepsilon^2}{\varepsilon}e^{-c\varepsilon^2} \geq \frac{s^2}{\varepsilon}e^{-c\varepsilon^2}$, hence $g'(x) \geq 0$. Since $g(0) \geq 1$, this ensures equation (A.12).

The sub-case $s \in (-\varepsilon, 0)$ offers much more resistance. To analyse it we exploit that $x \in \left(0, 1 + \frac{1}{\sqrt{2c}}\right)$ implies that $cx \leq \sqrt{2c}$ for $c < \frac{1}{2}$ and make extensive use of Taylor's theorem to approximate the exponential functions.

We start by rewriting equation (A.12) as

$$\varepsilon e^{-2csx-cs^2} \leq e^{2c\varepsilon x}\left(\frac{\varepsilon}{2} - \frac{s}{2}e^{-c\varepsilon^2}\right) + e^{-2c\varepsilon x}\left(\frac{\varepsilon}{2} + \frac{s}{2}e^{-c\varepsilon^2}\right). \tag{A.13}$$

By Taylor's theorem, there exist $\lambda_1, \lambda_2 \in [0, 2c\varepsilon x] \subseteq [0, 2\sqrt{2c}\varepsilon]$, $\lambda_3 \in [0, -2csx] \subseteq [0, 2\sqrt{2c}\varepsilon]$, $\lambda_4 \in [0, cs^2]$, $\lambda_5 \in [0, c\varepsilon^2]$ such that

$$\begin{aligned} e^{+2c\varepsilon x} &= 1 + 2c\varepsilon x + 2c^2\varepsilon^2x^2 + \frac{4}{3}c^3\varepsilon^3x^3e^{\lambda_1}, \\ e^{-2c\varepsilon x} &= 1 - 2c\varepsilon x + 2c^2\varepsilon^2x^2 - \frac{4}{3}c^3\varepsilon^3x^3e^{\lambda_2}, \\ e^{-2csx} &= 1 - 2csx + 2c^2s^2x^2 - \frac{4}{3}c^3s^3x^3e^{\lambda_3}, \\ e^{-cs^2} &= 1 - cs^2 + \frac{c^2s^4}{2}e^{-\lambda_4}, \qquad (A.14) \\ e^{-c\varepsilon^2} &= 1 - c\varepsilon^2 + \frac{c^2\varepsilon^4}{2}e^{-\lambda_5}, \qquad (A.15) \end{aligned}$$

where we used second order approximations for the first three terms and first order approximations for the last two. Plugging those in equation (A.13) we obtain

$$\begin{aligned} \varepsilon e^{-cs^2}(1 - 2csx + 2c^2s^2x^2 - \frac{4}{3}c^3s^3x^3e^{\lambda_3}) \leq (1 + 2c\varepsilon x + 2c^2\varepsilon^2x^2 + \frac{4}{3}c^3\varepsilon^3x^3e^{\lambda_1})(\frac{\varepsilon}{2} - \frac{s}{2}e^{-c\varepsilon^2}) \\ + (1 - 2c\varepsilon x + 2c^2\varepsilon^2x^2 - \frac{4}{3}c^3\varepsilon^3x^3e^{\lambda_2})(\frac{\varepsilon}{2} + \frac{s}{2}e^{-c\varepsilon^2}). \end{aligned}$$

The latter becomes

$$\begin{aligned} &\varepsilon\left(1 - e^{-cs^2}\right) + 2cs\varepsilon x\left(e^{-cs^2} - e^{-c\varepsilon^2}\right) + 2c^2\varepsilon x^2\left(\varepsilon^2 - s^2e^{-cs^2}\right) \\ &+ \frac{4}{3}c^3\varepsilon x^3\left(\varepsilon^2e^{\lambda_1}\left(\frac{\varepsilon}{2} - \frac{s}{2}e^{-c\varepsilon^2}\right) - \varepsilon^2e^{-\lambda_2}\left(\frac{\varepsilon}{2} + \frac{s}{2}e^{-c\varepsilon^2}\right) + s^3e^{\lambda_3 - cs^2}\right) \geq 0 \end{aligned}$$

Now, notice that

$$\varepsilon^2e^{\lambda_1}\left(\frac{\varepsilon}{2} - \frac{s}{2}e^{-c\varepsilon^2}\right) - \varepsilon^2e^{-\lambda_2}\left(\frac{\varepsilon}{2} + \frac{s}{2}e^{-c\varepsilon^2}\right) \geq 0,$$

as $\frac{\varepsilon}{2} - \frac{s}{2}e^{-c\varepsilon^2} \geq \frac{\varepsilon}{2} + \frac{s}{2}e^{-c\varepsilon^2}$ since $-\varepsilon \leq s < 0$, and $\varepsilon^2 e^{\lambda_1} \geq \varepsilon^2 \geq \varepsilon^2 e^{-\lambda_2}$. Furthermore, observe that $s^3 e^{\lambda_3 - cs^2} \geq 2s^3$ as $s < 0$ and $\lambda_3 \leq 2\sqrt{2c}\varepsilon \leq \frac{1}{2}$ if $c \leq \frac{1}{32}$. Thus, the inequality is true if

$$\varepsilon\left(1 - e^{-cs^2}\right) + 2cs\varepsilon x\left(e^{-cs^2} - e^{-c\varepsilon^2}\right) + 2c^2\varepsilon x^2\left(\varepsilon^2 - s^2 e^{-cs^2}\right) + \frac{8}{3}c^3 s^3 \varepsilon x^3 \geq 0.$$

Applying equations (A.14) and (A.15), the latter inequality yields that

$$\begin{aligned}
&\varepsilon\left(cs^2 - \frac{c^2 s^4}{2}e^{-\lambda_4}\right) + 2cs\varepsilon x\left(c\varepsilon^2 - cs^2 - \frac{c^2\varepsilon^4}{2}e^{-\lambda_5} + \frac{c^2 s^4}{2}e^{-\lambda_4}\right) \\
&\qquad + 2c^2\varepsilon x^2\left(\varepsilon^2 - s^2 + cs^4 - \frac{c^2 s^6}{2}e^{-\lambda_4}\right) + \frac{8}{3}c^3 s^3 \varepsilon x^3 \\
&= \varepsilon cs^2 - \frac{c^2 s^4 \varepsilon}{2}e^{-\lambda_4} - c^3 s\varepsilon^5 x e^{-\lambda_5} + c^3 s^5 \varepsilon x e^{-\lambda_4} + \left(2c^3 s^4 \varepsilon x^2 - c^4 s^6 \varepsilon x^2 e^{-\lambda_4}\right) + \frac{8}{3}c^3 s^3 \varepsilon x^3 \\
&\qquad + 2c^2\varepsilon x\left(\varepsilon^2 - s^2\right)(x + s).
\end{aligned}$$

Now observe that

$$\left(2c^3 s^4 \varepsilon x^2 - c^4 s^6 \varepsilon x^2 e^{-\lambda_4}\right) \geq 0$$

as $c < 1, s \leq \varepsilon \leq 1, e^{-\lambda_4} < 1$; $-c^3 s\varepsilon^5 x e^{-\lambda_5} > 0$ as $s < 0$;

$$\begin{aligned}
\varepsilon cs^2 - \frac{c^2 s^4 \varepsilon}{2}e^{-\lambda_4} + c^3 s^5 \varepsilon x e^{-\lambda_4} + \frac{8}{3}c^3 s^3 \varepsilon x^3 &\geq cs^2\varepsilon - \frac{c^2 s^2 \varepsilon^3}{2} - c^2\sqrt{2c}s^2\varepsilon^4 - \frac{8}{3}c^3 s^2 \varepsilon^2 x^3 \\
&> cs^2\varepsilon - \frac{c^2 s^2 \varepsilon^3}{2} - c^2\sqrt{2c}s^2\varepsilon^4 - 6c\sqrt{2c}\varepsilon^3 x^3 \qquad \text{(A.16)} \\
&= cs^2\varepsilon\left(1 - \frac{c\varepsilon^2}{2} - c^2\sqrt{2c}\varepsilon^3 - 6\sqrt{2c}\varepsilon\right) \\
&\geq cs^2\varepsilon\left(1 - \frac{c}{2} - c^2\sqrt{2c} - 6\sqrt{2c}\right) \qquad \text{(A.17)} \\
&\geq \frac{cs^2\varepsilon}{5}, \qquad \text{(A.18)}
\end{aligned}$$

where in equation (A.16) we used that $cx \leq \sqrt{2c}$, in equation (A.17) that $\varepsilon \leq 1$, and in equation (A.18) we used i) $c^2\sqrt{2c} \leq \frac{c}{2}$ when $c \leq \frac{1}{\sqrt[3]{2}}$, ii) $c < \sqrt{2c}$ since $c < 1$ and iii) $1 - 7\sqrt{2c} \geq \frac{1}{5}$, whenever $c \leq \frac{1}{162}$.

Going back to the inequality, we now have that

$$\frac{cs^2\varepsilon}{5} + 2c^2\varepsilon x\left(\varepsilon^2 - s^2\right)(x + s).$$

If $x \geq |s|$ the latter is positive, otherwise it becomes

$$\begin{aligned}
\frac{cs^2\varepsilon}{5} + 2c^2\varepsilon x\left(\varepsilon^2 - s^2\right)(x + s) &\geq \frac{cs^2\varepsilon}{5} + 2c^2\varepsilon x^2\left(\varepsilon^2 - s^2\right) - 2c^2\varepsilon s^2\left(\varepsilon^2 - s^2\right) \\
&\geq \frac{cs^2\varepsilon}{5} - 2c^2\varepsilon s^2 + 2c^2\varepsilon x^2\left(\varepsilon^2 - s^2\right) \\
&\geq cs^2\varepsilon\left(\frac{1}{5} - 2c\right),
\end{aligned}$$

which is positive for $c < \frac{1}{10}$. □

Claim A.7. *For all $x \in \mathbb{R}$, $c \in \left(0, \frac{1}{10}\right)$, and $\varepsilon \in (0,1)$, it holds that*

$$\left(\int_{x-\varepsilon}^{x+\varepsilon} \exp\left(-cy^2\right)\,\mathrm{d}y\right)^2 \geq \int_{x-\varepsilon}^{x+\varepsilon} \exp\left(-c(x-\varepsilon)^2\right)\,\mathrm{d}y \cdot \int_{x-\varepsilon}^{x+\varepsilon} \exp\left(-c(x+\varepsilon)^2\right)\,\mathrm{d}y. \tag{A.19}$$

Proof. We can express equation (A.19) as

$$\begin{aligned}
&\left[\int_{x-\varepsilon}^{x+\varepsilon} \exp\left(-cy^2\right)\,\mathrm{d}y\right]^2 - \left[\int_{x-\varepsilon}^{x+\varepsilon} \exp\left(-c(x^2+\varepsilon^2)\right)\,\mathrm{d}y\right]^2 \\
&\quad = \left[\int_{x-\varepsilon}^{x+\varepsilon} \exp\left(-cy^2\right) - \exp\left(-c(x^2+\varepsilon^2)\right)\,\mathrm{d}y\right] \cdot \left[\int_{x-\varepsilon}^{x+\varepsilon} \exp\left(-cy^2\right) + \exp\left(-c(x^2+\varepsilon^2)\right)\,\mathrm{d}y\right] \\
&\quad \geq 0,
\end{aligned}$$

which holds if and only if

$$\int_{-\varepsilon}^{+\varepsilon} \exp\left(-c(x+s)^2\right)\,\mathrm{d}s \geq \int_{-\varepsilon}^{+\varepsilon} \exp\left(-c(x^2+\varepsilon^2)\right)\,\mathrm{d}s. \tag{A.20}$$

The result is immediate for $x = 0$, so we assume $x > 0$ and the claim follows by symmetry. Let

$$f_x(s) = \exp(-c(x+s)^2).$$

We provide distinct arguments depending on whether x is small or large.

Case $x \in (0,1)$. Since we assume $c < \frac{1}{8}$ and $\varepsilon < 1$, we have for any $x \leq 1$ that f_x is concave in $(-\varepsilon, \varepsilon)$. That is,

$$f_x(s) \geq \frac{f_x(\varepsilon) - f_x(-\varepsilon)}{2\varepsilon} s + \frac{f_x(\varepsilon) + f_x(\varepsilon)}{2},$$

for all $s \in (-\varepsilon, \varepsilon)$. Thus,

$$\begin{aligned}
\int_{-\varepsilon}^{\varepsilon} f_x(s)\,\mathrm{d}s &\geq \int_{-\varepsilon}^{\varepsilon} \frac{f_x(\varepsilon) - f_x(-\varepsilon)}{2\varepsilon} s + \frac{f_x(\varepsilon) + f_x(-\varepsilon)}{2}\,\mathrm{d}s \\
&= \int_{-\varepsilon}^{\varepsilon} \frac{f_x(\varepsilon) + f_x(-\varepsilon)}{2}\,\mathrm{d}s \\
&= \int_{-\varepsilon}^{\varepsilon} \exp\left(-c(x^2+\varepsilon^2)\right) \cdot \frac{\exp\left(-2cx\varepsilon\right) + \exp\left(2cx\varepsilon\right)}{2}\,\mathrm{d}s \\
&\geq \int_{-\varepsilon}^{\varepsilon} \exp\left(-c(x^2+\varepsilon^2)\right)\,\mathrm{d}s.
\end{aligned}$$

Case $x \geq 1$. The integral on the right-hand side of equation (A.20) has the same value for any affine integrand r_x for which $r_x(0) = \exp\left(-c(x^2+\varepsilon^2)\right)$. Thus, proving that $f_x(s) \geq r_x(s)$, for all $s \in (-\varepsilon, \varepsilon)$, concludes the proof.

In particular, we can choose

$$r_x(s) = f_x'(0) \cdot s + \exp\left(-c(x^2+\varepsilon^2)\right).$$

Since

$$f'_x(s) = -2c(x+s)\exp\left(-c(x+s)^2\right),$$

we aim to show that

$$\exp\left(-c(x+s)^2\right) \geq -2csx\exp\left(-cx^2\right) + \exp\left(-c(x^2+\varepsilon^2)\right)$$

for $s \in (-\varepsilon, \varepsilon)$. Dividing by $\exp(-c(x^2+s^2))$ and rearranging, we obtain

$$\exp(-2csx) + 2csx\exp\left(cs^2\right) - \exp\left(-c\left(\varepsilon^2 - s^2\right)\right) \geq 0. \tag{A.21}$$

Now, if $s \geq 0$, we have that

$$\begin{aligned}\exp(-2csx) + 2csx\exp(cs^2) - \exp(-c(\varepsilon^2 - s^2)) &\geq 1 - 2csx + 2csx(1+cs^2) - \exp(-c\varepsilon^2) \qquad \text{(A.22)}\\ &= 1 + 2c^2s^3x - \exp(-c\varepsilon^2)\\ &\geq 2c^2s^3x\\ &\geq 0,\end{aligned}$$

where in equation (A.22) we used that $e^y \geq 1+y$.

Now consider the sub-case $s < 0$. By Taylor's theorem,

$$\exp(y) = 1 + y + \frac{y^2}{2} + \frac{\exp(\xi_1)\cdot y^3}{6}$$

and

$$\exp(y) = 1 + y + \frac{\exp(\xi_2)\cdot y^2}{2},$$

for some $\xi_1, \xi_2 \in [0, y]$. Letting $\ell = -s \in (0,1)$, we have

$$\exp(2c\ell x) \geq 1 + 2c\ell x + \frac{(2c\ell x)^2}{2} + \frac{(2c\ell x)^3}{6}$$

and

$$\begin{aligned}\exp(c\ell^2) &\leq 1 + c\ell^2 + \frac{\exp(c\ell^2)(c\ell^2)^2}{2}\\ &\leq 1 + c\ell^2 + \frac{\left(1+3(c\ell^2)\right)(c\ell^2)^2}{2}.\end{aligned}$$

since $e^y \leq 1+3y$ for $0 \leq y \leq 1$. Finally, applying this to equation (A.21), we have

$$\begin{aligned}&\exp(-2csx) + 2csx\exp\left(cs^2\right) - \exp\left(-c\left(\varepsilon^2 - s^2\right)\right)\\ &\quad\geq \exp(2c\ell x) - 2c\ell x\exp\left(c\ell^2\right) - 1\\ &\quad\geq 1 + 2c\ell x + \frac{(2c\ell x)^2}{2} + \frac{(2c\ell x)^3}{6} - 2c\ell x\left(1 + c\ell^2 + \frac{c^2\ell^4(1+3c\ell^2)}{2}\right) - 1\\ &\quad= 2c^2\ell^2x^2 + \frac{4}{3}c^3\ell^3x^3 - 2c\ell x\left(c\ell^2 + \frac{c^2\ell^4(1+3c\ell^2)}{2}\right)\\ &\quad= 2c^2\ell^2x(x-\ell) + c^3\ell^3x\left(\frac{4}{3}x^2 - \ell^2(1+3c\ell^2)\right).\end{aligned}$$

The latter is non negative for $x \geq 1$ and $c \leq \frac{1}{9}$, since $\ell = -s \leq \varepsilon < 1$, so that $\frac{4}{3}x^2 - \ell^2(1+3c\ell^2) \geq \frac{4}{3} - 1 - \frac{1}{3} = 0$. □

B Proofs omitted

B.1 Proof of Lemma 3.5.1

By the distribution of $\mathbf{X}$,

$$\Pr[\mathbf{X} \in \mathfrak{B}_\infty(\boldsymbol{z}, \varepsilon)] = \int_{\mathfrak{B}_\infty(\boldsymbol{z},\varepsilon)} \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} \cdot \exp\left(-\frac{\|\boldsymbol{x}\|_2^2}{2\sigma^2}\right) \mathrm{d}\boldsymbol{x}.$$

Since $\mathfrak{B}_\infty(\boldsymbol{z}, \varepsilon) \subseteq \mathfrak{B}_\infty(\mathbf{0}, 2)$ and for all $\boldsymbol{x} \in \mathbb{R}^d$ it holds that $\|\boldsymbol{x}\|_2 \leq \sqrt{d} \cdot \|\boldsymbol{x}\|_\infty$, and, thus,

$$\exp\left(-\frac{2d}{\sigma^2}\right) \leq \exp\left(-\frac{\|\boldsymbol{x}\|_2^2}{2\sigma^2}\right) \leq 1.$$

The thesis follows by noting that the hypercube $\mathfrak{B}_\infty(\boldsymbol{z}, \varepsilon)$ has measure $(2\varepsilon)^d$.

B.2 Proof of Lemma 3.7.1

Inheriting the setup from the proof of Lemma 3.5.3 and proceeding analogously we obtain that $\sigma_{\mathrm{A}}^2 = \alpha n \left(1 - \frac{\alpha}{2}\right)$ and $\sigma_{\mathrm{B}}^2 = \frac{\alpha^2 n}{2}$. We diverge from that argument after equation (3.3). Preserving equality for a bit longer, we have that

$$\begin{aligned}(\Pr[\mathrm{Y}_S = 1, \mathrm{Y}_T = 1])^{\frac{1}{d}} &= \int_{\mathbb{R}} \varphi_{\mathrm{B}}(x) \cdot (\Pr[\mathrm{A} \in (z - x - \varepsilon, z - x + \varepsilon)])^2 \, \mathrm{d}x \\ &= \int_{\mathbb{R}} \varphi_{\mathrm{B}}(x) \cdot \left(\int_{z-x-\varepsilon}^{z-x+\varepsilon} \varphi_{\mathrm{A}}(y) \, \mathrm{d}y\right)^2 \mathrm{d}x.\end{aligned}$$

The hypothesis on n implies that $2\sigma_a^2 \geq 10$, so, by Claim A.7,

$$\begin{aligned}\left(\int_{z-x-\varepsilon}^{z-x+\varepsilon} \varphi_{\mathrm{A}}(y) \, \mathrm{d}y\right)^2 &\geq (2\varepsilon)^2 \cdot \varphi_{\mathrm{A}}(z - x - \varepsilon) \cdot \varphi_{\mathrm{A}}(z - x + \varepsilon) \\ &= \frac{(2\varepsilon)^2}{2\pi\sigma_{\mathrm{A}}^2} \cdot \exp\left(-\frac{(z - x - \varepsilon)^2}{2\sigma_{\mathrm{A}}^2}\right) \cdot \exp\left(-\frac{(z - x + \varepsilon)^2}{2\sigma_{\mathrm{A}}^2}\right) \\ &= e^{-\varepsilon^2/\sigma_{\mathrm{A}}^2} \cdot \frac{1}{\sqrt{2}} \cdot \frac{(2\varepsilon)^2}{\sqrt{2\pi\sigma_{\mathrm{A}}^2}} \cdot \frac{1}{\sqrt{\pi\sigma_{\mathrm{A}}^2}} \cdot \exp\left(-\frac{(z - x)^2}{\sigma_{\mathrm{A}}^2}\right) \\ &= e^{-\varepsilon^2/\sigma_{\mathrm{A}}^2} \cdot \frac{1}{\sqrt{2}} \cdot \frac{(2\varepsilon)^2}{\sqrt{2\pi\sigma_{\mathrm{A}}^2}} \cdot \varphi_{\mathrm{A}/\sqrt{2}}(z - x).\end{aligned}$$

Then, as before, we can reduce the main integral to a convolution. Namely, it holds that

$$\begin{aligned}\int_{\mathbb{R}} \varphi_{\mathrm{B}}(x) \cdot \varphi_{\mathrm{A}/\sqrt{2}}(z - x) \, \mathrm{d}x &= \varphi_{\mathrm{B+A}/\sqrt{2}}(z) \\ &= \frac{1}{\sqrt{2\pi\sigma_{\mathrm{B+A}/\sqrt{2}}^2}} \cdot \exp\left(-\frac{z^2}{2\sigma_{\mathrm{B+A}/\sqrt{2}}^2}\right).\end{aligned}$$

Altogether, we have that

$$(\Pr[\mathrm{Y}_S = 1, \mathrm{Y}_T = 1])^{\frac{1}{d}} \geq \frac{(2\varepsilon)^2}{2\pi} \cdot \frac{1}{\sqrt{2\sigma_{\mathrm{A}}^2 \sigma^2_{\mathrm{B+A}/\sqrt{2}}}} \cdot \exp\left(-\frac{\varepsilon^2}{\sigma_{\mathrm{A}}^2} - \frac{z^2}{2\sigma^2_{\mathrm{B+A}/\sqrt{2}}}\right)$$
$$= \frac{(2\varepsilon)^2}{2\pi\alpha n} \cdot \frac{1}{\sqrt{1-\frac{\alpha^2}{4}}} \cdot \exp\left(-\frac{1}{\alpha n} \cdot \left(\frac{2\varepsilon^2}{2-\alpha} + \frac{2z^2}{2+\alpha}\right)\right).$$

where the last equality follows from recalling that $\sigma_{\mathrm{B}}^2 = \frac{\alpha^2 n}{2}$ and $\sigma_{\mathrm{A}}^2 = \alpha n\left(1-\frac{\alpha}{2}\right)$, which implies that $\sigma^2_{\mathrm{B+A}/\sqrt{2}} = \frac{\alpha^2 n}{2} + \frac{\alpha n}{2}\left(1-\frac{\alpha}{2}\right)$. Finally, the hypotheses $z \in [-1,1]$, $\varepsilon \in (0,1)$, and $\alpha \in \left(0, \frac{1}{2}\right)$ imply that $\frac{2\varepsilon^2}{2-\alpha} + \frac{2z^2}{2+\alpha} < 3$.

B.3 Proof of Theorem 3.5.5

Let $\delta = 1396 \cdot \log\frac{9}{5}$. Observe that, if $\alpha = \frac{1}{6\sqrt{d}}$, then

$$\delta N \geq \max\left\{\frac{18d\log\frac{d}{\alpha}}{\alpha^2}, \frac{8d}{\alpha\log\frac{1}{\alpha}}\left(\log d + \log\frac{1}{\varepsilon} + 1\right) + \frac{8}{\alpha}\right\}.$$

Hence, δN input variables suffice to apply Lemma 3.5.4. Let $n = k \cdot \delta N$ with $k \in \mathbb{N}$. By Lemma 3.5.4, for any $\boldsymbol{z} \in [-1,1]^d$, the probability than no subset-sum is sufficiently close to $\boldsymbol{z}$ is at most $\left(\frac{5}{9}\right)^k$. Leveraging the fact that it is possible to cover $[-1,1]^d$ by $\frac{1}{\varepsilon^d}$ hypercubes of radius ε, we can ensure that the probability of failing to 2ε-approximate any $\boldsymbol{z}$ is, by the union bound, at most

$$\frac{1}{\varepsilon^d} \cdot \left(\frac{5}{9}\right)^k = 2^{-k\log\frac{9}{5} + d\log\frac{1}{\varepsilon}}$$
$$= \exp\left[-\ln 2 \frac{n - \frac{d\delta N}{\log\frac{9}{5}}\log\frac{1}{\varepsilon}}{\frac{\delta N}{\log\frac{9}{5}}}\right].$$

Thus, we can conclude the result for $C = \frac{\delta}{\log\frac{9}{5}} = 1396$.

C Generalisation of our result

If the target value $\boldsymbol{z}$ lies in the hypercube $[-\lambda\sqrt{n}, \lambda\sqrt{n}]^d$, for some $\lambda > \frac{1}{\sqrt{n}}$, we have slightly different bounds for the expectation and for the variance of Y. In particular, Corollary 3.5.2 would give

$$e^{-\frac{2\lambda^2 d}{\alpha}} \frac{(2\varepsilon)^d |\mathcal{C}|}{(2\pi\alpha n)^{\frac{d}{2}}} \leq \mathbb{E}[\mathrm{Y}] \leq \frac{(2\varepsilon)^d |\mathcal{C}|}{(2\pi\alpha n)^{\frac{d}{2}}}. \tag{C.23}$$

On the other hand, as the proof of Lemma 3.5.3 never uses that $\boldsymbol{z} \in [-1,1]^d$ but only exploits the bound on the expectation, it would yield

$$\mathrm{Var}(\mathrm{Y}) \leq \frac{(2\varepsilon)^{2d}|\mathcal{C}|^2}{(2\pi\alpha n)^d}\left[(1-4\alpha^2)^{-\frac{d}{2} - e^{-\frac{4\lambda^2 d}{\alpha}}}\right] + \frac{(2\varepsilon)^d|\mathcal{C}|}{(2\pi\alpha n)^{\frac{d}{2}}}. \tag{C.24}$$

We focus on the case $\lambda = \frac{1}{2}\sqrt{\frac{\alpha}{17d}}$ when $n > \frac{68d}{\alpha}$ (which implies $\lambda\sqrt{n} > 1$). Thus, we have a new estimation for the probability of hitting a single value.

Lemma C.1. *Given $d, n \in \mathbb{N}$, $\varepsilon \in (0,1)$, and $\alpha \in (0, \frac{1}{6}]$, let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ i.i.d. following $\mathcal{N}\,\mathbf{0}, \boldsymbol{I}_d$, $\boldsymbol{z} \in [-\lambda\sqrt{n}, \lambda\sqrt{n}]^d$, with $\lambda = \frac{1}{2}\sqrt{\frac{\alpha}{17d}}$, and $\boldsymbol{\mathcal{C}} \subseteq \binom{[n]}{\alpha n}$. If any two subsets in $\boldsymbol{\mathcal{C}}$ intersect in at most $2\alpha^2 n$ elements, $\alpha \leq \frac{1}{6\sqrt{d}}$, and*

$$n \geq \frac{144d}{\alpha^2}\left(\log\frac{1}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right),$$

then

$$\Pr[\mathrm{Y} \geq 1] \geq \frac{1}{3}.$$

Proof. By Chebyshev's inequality, it holds that

$$\begin{aligned}\Pr[\mathrm{Y} \geq 1] &\geq \Pr[|\mathrm{Y} - \mathbb{E}[\mathrm{Y}]| < \frac{\mathbb{E}[\mathrm{Y}]}{2}] \\ &\geq 1 - \frac{4 \cdot \mathrm{Var}(\mathrm{Y})}{\mathbb{E}[\mathrm{Y}]^2}.\end{aligned}$$

Notice that $\frac{4\lambda^2 d}{\alpha} = \frac{1}{17}$. Hence, using equations (C.23) and (C.24), we get that

$$\begin{aligned}\frac{4 \cdot \mathrm{Var}(\mathrm{Y})}{\mathbb{E}[\mathrm{Y}]^2} &\leq 4 \cdot \frac{e^{\frac{1}{17}} \cdot (2\pi\alpha n)^d}{(2\varepsilon)^{2d}|\boldsymbol{\mathcal{C}}|^2} \cdot \left[\frac{(2\varepsilon)^{2d}|\boldsymbol{\mathcal{C}}|^2}{(2\pi\alpha n)^d} \cdot \left[(1-4\alpha^2)^{-\frac{d}{2}} - e^{-\frac{1}{17}}\right] + \frac{(2\varepsilon)^d|\boldsymbol{\mathcal{C}}|}{(2\pi\alpha n)^{\frac{d}{2}}}\right] \\ &= 4 \cdot \left(\frac{e^{\frac{1}{17}}}{(1-4\alpha^2)^{\frac{d}{2}}} - 1\right) + \frac{4e^{\frac{1}{17}} \cdot (2\pi\alpha n)^{\frac{d}{2}}}{(2\varepsilon)^d|\boldsymbol{\mathcal{C}}|}.\end{aligned}$$

Note that Claim A.4 holds exactly as it is for the ratio

$$\frac{e^{\frac{1}{17}}}{(1-4\alpha^2)^{\frac{d}{2}}}$$

obtaining the same bound for $n \geq \frac{68d}{\alpha}$ and $\alpha \leq \frac{1}{6\sqrt{d}}$, which yields

$$4 \cdot \left(\frac{e^{\frac{1}{17}}}{(1-4\alpha^2)^{\frac{d}{2}}} - 1\right) \leq \frac{1}{2}.$$

Furthermore, also Claim A.3 is true replacing $e^{\frac{4d}{\alpha n}}$ by $e^{\frac{1}{17}}$. Thus, as $n \geq \frac{144d}{\alpha^2}\left(\log\frac{1}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)$ and $\alpha \leq \frac{1}{6}$, Claim A.3 implies that

$$\frac{4e^{\frac{1}{17}} \cdot (2\pi\alpha n)^{\frac{d}{2}}}{(2\varepsilon)^d|\boldsymbol{\mathcal{C}}|} \leq \varepsilon.$$

□

We remark that we cannot let λ be asymptotically greater than $\sqrt{\frac{\alpha}{d}}$ otherwise our method fails. Indeed, by Remark 3.7.1, the term $\frac{4\,\mathrm{Var}(Y)}{\mathbb{E}[Y]^2}$ is at least

$$4 \cdot \left(\frac{e^{\frac{4\lambda^2 d}{\alpha} - \frac{3\lambda^2 d}{\alpha}}}{\left(1 - \frac{\alpha^2}{4}\right)^{\frac{d}{2}}} - 1 \right).$$

The latter is greater than or equal to 1 if $\lambda \geq \sqrt{\frac{\alpha}{d}}$ since $e^{\frac{\lambda^2 d}{\alpha}} \geq 1 + \frac{\lambda^2 d}{\alpha}$.

We are ready to state our first generalised version of Theorem 3.5.5.

Theorem C.2. *For given d and $\varepsilon \in (0,1)$, let $\mathbf{X}_1, \dots, \mathbf{X}_n$ be n independent standard normal d-dimensional random vectors and let $\alpha \in (0, \frac{1}{6\sqrt{d}}]$. There exist two universal constants $C > \delta > 0$ such that, if*

$$n \geq C \frac{d^2}{\alpha^2} \left(\log \frac{1}{\varepsilon} + \log d + \log \frac{1}{\alpha} \right)^2,$$

the following holds with probability at least

$$1 - \exp\left[-\ln 2 \cdot \left(\frac{n}{\delta \frac{d}{\alpha^2} \left(\log \frac{1}{\varepsilon} + \log d + \log \frac{1}{\alpha} \right)} - d \log \frac{1}{\varepsilon} \right) \right] :$$

for all $\boldsymbol{z} \in [-\lambda\sqrt{n}, \lambda\sqrt{n}]^d$, with $\lambda = \frac{1}{2}\sqrt{\frac{\alpha}{17d}}$, there exists a subset $S_{\boldsymbol{z}} \subseteq [n]$, such that

$$\left\| \boldsymbol{z} - \sum_{i \in S_{\boldsymbol{z}}} \mathbf{X}_i \right\|_\infty \leq 2\varepsilon.$$

Moreover, the property above remains true even if we restrict to subsets of size αn.

Proof. Let $\frac{n}{\frac{144d}{\alpha^2}\left(\log \frac{1}{\varepsilon} + \log d + \log \frac{1}{\alpha}\right)} = k \geq 1$ with $k \in \mathbb{N}$. By Lemma C.1, for any $\boldsymbol{z} \in [-\lambda\sqrt{n}, \lambda\sqrt{n}]^d$, the probability than no subset-sum is sufficiently close to $\boldsymbol{z}$ is at most $\left(\frac{2}{3}\right)^k$. Leveraging the fact that it is possible to cover $[-\lambda\sqrt{n}, \lambda\sqrt{n}]^d$ by $\left(\frac{\lambda\sqrt{n}}{\varepsilon}\right)^d$ hypercubes of radius ε, we can ensure that the probability of failing to 2ε-approximate any $\boldsymbol{z}$ is, by the union bound, at most

$$\begin{aligned}
\left(\frac{\lambda\sqrt{n}}{\varepsilon}\right)^d \cdot \left(\frac{2}{3}\right)^k &= 2^{-k \log \frac{3}{2} + d\left(\log \frac{1}{\varepsilon} + \frac{1}{2}\log n + \log \lambda\right)} \\
&= \exp\left[-\ln 2 \cdot \frac{n - \frac{144d^2}{\alpha^2 \log \frac{3}{2}} \left(\log \frac{1}{\varepsilon} + \frac{1}{2}\log n + \log \lambda\right)\left(\log \frac{1}{\varepsilon} + \log d + \log \frac{1}{\alpha}\right)}{\frac{144d}{\alpha^2 \log \frac{3}{2}} \left(\log \frac{1}{\varepsilon} + \log d + \log \frac{1}{\alpha}\right)} \right] \\
&\leq \exp\left[-\ln 2 \cdot \frac{n - \frac{144d^2}{\alpha^2 \log \frac{3}{2}} \left(\log \frac{1}{\varepsilon} + \frac{1}{2}\log n\right)\left(\log \frac{1}{\varepsilon} + \log d + \log \frac{1}{\alpha}\right)}{\frac{144d}{\alpha^2 \log \frac{3}{2}} \left(\log \frac{1}{\varepsilon} + \log d + \log \frac{1}{\alpha}\right)} \right]
\end{aligned} \tag{C.25}$$

since $\lambda < 1$. Consider $\frac{n}{2} - \frac{144d^2}{2\alpha^2 \log\frac{3}{2}} \log n \left(\log\frac{1}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)$. Let $k = k'\left(\log\frac{1}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)$, which means that $n = \frac{144k'd}{\alpha^2}\left(\log\frac{1}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)^2$. Then

$$\begin{aligned}
&\frac{n}{2} - \frac{144d^2}{2\alpha^2 \log\frac{3}{2}} \log n \left(\log\frac{1}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right) \\
&\quad = \frac{144d}{2\alpha^2}\left(\log\frac{1}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)\left[k'\left(\log\frac{1}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)\right. \\
&\qquad \left. - \frac{d}{\log\frac{3}{2}}\left(\log\frac{144}{\log\frac{3}{2}} + \log k' + \log d + 2\log\frac{1}{\alpha} + 2\log\left(\log\frac{1}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)\right)\right] \\
&\quad \geq \frac{144d}{2\alpha^2}\left(\log\frac{1}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)\left[k'\left(\log\frac{1}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)\right. \\
&\qquad \left. - 2d\left(8 + \log k' + \log d + 2\log\frac{1}{\alpha} + 2\log\left(\log\frac{1}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)\right)\right]
\end{aligned}$$

If $k' = 17d$, we have that

$$\begin{aligned}
&k'\left(\log\frac{1}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right) - 2d\left(8 + \log k' + \log d + 2\log\frac{1}{\alpha} + 2\log\left(\log\frac{1}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)\right) \\
&\quad \geq 4d\left(\log\frac{1}{\varepsilon} + \log d + \log\frac{1}{\alpha} - \log\left(\log\frac{1}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)\right) + 13d\log d + 13d\log\frac{1}{\alpha} \\
&\qquad - 16d - 2d\log c - 3d\log d - 4d\log\frac{1}{\alpha} \\
&\quad = 10d\log d + 9d\log\frac{1}{\alpha} - 16d - 2d\log 17 \geq 0,
\end{aligned}$$

as $\alpha \leq \frac{1}{6}$. Thus, for $n \geq \frac{17\cdot 144d^2}{\alpha^2}\left(\log\frac{1}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)^2$, we have that the expression in equation (C.25) is at most

$$\exp\left[-\ln 2 \cdot \frac{n - \frac{288d^2}{\alpha^2 \log\frac{3}{2}} \log\frac{1}{\varepsilon}\left(\log\frac{1}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)}{\frac{288d}{\alpha^2 \log\frac{3}{2}}\left(\log\frac{1}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)}\right].$$

We have the thesis by setting $\delta = \frac{288}{\log\frac{3}{2}}$ and $C = 17 \cdot 144$. □

Our analysis, which relies on fixed subset sizes, easily extends Theorem C.2 for non-centred and non-unitary normal random vectors.

Corollary C.3. *Let $\sigma > 0$ and $\varepsilon \in (0, \sigma)$. Given $d, n \in \mathbb{N}$ let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be independent normal d-dimensional random vectors with $\mathbf{X}_i \sim \mathcal{N}(\boldsymbol{v}, \sigma^2 \cdot \boldsymbol{I}_d)$, for any vector $\boldsymbol{v} \in \mathbb{R}^d$. Furthermore, let $\alpha \in (0, \frac{1}{6\sqrt{d}})$. There exist two universal constants $C > \delta > 0$ such that, if*

$$n \geq C\frac{d^2}{\alpha^2}\left(\log\frac{\sigma}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)^2,$$

then, with probability

$$1 - \exp\left[-\ln 2 \cdot \left(\frac{n}{\delta\frac{d}{\alpha^2}\left(\log\frac{\sigma}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)} - d\log\frac{\sigma}{\varepsilon}\right)\right],$$

for all $\boldsymbol{z} \in [-\sigma\lambda\sqrt{n}, \sigma\lambda\sqrt{n}]^d + \alpha n\boldsymbol{v}$*, with* $\lambda = \frac{1}{2}\sqrt{\frac{\alpha}{17d}}$*, there exists a subset* $S_{\boldsymbol{z}} \subseteq [n]$ *for which*

$$\left\|\boldsymbol{z} - \sum_{i \in S_{\boldsymbol{z}}} \mathbf{X}_i\right\|_\infty \leq 2\varepsilon.$$

Moreover, this remains true even when restricted to subsets of size αn*.*

Proof. Simply apply Theorem C.2 to the random vectors $\frac{\mathbf{X}_i - \boldsymbol{v}}{\sigma}$ with error $\frac{\varepsilon}{\sigma}$. □

Following the line of Lueker (1998), we also observe that our results extend to a wider class of probability distributions.

Definition C.1. Consider any two random variables X and Y having the same codomain, and let $\varphi_{\mathrm{X}}(x), \varphi_{\mathrm{Y}}(x)$ be their probability density functions. We say that X *contains* Y with probability p if a constant $p \in (0, 1]$ exists such that $\varphi_{\mathrm{X}}(x) = p \cdot \varphi_{\mathrm{Y}}(x) + (1 - p)f(x)$ for any function $f(x)$.

If X contains Y with probability p, we can describe the behaviour of X as follows: with probability p, draw Y; with probability $1 - p$, draw something else. An adapted version of our result holds for random variables containing Gaussian distributions.

Corollary C.4. *Let* $\sigma > 0$*,* $\varepsilon \in (0, \sigma)$*, and let* $p \in (0, 1]$ *be a constant. Given* $d, n \in \mathbb{N}$ *let* $\mathbf{Y}_1, \ldots, \mathbf{Y}_n$ *be independent* d*-dimensional random vectors containing* d*-dimensional normal random vectors* $\mathbf{X} \sim \mathcal{N}(\boldsymbol{v}, \sigma^2 \cdot \boldsymbol{I}_d)$ *with probability* p*, where* $\boldsymbol{v}$ *is any vector in* $\mathbb{R}^d$*. Furthermore, let* $\alpha \in (0, \frac{1}{6\sqrt{d}})$*. There exist two universal constants* $C > \delta > 0$ *such that, if*

$$n \geq 2C\frac{d^2}{p\alpha^2}\left(\log\frac{\sigma}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)^2,$$

then, with probability

$$1 - 2\exp\left[-\ln 2 \cdot \left(\frac{pn}{2\delta\frac{d}{\alpha^2}\left(\log\frac{\sigma}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)} - d\log\frac{\sigma}{\varepsilon}\right)\right],$$

for all $\boldsymbol{z} \in [-\sigma\lambda\sqrt{\frac{pn}{2}}, \sigma\lambda\sqrt{\frac{pn}{2}}]^d + \frac{\alpha pn}{2}\boldsymbol{v}$*, with* $\lambda = \frac{1}{2}\sqrt{\frac{\alpha}{17d}}$*, there exists a subset* $S_{\boldsymbol{z}} \subseteq [n]$ *for which*

$$\left\|\boldsymbol{z} - \sum_{i \in S_{\boldsymbol{z}}} \mathbf{X}_i\right\|_\infty \leq 2\varepsilon.$$

Moreover, this remains true even when restricted to subsets of size $\frac{\alpha pn}{2}$*.*

Proof. With a simple application of the Chernoff bound, we have that at least $\frac{pn}{2}$ random vectors are normal random vectors with probability $1 - e^{-\frac{pn}{8}}$. Conditional on this event, we can apply Corollary C.3 to the $\frac{pn}{2}$ normal random vectors. Since $\Pr[A, B] \geq \Pr[A \mid B]\Pr[B]$ for any two events A, B, and $2\delta\frac{d}{\alpha^2}\left(\log\frac{\sigma}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right) \geq 8$, the thesis holds with probability at least

$$1 - \exp\left[-\ln 2 \cdot \left(\frac{pn}{2\delta\frac{d}{\alpha^2}\left(\log\frac{\sigma}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)} - d\log\frac{\sigma}{\varepsilon}\right)\right] - \exp\left[-\frac{pn}{8}\right]$$

$$\geq 1 - 2\exp\left[-\ln 2 \cdot \left(\frac{pn}{2\delta\frac{d}{\alpha^2}\left(\log\frac{\sigma}{\varepsilon} + \log d + \log\frac{1}{\alpha}\right)} - d\log\frac{\sigma}{\varepsilon}\right)\right].$$

□

D Discrete setting

We believe that it should not be hard to adapt our proof to several discrete distributions, in order to obtain results similar to those discussed in the Related Work section. We also note that our Theorem 3.1.2 already implies an analogous discrete result. Suppose that we quantise our random vectors by truncating them to the $\lfloor \log \frac{1}{\delta} \rfloor$-th binary place, obtaining vectors $\hat{\mathbf{X}}_i$ such that $\|\hat{\mathbf{X}}_i - \mathbf{X}_i\|_\infty < \delta$. For any $\boldsymbol{z} \in [-1,1]^d$, Theorem 3.1.2 guarantees that w.h.p. there is a subset of indices $I \subseteq [n]$ such that $\|\boldsymbol{z} - \sum_{i\in I} \mathbf{X}_i\|_\infty < \varepsilon$ and, hence, by the triangular inequality, $\|\boldsymbol{z} - \sum_{i\in I} \hat{\mathbf{X}}_i\|_\infty < n\delta + 2\varepsilon$. As a special case ($\delta = 2\varepsilon$), we have the following:

Corollary D.1 (Discretization of Theorem 3.1.2)**.** *Given $d \in \mathbb{N}$, $\varepsilon \in (0,1)$, let $\hat{\mathbf{X}}_1, \dots, \hat{\mathbf{X}}_n$ be independent standard normal d-dimensional vectors truncated to the $\lfloor \log \frac{1}{\varepsilon} \rfloor$-th binary place. There exists a universal constant $C > 0$ such that, if $n \geq Cd^3 \log \frac{1}{\varepsilon} \left(\log \frac{1}{\varepsilon} + \log d \right)$, then, with high probability, for all vectors $\hat{\boldsymbol{z}}$ with entries in $\{k\varepsilon\}_{\lceil -\frac{1}{\varepsilon} \rceil \leq k \leq \lfloor \frac{1}{\varepsilon} \rfloor}$ there exists a subset $S_{\boldsymbol{z}} \subseteq [n]$ for which*

$$\left\| \hat{\boldsymbol{z}} - \sum_{i \in S_{\boldsymbol{z}}} \hat{\mathbf{X}}_i \right\|_\infty \leq 2\varepsilon(n+1).$$

Moreover, the approximation can be achieved with subsets of size $\frac{n}{6\sqrt{d}}$.

E Connection with non-deterministic random walks

Consider a discrete-time stochastic process whose state space is $\mathbb{R}^d$ which starts at the origin. In the first step, the process "branches" in two processes, one of which keeps still, while the other moves by the vector $\mathbf{X}_1$. Recursively, given any time i and any process, at the next time step the process branches in two other processes, one of which keeps still, while the other moves by the vector $\mathbf{X}_{i+1}$. In this setting, when $\mathbf{X}_{i+1}$ are sampled from a standard multivariate normal distribution, our results imply that the resulting process is space-filling: the process eventually gets arbitrarily close to each point in $\mathbb{R}^d$. This should be contrasted with the fact that a Brownian motion is transient in dimension $d \geq 3$ (Mörters & Peres, 2010). The above process can also be interpreted as a multi-dimensional version of nondeterministic walks as introduced in Panafieu, Lamali, and Wallner (2019) in the context of the analysis of encapsulations and decapsulations of network protocols, where the i-th N-step is $\{\mathbf{X}_i, \vec{0}\}$.

Appendices of Chapter 4

F Bound on the Norm of a Convolution

The following proposition can be seen as a special case of Young's Convolution Inequality for functions over $\mathcal{Z}^n$ where the norms in the inequality are the ℓ_1 and ℓ_∞ norms.

Proposition F.1 (Tensor Convolution Inequality)**.** *Given* $\boldsymbol{K} \in \mathbb{R}^{d\times d\times c_0\times c_1}$ *and* $\boldsymbol{X} \in \mathbb{R}^{D\times D\times c_0}$*, we have*

$$\|\boldsymbol{K} * \boldsymbol{X}\|_{\max} \leq \|\boldsymbol{K}\|_1 \cdot \|\boldsymbol{X}\|_{\max}.$$

Proof. We have

$$\begin{aligned}
\|\boldsymbol{K} * \boldsymbol{X}\|_{\max} &\leq \max_{i,j\in[D],\ell\in[c_1]} \sum_{i',j'\in[d],k\in[c]} \left|K_{i',j',k,\ell} X_{i-i'+1,j-j'+1,k}\right| \\
&\leq \max_{i,j,\ell} \sum_{i',j',k} \left|K_{i',j',k,\ell}\right| \|\boldsymbol{X}\|_{\max} \\
&\leq \max_{i,j,\ell} \left(\sum_{i',j',k} \left|K_{i',j',k,\ell}\right|\right) \|\boldsymbol{X}\|_{\max} \\
&\leq \max_{i,j,\ell} \|\boldsymbol{K}\|_1 \|\boldsymbol{X}\|_{\max} = \|\boldsymbol{K}\|_1 \cdot \|\boldsymbol{X}\|_{\max}.
\end{aligned}$$

□

G CNN Approximation (Proof of Theorem 4.2.3)

Since Lemma 4.2.2 provide bounds in terms of the output of a layer, the study of the propagation of this error through the network is mostly independent of the layer type. Hence, the next proof follows the structure of (Pensia et al., 2020, Theorem 1), where our Lemma 4.2.2 assumes the role of their Lemma 3, and where we leverage our proposition F.1 in order to address the problem of bounding the maximum norm of a convolution.

Proof (of Theorem 4.2.3). For the sake of brevity, in the proof we denote the max-norm simply by $\|\cdot\|$. Let $\boldsymbol{X}^i$ be the input of the i-th layer of the network f. Thus,

1. $\boldsymbol{X}^1 = \boldsymbol{X}$,
2. $\boldsymbol{X}^{i+1} = \sigma(\boldsymbol{K}^i * \boldsymbol{X}^i)$ for $1 \leq i \leq \ell - 1$ and
3. $f(\boldsymbol{X}) = \boldsymbol{K}^\ell * \boldsymbol{X}^\ell$.

By applying Lemma 4.2.2 to each layer, we choose masks $\boldsymbol{S}^{2i}$ and $\boldsymbol{S}^{2i-1}$ so that

$$\sup_{\boldsymbol{X}} \left\| \boldsymbol{K}^i * \boldsymbol{X} - (\boldsymbol{L}^{2i} \odot \boldsymbol{S}^{2i}) * \sigma[(\boldsymbol{L}^{2i-1} \odot \boldsymbol{S}^{2i-1}) * \boldsymbol{X}] \right\| < \frac{\varepsilon}{2\ell} \tag{G.26}$$

with probability at least $1 - \frac{\varepsilon}{2\ell}$.

Since the ReLU function is 1-Lipschitz with respect to the max norm, the above event implies the following for all $i \in [\ell - 1]$:

$$\sup_{\boldsymbol{X}} \left\| \sigma(\boldsymbol{K}^i * \boldsymbol{X}) - \sigma[(\boldsymbol{L}^{2i} \odot \boldsymbol{S}^{2i}) * \sigma[(\boldsymbol{L}^{2i-1} \odot \boldsymbol{S}^{2i-1}) * \boldsymbol{X}]] \right\| < \frac{\varepsilon}{2\ell}. \tag{G.27}$$

By a union bound, with probability $1 - \varepsilon$, equations (G.26) and (G.27) hold for all layers simultaneously. In the rest of the proof, we implicitly condition on the latter event.

For any fixed function f, let g be the pruned network constructed layer-wise, by pruning with binary masks which satisfy equations (G.26) and (G.27). Let these pruned tensors be $\tilde{\boldsymbol{L}}^i$, and let $\tilde{\boldsymbol{X}}^i$ be the input to the $(2i-1)$-th layer of g.

We note that $\tilde{\boldsymbol{X}}^i$ satisfies the recurrence relation

1. $\tilde{\boldsymbol{X}}^1 = \boldsymbol{X}$,
2. $\tilde{\boldsymbol{X}}^{i+1} = \sigma(\tilde{\boldsymbol{L}}^{2i} * \sigma(\tilde{\boldsymbol{L}}^{2i-1} * \boldsymbol{X}^i))$ for $1 \leq i \leq \ell - 1$.

Because $\|\boldsymbol{X}\| \leq 1$, equation (G.27) implies that $\|\tilde{\boldsymbol{X}}^i\| \leq \left(1 + \frac{\varepsilon}{2\ell}\right)^{i-1}$. To see this, note that equation (G.27) implies, for $1 \leq i \leq \ell - 1$,

$$\left\| \frac{\sigma(\boldsymbol{K}^i * \tilde{\boldsymbol{X}}^i)}{\|\tilde{\boldsymbol{X}}^i\|} - \frac{\tilde{\boldsymbol{X}}^{i+1}}{\|\tilde{\boldsymbol{X}}^i\|} \right\| \leq \frac{\varepsilon}{2\ell},$$

thus

$$\|\sigma(\boldsymbol{K}^i * \tilde{\boldsymbol{X}}^i) - \tilde{\boldsymbol{X}}^{i+1}\| \leq \frac{\varepsilon}{2\ell} \|\tilde{\boldsymbol{X}}^i\|.$$

By the reverse triangle inequality, the last inequality implies

$$\begin{aligned}
\|\tilde{\boldsymbol{X}}^{i+1}\| &\leq \frac{\varepsilon}{2\ell} \|\tilde{\boldsymbol{X}}^i\| + \|\sigma(\boldsymbol{K}^i * \tilde{\boldsymbol{X}}^i)\| \\
&\leq \frac{\varepsilon}{2\ell} \|\tilde{\boldsymbol{X}}^i\| + \|\boldsymbol{K}^i * \tilde{\boldsymbol{X}}^i\| \\
&\leq \frac{\varepsilon}{2\ell} \|\tilde{\boldsymbol{X}}^i\| + \|\boldsymbol{K}^i\|_1 \cdot \|\tilde{\boldsymbol{X}}^i\| \\
&\leq \frac{\varepsilon}{2\ell} \|\tilde{\boldsymbol{X}}^i\| + \|\tilde{\boldsymbol{X}}^i\| \\
&\leq \left(1 + \frac{\varepsilon}{2\ell}\right) \|\tilde{\boldsymbol{X}}^i\|.
\end{aligned}$$

Applying this inequality recursively, we get that $\|\tilde{\boldsymbol{X}}^i\| \leq \left(1 + \frac{\varepsilon}{2\ell}\right)^{i-1}$ for $1 \leq i \leq \ell - 1$. This allows us to bound the error between $\boldsymbol{X}^i$ and $\tilde{\boldsymbol{X}}^i$. For $1 \leq i \leq \ell - 1$, we have

$$\begin{aligned}
\|\boldsymbol{X}^{i+1} - \tilde{\boldsymbol{X}}^{i+1}\| &= \|\sigma(\boldsymbol{K}^i * \boldsymbol{X}^i) - \sigma(\tilde{\boldsymbol{L}}^{2i} * \sigma(\tilde{\boldsymbol{L}}^{2i-1} * \tilde{\boldsymbol{X}}^i))\| \\
&\leq \|\sigma(\boldsymbol{K}^i * \boldsymbol{X}^i) - \sigma(\boldsymbol{K}^i * \tilde{\boldsymbol{X}}^i)\| + \|\sigma(\boldsymbol{K}^i * \tilde{\boldsymbol{X}}^i) - \sigma(\tilde{\boldsymbol{L}}^{2i} * \sigma(\tilde{\boldsymbol{L}}^{2i-1} * \tilde{\boldsymbol{X}}^i))\| \\
&\leq \|\boldsymbol{K}^i\|_1 \|\boldsymbol{X}^i - \tilde{\boldsymbol{X}}^i\| + \|\sigma(\boldsymbol{K}^i * \tilde{\boldsymbol{X}}^i) - \sigma(\tilde{\boldsymbol{L}}^{2i} * \sigma(\tilde{\boldsymbol{L}}^{2i-1} * \tilde{\boldsymbol{X}}^i))\| \\
&\leq \|\boldsymbol{X}^i - \tilde{\boldsymbol{X}}^i\| + \|\sigma(\boldsymbol{K}^i * \tilde{\boldsymbol{X}}^i) - \sigma(\tilde{\boldsymbol{L}}^{2i} * \sigma(\tilde{\boldsymbol{L}}^{2i-1} * \tilde{\boldsymbol{X}}^i))\| \\
&\leq \|\boldsymbol{X}^i - \tilde{\boldsymbol{X}}^i\| + \left(1 + \frac{\varepsilon}{2\ell}\right)^{i-1} \frac{\varepsilon}{2\ell},
\end{aligned} \tag{G.28}$$

where for the last inequality we use equation (G.26). Unrolling equation (G.28) we get

$$\|\boldsymbol{X}^\ell - \tilde{\boldsymbol{X}}^\ell\| \leq \sum_{i=1}^{\ell-1} \Big(1 + \frac{\varepsilon}{2\ell}\Big)^{i-1} \frac{\varepsilon}{2\ell}.$$

Finally, this last inequality leads, with probability at least $1 - \varepsilon$, to

$$\begin{aligned}
\|f(\boldsymbol{X}) - g(\boldsymbol{X})\| &= \|\boldsymbol{K}^\ell * \boldsymbol{X}^\ell - \tilde{\boldsymbol{L}}^{2\ell} * \sigma(\tilde{\boldsymbol{L}}^{2\ell-1} * \tilde{\boldsymbol{X}}^\ell)\| \\
&\leq \|\boldsymbol{K}^\ell * \boldsymbol{X}^\ell - \boldsymbol{K}^\ell * \tilde{\boldsymbol{X}}^\ell\| + \|\boldsymbol{K}^\ell * \tilde{\boldsymbol{X}}^\ell - \tilde{\boldsymbol{L}}^{2\ell} * \sigma(\tilde{\boldsymbol{L}}^{2\ell-1} * \tilde{\boldsymbol{X}}^\ell)\| \\
&\leq \|\boldsymbol{K}^\ell\|_1 \|\boldsymbol{X}^\ell - \tilde{\boldsymbol{X}}^\ell\| + \|\boldsymbol{K}^\ell * \tilde{\boldsymbol{X}}^\ell - \tilde{\boldsymbol{L}}^{2\ell} * \sigma(\tilde{\boldsymbol{L}}^{2\ell-1} * \tilde{\boldsymbol{X}}^\ell)\| \\
&\leq \|\boldsymbol{X}^\ell - \tilde{\boldsymbol{X}}^\ell\| + \|\boldsymbol{K}^\ell * \tilde{\boldsymbol{X}}^\ell - \tilde{\boldsymbol{L}}^{2\ell} * \sigma(\tilde{\boldsymbol{L}}^{2\ell-1} * \tilde{\boldsymbol{X}}^\ell)\| \\
&\leq \|\boldsymbol{X}^\ell - \tilde{\boldsymbol{X}}^\ell\| + \Big(1 + \frac{\varepsilon}{2\ell}\Big)^{\ell-1} \frac{\varepsilon}{2\ell} \\
&\leq \left(\sum_{i=1}^{\ell-1} \Big(1 + \frac{\varepsilon}{2\ell}\Big)^{i-1} \frac{\varepsilon}{2\ell}\right) + \Big(1 + \frac{\varepsilon}{2\ell}\Big)^{\ell-1} \frac{\varepsilon}{2\ell} \\
&\leq \sum_{i=1}^{\ell} \Big(1 + \frac{\varepsilon}{2\ell}\Big)^{i-1} \frac{\varepsilon}{2\ell} \\
&= \Big(1 + \frac{\varepsilon}{2\ell}\Big)^{\ell} - 1 \\
&< e^{\varepsilon/2} - 1 \\
&< \varepsilon,
\end{aligned}$$

where the last inequality holds because $\varepsilon < 1$.

Replacing ε in this proof with $\min\{\varepsilon, \delta\}$ concludes the proof of the theorem. □

H Random Subset-Sum Theorem

For the sake of completeness, in this section we recall a result by Lueker (1998) together with the necessary definitions.

Definition H.1. Given two positive constants a and b, we say that a distribution with density φ contains a b-scaled $\mathrm{Uniform}([-a, a])$ distribution if for each $x \in [-a, a]$ it holds $\varphi(x) \geq b$. We simply say that a distribution F contains a uniform distribution if there exist positive constants a and b such that F contains a b-scaled $\mathrm{Uniform}([-a, a])$ distribution.

The following is a weaker version of Corollary 1 in the Appendix of Pensia et al. (2020).

Lemma H.1. *Let* X_1 *and* X_2 *be two independent random variables following a* $\mathrm{Uniform}([-1, 1])$ *distribution. Then* $\mathrm{X}_1 \cdot \mathrm{X}_2$ *contains a* $\frac{\log 2}{2}$*-scaled* $\mathrm{Uniform}([-\frac{1}{2}, \frac{1}{2}])$ *distribution.*

We say that z is 2η-subsetsum-approximated with $S = \{\mathrm{X}_1, \ldots, \mathrm{X}_n\}$ if there exists a subset $I_z \subseteq [n]$ such that $|\sum_{i \in I_z} \mathrm{X}_i - z| \leq 2\eta$.

Definition H.2. The $[a, b]$-subset-sum gap of $S = \{\mathrm{X}_1, \ldots, \mathrm{X}_n\}$ is the smallest value of η such that each $z \in [a, b]$, can be 2η-subsetsum-approximated with S.

Theorem H.2 (Corollary 3.3 in Lueker (1998)). *Let $S = \{\mathrm{X}_1, \ldots, \mathrm{X}_n\}$ be n i.i.d. bounded random variables and $\xi > 0$ any constant. Suppose that the distribution of X_1 contains a uniform distribution. Let $\mu_- = \mathbb{E}[\mathbf{1}_{\mathrm{X} \leq 0}\mathrm{X}]$, $\mu_+ = \mathbb{E}[\mathbf{1}_{\mathrm{X} \geq 0}\mathrm{X}]$, $\mu_{abs} = \mathbb{E}[|\mathrm{X}|] = \mu_+ - \mu_-$. The expected value of the $[(\mu_- + \xi)n, (\mu_+ - \xi)n]$-subset-sum gap of S is exponentially small with respect to n.*

Appendices of Chapter 5

I Technical tools

I.1 Concentration inequalities

Lemma I.1 (Most-probable normal interval)**.** *Let X follow a zero-mean normal distribution with variance* relu^2*. For any $z, \varepsilon \in \mathbb{R}$*

$$\Pr[X \in [z-\varepsilon, z+\varepsilon]] \leq \Pr[X \in [-\varepsilon, \varepsilon]].$$

Proof. Let $\varphi_X(x)$ denote the probability density function of X. Then,

$$\Pr[X \in [-\varepsilon, \varepsilon]] - \Pr[X \in [z-\varepsilon, z+\varepsilon]] = \int_{-\varepsilon}^{\varepsilon} \varphi_X(x)\,\mathrm{d}x - \int_{z-\varepsilon}^{z+\varepsilon} \varphi_X(x)\,\mathrm{d}x.$$

If $z-\varepsilon \geq \varepsilon$ or $z+\varepsilon \leq -\varepsilon$, the thesis is trivial as $\varphi_X(|x|)$ decreases in x. W.l.o.g., suppose z is positive and $z-\varepsilon < \varepsilon$. Then, $-\varepsilon < z-\varepsilon < \varepsilon < z+\varepsilon$. It follows that

$$\begin{aligned}\int_{-\varepsilon}^{\varepsilon} \varphi_X(x)\,\mathrm{d}x - \int_{z-\varepsilon}^{z+\varepsilon} \varphi_X(x)\,\mathrm{d}x &= \int_{-\varepsilon}^{z-\varepsilon} \varphi_X(x)\,\mathrm{d}x - \int_{\varepsilon}^{z+\varepsilon} \varphi_X(x)\,\mathrm{d}x \\ &= \int_{-\varepsilon}^{z-\varepsilon} \varphi_X(x) - \varphi_X(x+2\varepsilon)\,\mathrm{d}x\end{aligned}$$

which is non-negative as $\varphi_X(x) \geq \varphi_X(x+2\varepsilon)$ for $x \geq -\varepsilon$. □

Lemma I.2 (Second moment method)**.** *If Z is a non-negative random variable then*

$$\Pr[Z > 0] \geq \frac{\mathbb{E}[Z]^2}{\mathbb{E}[Z^2]}.$$

Lemma I.3 (Chernoff-Hoeffding bounds (Dubhashi & Panconesi, 2009))**.** *Let $X_1, X_2, \ldots, X_n$ be independent random variables such that $\Pr[0 \leq X_i \leq 1] = 1$ for all $i \in [n]$. Let $X = \sum_{i=1}^n X_i$ and $\mathbb{E}[X] = \mu$. Then, for any $\delta \in (0,1)$ the following holds:*

1. *if $\mu \leq \mu_+$, then $\Pr[X \geq (1+\delta)\mu_+] \leq \exp\left(-\frac{\delta^2\mu_+}{3}\right)$;*
2. *if $0 \leq \mu_- \leq \mu$, then $\Pr[X \leq (1-\delta)\mu_-] \leq \exp\left(-\frac{\delta^2\mu_+}{2}\right)$.*

Lemma I.4 (Corollary of (Laurent & Massart, 2000, Lemma 1))**.** *Let $X \sim \chi^2_d$ be a chi-squared random variable with d degrees of freedom. For any $t > 0$, it holds that*

1. $\Pr\left[X \geq d + 2\sqrt{dt} + 2t\right] \leq \exp(-t)$;
2. $\Pr\left[X \leq d - 2\sqrt{dt}\right] \leq \exp(-t)$.

I.2 Supporting results

Lemma I.5 (NSN with positive scalar)**.** *If a d-dimensional random vector Y is such that, for each $i \in [d]$, $Y_i = \tilde{\boldsymbol{Z}} \cdot \tilde{\boldsymbol{Z}}_i$, where $\tilde{\boldsymbol{Z}}_1, \dots, \tilde{\boldsymbol{Z}}_n$ are identically distributed random variables following a standard normal distribution, $\tilde{\boldsymbol{Z}}$ is a half-normal distribution,*[2] *and $\tilde{\boldsymbol{Z}}, \tilde{\boldsymbol{Z}}_1, \dots, \tilde{\boldsymbol{Z}}_n$ are independent, then Y follows an NSN distribution.*

Proof. By Definition 5.3.2, Y is NSN if, for each $i \in [d]$, $Y_i = Z \cdot Z_i$ where $Z, Z_1, \dots, Z_n$ are i.i.d. random variables following a standard normal distribution. If $\tilde{\boldsymbol{Z}} = |Z|$, we can rewrite $\tilde{\boldsymbol{Z}}_i = \operatorname{sign}(Z)\operatorname{sign}(Z_i)|Z_i|$ for each $i = 1, \dots, n$, where $Z, Z_1, \dots, Z_n$ are i.i.d. standard normal random variables, as $\operatorname{sign}(Z)\operatorname{sign}(Z_i)$ is independent of $\operatorname{sign}(Z)$ and of $\operatorname{sign}(Z)\operatorname{sign}(Z_j)$ for $i \neq j$. Then,

$$\begin{aligned} Y_i &= \tilde{\boldsymbol{Z}} \cdot \tilde{\boldsymbol{Z}}_i \\ &= |Z| \cdot \operatorname{sign}(Z)\operatorname{sign}(Z_i)|Z_i| \\ &= \operatorname{sign}(Z)|Z| \cdot \operatorname{sign}(Z_i)|Z_i| \\ &= Z \cdot Z_i, \end{aligned}$$

implying the thesis.

□

Corollary I.6 (of Theorem 5.3.2)**.** *Let d, k, and n be positive integers with $n \geq C_1 k^2 \log\left(\frac{1}{\varepsilon}\right)$ and $k \geq C_2 d^3 \log\frac{d}{\varepsilon}$ for some universal constants $C_1, C_2 \in \mathbb{R}_{>0}$. Let $X_1, \dots, X_n$ be d-dimensional i.i.d. NSN random vectors. For any $0 < \varepsilon \leq \frac{1}{4}$ and $\boldsymbol{z} \in \mathbb{R}^d$ with $\|\boldsymbol{z}\|_1 \leq \sqrt{k}$ it holds*

$$\Pr\left[\exists S : |S| = k,\ \|\left(\sum_{i \in S} X_i\right) - \boldsymbol{z}\|_\infty \leq \varepsilon\right] \geq 1 - \varepsilon.$$

Proof. Let $s = \left\lceil C_1 \log\left(\frac{1}{\varepsilon}\right)\right\rceil$ and let us partition the n vectors $X_1, \dots, X_n$ in s disjoint sets $G_1, \dots, G_s$ of at least k^2 vectors each. By Theorem 5.3.2, there is a constant $c \in (0, 1)$ such that for each group G_i ($i \in [s]$)

$$\Pr\left[\exists S \subset G_i : |S| = k,\ \|\boldsymbol{z} - \sum_{i \in S} X_i\|_\infty \leq \varepsilon\right] \geq c. \tag{I.29}$$

It follows that

$$\begin{aligned} &\Pr\left[\exists S : |S| = k,\ \|\left(\sum_{i \in S} X_i\right) - \boldsymbol{z}\|_\infty \leq \varepsilon\right] \\ \geq\ &\Pr\left[\exists i \in [s], \exists S \subset G_i : |S| = k,\ \|\left(\sum_{i \in S} X_i\right) - \boldsymbol{z}\|_\infty \leq \varepsilon\right] \\ =\ &1 - \Pr\left[\forall i \in [s], \forall S \subset G_i : |S| = k,\ \|\left(\sum_{i \in S} X_i\right) - \boldsymbol{z}\|_\infty > \varepsilon\right] \\ \geq\ &1 - (1 - c)^{\left\lceil C_1 \log\left(\frac{1}{\varepsilon}\right)\right\rceil}, \end{aligned}$$

[2] I.e., $\tilde{\boldsymbol{Z}} = |Z|$ where Z is a standard normal distribution.

where the latter inequality comes from equation (I.29) and the independence of the variables across different G_i. By choosing C_1 large enough,

$$1 - (1-c)^{\lceil C_1 \log(\frac{1}{\varepsilon})\rceil} \geq 1 - \varepsilon.$$

□

Lemma I.7 (Tensor Convolution Inequality)**.** *Given real tensors K and X of respective sizes $d \times d' \times c_0 \times c_1$ and $D \times D' \times c_0$, it holds*

$$\|K * X\|_\infty \leq \|K\|_1 \cdot \|X\|_\infty.$$

Proof. We have

$$\begin{aligned}
&\|K * X\|_\infty \\
&\leq \max_{i,j\in[D],\ell\in[c_1]} \sum_{i',j'\in[d],k\in[c]} \left|K_{i',j',k,\ell} X_{i-i'+1,j-j'+1,k}\right| \\
&\leq \max_{i,j\in[D],\ell\in[c_1]} \left(\sum_{i',j'\in[d],k\in[c]} \left|K_{i',j',k,\ell}\right|\right) \|X\|_\infty \\
&\leq \max_{i,j\in[D],\ell\in[c_1]} \|K\|_1 \cdot \|X\|_\infty \\
&= \|K\|_1 \cdot \|X\|_\infty.
\end{aligned}$$

□

J Omitted proofs and results

J.1 Multidimensional Random Subset Sum for normally-scaled normal vectors

Proof of Lemma 5.4.1. By Definition 5.3.2, the j-th entry of each vector X_i is $(X_i)_j = Z_i \cdot Z_{i,j}$ where each Z_i and $Z_{i,j}$ are i.i.d. random variables following a standard normal distribution. Let $\mathcal{E}^{(\updownarrow)}$ be the event that $k\left(1 - 2\sqrt{c_d}\right) \leq \sum_{i=1}^k Z_i^2 \leq k\left(1 + 2\sqrt{c_d} + 2c_d\right)$, and denote $X = \sum_{i=1}^k X_i$. By the law of total probability, it holds that

$$\Pr[X \in I_\varepsilon(\boldsymbol{z})] = \mathbb{E}_{Z_1,\ldots,Z_n}[\Pr[X \in I_\varepsilon(\boldsymbol{z}) \mid Z_1, \ldots, Z_k]].$$

As, conditional on $Z_1, \ldots, Z_k$, the d entries of X are independent, it follows that

$$\begin{aligned}
&\mathbb{E}_{Z_1,\ldots,Z_n}[\Pr[X \in I_\varepsilon(\boldsymbol{z}) \mid Z_1, \ldots, Z_k]] \\
&= \mathbb{E}_{Z_1,\ldots,Z_n}\left[\prod_{j=1}^d \Pr[(X)_j \in I_\varepsilon(z_i) \mid Z_1, \ldots, Z_k]\right] \\
&\geq \mathbb{E}_{Z_1,\ldots,Z_n}\left[\prod_{j=1}^d \Pr\left[(X)_j \in I_\varepsilon(z_i) \,\middle|\, Z_1, \ldots, Z_k, \mathcal{E}^{(\updownarrow)}\right]\right] \Pr\left[\mathcal{E}^{(\updownarrow)}\right], \qquad \text{(J.30)}
\end{aligned}$$

where the inequality in equation (J.30) holds by applying again the law of total probability.

Conditional on $Z_1, \dots, Z_k$, we have that $(X)_j \sim \mathcal{N}\, 0, \sum_{i=1}^k Z_i^2$. Hence,

$$\begin{aligned}
&\mathbb{E}_{Z_1,\dots,Z_n}\left[\prod_{j=1}^{d} \Pr\left[(X)_j \in I_\varepsilon(z_i) \;\middle|\; Z_1, \dots, Z_k, \mathcal{E}^{(\updownarrow)}\right]\right] \Pr\left[\mathcal{E}^{(\updownarrow)}\right] \\
&\geq \mathbb{E}_{Z_1,\dots,Z_k}\left[\prod_{j=1}^{d}\left(\frac{2\varepsilon}{\sqrt{\pi\left(\sum_{i=1}^k Z_i^2\right)}} \exp\left(-\frac{(|z_i|+\varepsilon)^2}{2\sum_{i=1}^k Z_i^2}\right)\right) \;\middle|\; Z_1, \dots, Z_k, \mathcal{E}^{(\updownarrow)}\right] \\
&\quad \cdot \Pr\left[\mathcal{E}^{(\updownarrow)}\right]
\end{aligned}$$

Notice that the term $\sum_i Z_i^2$ is a sum of chi-square random variables, for which there are known concentration bounds (Lemma I.4). By definition of $\mathcal{E}^{(\updownarrow)}$ and by applying Lemma I.4 to estimate the term $\Pr\left[\mathcal{E}^{(\updownarrow)}\right]$, we get that

$$\begin{aligned}
&\mathbb{E}_{Z_1,\dots,Z_k}\left[\prod_{j=1}^{d}\left(\frac{2\varepsilon}{\sqrt{\pi\left(\sum_{i=1}^k Z_i^2\right)}} \exp\left(-\frac{(|z_i|+\varepsilon)^2}{2\sum_{i=1}^k Z_i^2}\right)\right) \;\middle|\; Z_1, \dots, Z_k, \mathcal{E}^{(\updownarrow)}\right] \\
&\quad \cdot \Pr\left[\mathcal{E}^{(\updownarrow)}\right] \\
&\geq \left(\frac{2\varepsilon}{\sqrt{\pi\left(1+2\sqrt{c_d}+2c_d\right)k}}\right)^d \exp\left(-\frac{\sum_i |z_i|^2 + 2\varepsilon\sum_i |z_i| + d\varepsilon^2}{2\left(1-2\sqrt{c_d}\right)k}\right) \Pr\left[\mathcal{E}^{(\updownarrow)}\right] \\
&= \left(\frac{2\varepsilon}{\sqrt{\pi\left(1+2\sqrt{c_d}+2c_d\right)k}}\right)^d \exp\left(-\frac{\|\boldsymbol{z}\|_2^2 + 2\varepsilon\|\boldsymbol{z}\|_1 + d\varepsilon^2}{2\left(1-2\sqrt{c_d}\right)k}\right) \Pr\left[\mathcal{E}^{(\updownarrow)}\right] \\
&\geq \left(\frac{2\varepsilon}{\sqrt{\pi\left(1+2\sqrt{c_d}+2c_d\right)k}}\right)^d \exp\left(-\frac{\|\boldsymbol{z}\|_2^2 + 2\varepsilon\|\boldsymbol{z}\|_1 + d\varepsilon^2}{2\left(1-2\sqrt{c_d}\right)k}\right) \left(1-2e^{-c_d k}\right).
\end{aligned}$$

As $c_d k \geq 1$ by hypotheses, $1 - 2e^{-c_d k} \geq 1/4$. Then,

$$
\begin{aligned}
&\left(\frac{2\varepsilon}{\sqrt{\pi\left(1+2\sqrt{c_d}+2c_d\right)k}}\right)^d \exp\left(-\frac{\|\boldsymbol{z}\|_2^2+2\varepsilon\|\boldsymbol{z}\|_1+d\varepsilon^2}{2\left(1-2\sqrt{c_d}\right)k}\right)\left(1-2e^{-c_d k}\right)\\
&\geq \frac{1}{4}\left(\frac{2\varepsilon}{\sqrt{\pi\left(1+2\sqrt{c_d}+2c_d\right)k}}\right)^d \exp\left(-\frac{\|\boldsymbol{z}\|_2^2+2\varepsilon\|\boldsymbol{z}\|_1+d\varepsilon^2}{2\left(1-2\sqrt{c_d}\right)k}\right)\\
&\geq \frac{1}{4}\left(\frac{2\varepsilon}{\sqrt{\pi\left(1+2\sqrt{c_d}+2c_d\right)k}}\right)^d \exp\left(-\frac{k+2\varepsilon\sqrt{k}+d\varepsilon^2}{2\left(1-2\sqrt{c_d}\right)k}\right) \qquad \text{(J.31)}\\
&= \frac{1}{4}\left(\frac{2\varepsilon}{\sqrt{\pi\left(1+2\sqrt{c_d}+2c_d\right)k}}\right)^d \exp\left(-\frac{1+\frac{2\varepsilon}{\sqrt{k}}+\frac{d\varepsilon^2}{k}}{2\left(1-2\sqrt{c_d}\right)}\right)\\
&\geq \frac{1}{4}\left(\frac{2\varepsilon}{\sqrt{\pi\left(1+2\sqrt{c_d}+2c_d\right)k}}\right)^d \exp\left(-\frac{1+\frac{1}{8}+\frac{1}{16}}{2\left(1-2\sqrt{c_d}\right)}\right) \qquad \text{(J.32)}\\
&\geq \frac{1}{16}\left(\frac{2\varepsilon}{\sqrt{\pi\left(1+2\sqrt{c_d}+2c_d\right)k}}\right)^d, \qquad \text{(J.33)}
\end{aligned}
$$

where we have used that $\|\boldsymbol{z}\|_2 \leq \|\boldsymbol{z}\|_1 \leq \sqrt{k}$ in equation (J.31), that $k \geq 16$, $k \geq d$, that $\varepsilon < 1/4$ in equation (J.32), and that

$$
\exp\left(-\frac{1+\frac{1}{8}+\frac{1}{16}}{2\left(1-2\sqrt{c_d}\right)}\right) \geq \exp\left(-\frac{1+\frac{1}{8}+\frac{1}{16}}{2\left(1-2\sqrt{\frac{1}{16}}\right)}\right) \geq \frac{1}{16}
$$

in equation (J.33). □

Proof of Lemma 5.4.2. Since the X_is are NSN random vectors, for each $i \in [n]$ and $j \in [d]$ we can write the j-th entry of X_i as $(X_i)_j = Z_i \cdot Z_{i,j}$ where the variables in $\{Z_i\}_{i\in[n]}$ and in $\{Z_{i,j}\}_{i\in[n],j\in[d]}$ are i.i.d. random variables following a standard normal distribution. By the law of total probability, we have

$$
\begin{aligned}
&\Pr[\mathcal{A}+\mathcal{B}\in I_\varepsilon(\boldsymbol{z}),\mathcal{B}+\mathcal{C}\in I_\varepsilon(\boldsymbol{z})]\\
&= \mathbb{E}_{Z_1,\dots,Z_n}[\Pr[\mathcal{A}+\mathcal{B}\in I_\varepsilon(\boldsymbol{z}),\mathcal{B}+\mathcal{C}\in I_\varepsilon(\boldsymbol{z})\mid Z_1,\dots,Z_n]]\\
&= \mathbb{E}_{Z_1,\dots,Z_n}\left[\prod_{i=1}^d \Pr[\mathcal{A}_i+\mathcal{B}_i\in I_\varepsilon(z_i),\mathcal{B}_i+\mathcal{C}_i\in I_\varepsilon(z_i)\mid Z_1,\dots,Z_n]\right], \qquad \text{(J.34)}
\end{aligned}
$$

where the latter equality holds by independence.

Then,

$$
\begin{aligned}
&\Pr[\mathcal{A}_i+\mathcal{B}_i\in I_\varepsilon(z_i),\mathcal{B}_i+\mathcal{C}_i\in I_\varepsilon(z_i)\mid Z_1,\dots,Z_n]\\
&= \mathbb{E}_{B_i}[\Pr[\mathcal{A}_i\in I_\varepsilon(z_i-\mathcal{B}_i),\mathcal{C}_i\in I_\varepsilon(z_i-\mathcal{B}_i)\mid Z_1,\dots,Z_n,B_i]]\\
&= \mathbb{E}_{B_i}[\Pr[\mathcal{A}_i\in I_\varepsilon(z_i-\mathcal{B}_i)\mid Z_1,\dots,Z_n,B_i]\Pr[\mathcal{C}_i\in I_\varepsilon(z_i-\mathcal{B}_i)\mid Z_1,\dots,Z_n,B_i]],
\end{aligned}
$$

where the latter inequality holds by independence of A_i and C_i. By Lemma I.1, it holds that

$$\begin{aligned}
&\mathbb{E}_{B_i}[\Pr[\mathcal{A}_i \in I_\varepsilon(z_i - \mathcal{B}_i) \mid Z_1, \dots, Z_n, B_i] \Pr[\mathcal{C}_i \in I_\varepsilon(z_i - \mathcal{B}_i) \mid Z_1, \dots, Z_n, B_i]] \\
&\leq \mathbb{E}_{B_i}[\Pr[\mathcal{A}_i \in I_\varepsilon(0) \mid Z_1, \dots, Z_n, B_i] \Pr[\mathcal{C}_i \in I_\varepsilon(0) \mid Z_1, \dots, Z_n, B_i]] \\
&= \Pr[\mathcal{A}_i \in I_\varepsilon(0) \mid Z_1, \dots, Z_n] \Pr[\mathcal{C}_i \in I_\varepsilon(0) \mid Z_1, \dots, Z_n] \\
&\leq \frac{2\varepsilon}{\sqrt{\pi\left(\sum_{r=1}^{j} Z_r^2\right)}} \cdot \frac{2\varepsilon}{\sqrt{\pi\left(\sum_{r=k+1}^{k+j} Z_r^2\right)}},
\end{aligned} \tag{J.35}$$

where the latter inequality comes from the fact that, conditioned on $Z_1, \dots, Z_n$, we have that $\mathcal{A}_i \sim \mathcal{N}\,0, \sum_{r=1}^{j} Z_r^2$, $\mathcal{B}_i \sim \mathcal{N}\,0, \sum_{r=j+1}^{k} Z_r^2$, and $\mathcal{C}_i \sim \mathcal{N}\,0, \sum_{r=k+1}^{k+j} Z_r^2$ for each $i \in [d]$.

We now proceed similarly to the proof of Lemma 5.4.1. We denote the event that $(1 - 2\sqrt{c_d})\, j \leq \sum_{i=1}^{j} Z_i^2, \sum_{i=k+1}^{k+j} Z_i^2$ by $\mathcal{E}^{(\downarrow)}$. Then, by equation (J.34) and the law of total probability, we have that

$$\begin{aligned}
&\Pr[\mathcal{A} + \mathcal{B} \in I_\varepsilon(\boldsymbol{z}), \mathcal{B} + \mathcal{C} \in I_\varepsilon(\boldsymbol{z})] \\
&= \mathbb{E}_{Z_1,\dots,Z_n}\left[\prod_{i=1}^{d} \Pr[\mathcal{A}_i + \mathcal{B}_i \in I_\varepsilon(z_i), \mathcal{B}_i + \mathcal{C}_i \in I_\varepsilon(z_i) \mid Z_1, \dots, Z_n]\right] \\
&\leq \mathbb{E}_{Z_1,\dots,Z_n}\left[\prod_{i=1}^{d} \Pr\left[\mathcal{A}_i + \mathcal{B}_i, \mathcal{B}_i + \mathcal{C}_i \in I_\varepsilon(z_i) \,\middle|\, Z_1, \dots, Z_n, \mathcal{E}^{(\downarrow)}\right]\right] + \Pr\left[\overline{\mathcal{E}^{(\downarrow)}}\right].
\end{aligned}$$

Equation (J.35) implies that

$$\begin{aligned}
&\mathbb{E}_{Z_1,\dots,Z_n}\left[\prod_{i=1}^{d} \Pr\left[\mathcal{A}_i + \mathcal{B}_i, \mathcal{B}_i + \mathcal{C}_i \in I_\varepsilon(z_i) \,\middle|\, Z_1, \dots, Z_n, \mathcal{E}^{(\downarrow)}\right]\right] + \Pr\left[\overline{\mathcal{E}^{(\downarrow)}}\right] \\
&\leq \mathbb{E}_{Z_1,\dots,Z_n}\left[\prod_{i=1}^{d} \frac{2\varepsilon}{\sqrt{\pi\left(\sum_{r=1}^{j} Z_r^2\right)}} \cdot \frac{2\varepsilon}{\sqrt{\pi\left(\sum_{r=k+1}^{k+j} Z_r^2\right)}} \,\middle|\, \mathcal{E}^{(\downarrow)}\right] + \Pr\left[\overline{\mathcal{E}^{(\downarrow)}}\right] \\
&= \mathbb{E}_{Z_1,\dots,Z_n}\left[\left(\frac{4\varepsilon^2}{\pi\sqrt{\left(\sum_{r=1}^{j} Z_r^2\right)\left(\sum_{r=k+1}^{k+j} Z_r^2\right)}}\right)^d \,\middle|\, \mathcal{E}^{(\downarrow)}\right] + \Pr\left[\overline{\mathcal{E}^{(\downarrow)}}\right].
\end{aligned}$$

By independence of $\sum_{r=1}^{j} Z_r^2$ and $\sum_{r=k+1}^{k+j} Z_r^2$ and by Lemma I.4, we obtain that

$$
\begin{aligned}
&\mathbb{E}_{Z_1,\dots,Z_n}\left[\left(\frac{4\varepsilon^2}{\pi\sqrt{\left(\sum_{i=1}^{j} Z_i^2\right)\left(\sum_{i=k+1}^{k+j} Z_i^2\right)}}\right)^d \,\middle|\, \mathcal{E}^{(\downarrow)}\right] + \Pr\left[\overline{\mathcal{E}^{(\downarrow)}}\right] \\
&\leq \left(\frac{4\varepsilon^2}{\pi j\left(1-2\sqrt{c_d}\right)}\right)^d + \Pr\left[\overline{\mathcal{E}^{(\downarrow)}}\right] \\
&\leq \left(\frac{4\varepsilon^2}{\pi j\left(1-2\sqrt{c_d}\right)}\right)^d + 2\exp(-c_d j) \\
&= \exp\left(-d\log\frac{\pi j\left(1-2\sqrt{c_d}\right)}{4\varepsilon^2}\right) + 2\exp(-c_d k) \\
&\leq 3\left(\frac{4\varepsilon^2}{\pi\left(1-2\sqrt{c_d}\right)k}\right)^d.
\end{aligned}
$$

Finally, for a large enough constant C, the hypothesis on k implies that $k \geq 2\frac{d}{c_d}\log\frac{\pi k\left(1-2\sqrt{c_d}\right)}{4\varepsilon^2}$. Hence,

$$
\exp\left(-d\log\frac{\pi j\left(1-2\sqrt{c_d}\right)}{4\varepsilon^2}\right) + 2\exp(-c_d k) \leq 3\left(\frac{4\varepsilon^2}{\pi\left(1-2\sqrt{c_d}\right)k}\right)^d.
$$

□

Proof of Theorem 5.3.2. We use the second moment method (Lemma I.2) on the ε-subset-sum number $T_{n,k}$ of $X_1,\dots,X_n$. Thus, we aim to provide a lower bound on the right-hand side of

$$
\Pr[T>0] \geq \frac{\mathbb{E}[T_{n,k}]^2}{\mathbb{E}[T_{n,k}^2]}. \tag{J.36}
$$

Equivalently, we can provide an upper bound on the inverse $\frac{\mathbb{E}[T_{n,k}^2]}{\mathbb{E}[T_{n,k}]^2}$. By Lemma 5.4.3

$$
\mathbb{E}[T_{n,k}^2] = \binom{n}{k}^2 \sum_{j=0}^{k} \Pr\left[|\mathcal{S}\cap\mathcal{S}'| = k-j\right] \Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})} \cap \mathcal{E}_{\mathcal{S}_j}^{(\boldsymbol{v})}\right] \tag{J.37}
$$

where $\mathcal{S},\mathcal{S}',\mathcal{S}_i$ and $\mathcal{E}_{\mathcal{S}}^{(\boldsymbol{z})}$ are defined as in the statement of the lemma. Observe also that

$$
\mathbb{E}[T_{n,k}] = \sum_{\mathcal{S}\in\binom{[n]}{k}} \mathbb{E}[\mathbf{1}_{\mathcal{E}_{\mathcal{S}}^{(\boldsymbol{z})}}] = \sum_{\mathcal{S}\in\binom{[n]}{k}} \Pr\left[\mathcal{E}_{\mathcal{S}}^{(\boldsymbol{z})}\right] = \binom{n}{k}\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})}\right]. \tag{J.38}
$$

By equations (J.37) and (J.38), we have

$$
\begin{aligned}
\frac{\mathbb{E}[T_{n,k}^2]}{\mathbb{E}[T_{n,k}]^2} &= \frac{\binom{n}{k}^2}{\mathbb{E}[T_{n,k}]^2}\sum_{j=0}^{k}\Pr\left[|\mathcal{S}\cap\mathcal{S}'| = k-j\right]\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})}\cap\mathcal{E}_{\mathcal{S}_j}^{(\boldsymbol{v})}\right] \\
&= \sum_{j=0}^{k}\Pr\left[|\mathcal{S}\cap\mathcal{S}'| = k-j\right]\frac{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})}\cap\mathcal{E}_{\mathcal{S}_j}^{(\boldsymbol{v})}\right]}{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})}\right]^2}.
\end{aligned} \tag{J.39}
$$

As for the denominator of the second term in equation (J.39), by Lemma 5.4.1 we have

$$\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})}\right]^2 \geq \frac{1}{256}\left(\frac{4\varepsilon^2}{\pi\left(1+2\sqrt{c_d}+2c_d\right)k}\right)^d. \tag{J.40}$$

As for the numerator of the second term in equation (J.39), Lemma 5.4.2 implies that we have

$$\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})} \cap \mathcal{E}_{\mathcal{S}_j}^{(\boldsymbol{v})}\right] \leq 3\left(\frac{4\varepsilon^2}{\pi\left(1-2\sqrt{c_d}\right)j}\right)^d. \tag{J.41}$$

By plugging equations (J.40) and (J.41) in equation (J.39), for $j \geq k\left(1-\frac{1}{d}\right)$ and $d > 1$ we can upper bound the factor $\frac{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})} \cap \mathcal{E}_{\mathcal{S}_j}^{(\boldsymbol{v})}\right]}{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})}\right]^2}$ of the summation as follows:

$$\begin{aligned}\frac{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})} \cap \mathcal{E}_{\mathcal{S}_j}^{(\boldsymbol{v})}\right]}{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})}\right]^2} &\leq \frac{3\left(\frac{4\varepsilon^2}{\pi\left(1-2\sqrt{c_d}\right)j}\right)^d}{\frac{1}{256}\left(\frac{4\varepsilon^2}{\pi\left(1+2\sqrt{c_d}+2c_d\right)k}\right)^d} \\ &= 768\left(\frac{\left(1+2\sqrt{c_d}+2c_d\right)k}{\left(1-2\sqrt{c_d}\right)j}\right)^d.\end{aligned}$$

As $j \geq k\left(1-\frac{1}{d}\right)$ with $d > 1$, then

$$\begin{aligned}768\left(\frac{\left(1+2\sqrt{c_d}+2c_d\right)k}{\left(1-2\sqrt{c_d}\right)j}\right)^d &\leq 768\left(\frac{\left(1+2\sqrt{c_d}+2c_d\right)}{\left(1-2\sqrt{c_d}\right)\left(1-\frac{1}{d}\right)}\right)^d \\ &\leq 768\left(\frac{2+7\sqrt{c_d}}{2-7\sqrt{c_d}}\right)^d \\ &\leq 270801,\end{aligned} \tag{J.42}$$

because $\left(\frac{\left(1+2\sqrt{c_d}+2c_d\right)}{\left(1-2\sqrt{c_d}\right)\left(1-\frac{1}{d}\right)}\right)^d$ is maximized or $d = 4$. Let $C' = 270801$. If $d > 1$, by plugging equation (J.42) in equation (J.39), we have that

$$\begin{aligned}\frac{\mathbb{E}[T_{n,k}^2]}{\mathbb{E}[T_{n,k}]^2} &= \sum_{j=0}^{\lceil k-\frac{k}{d}\rceil-1} \Pr\left[|\mathcal{S}\cap\mathcal{S}'| = k-j\right]\frac{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})} \cap \mathcal{E}_{\mathcal{S}_j}^{(\boldsymbol{v})}\right]}{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})}\right]^2} \\ &\quad + \sum_{j=\lceil k-\frac{k}{d}\rceil}^{k} \Pr\left[|\mathcal{S}\cap\mathcal{S}'| = k-j\right]\frac{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})} \cap \mathcal{E}_{\mathcal{S}_j}^{(\boldsymbol{v})}\right]}{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})}\right]^2} \\ &\leq \sum_{j=0}^{\lceil k-\frac{k}{d}\rceil-1} \Pr\left[|\mathcal{S}\cap\mathcal{S}'| = k-j\right]\frac{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})} \cap \mathcal{E}_{\mathcal{S}_j}^{(\boldsymbol{v})}\right]}{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(\boldsymbol{v})}\right]^2} + C'.\end{aligned}$$

As $\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(v)} \cap \mathcal{E}_{\mathcal{S}_j}^{(v)}\right] \leq \Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(v)}\right]$, then

$$\begin{aligned}
&\sum_{j=0}^{\lceil k-\frac{k}{d}\rceil-1} \Pr[|\mathcal{S}\cap\mathcal{S}'| = k-j] \frac{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(v)} \cap \mathcal{E}_{\mathcal{S}_j}^{(v)}\right]}{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(v)}\right]^2} + C' \\
&\leq \frac{1}{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(v)}\right]} \sum_{j=0}^{\lceil k-\frac{k}{d}\rceil-1} \Pr[|\mathcal{S}\cap\mathcal{S}'| = k-j] + C' \\
&\leq \frac{\Pr\left[|\mathcal{S}\cap\mathcal{S}'| > \frac{k}{d}\right]}{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(v)}\right]} + C'. \qquad \text{(J.43)}
\end{aligned}$$

Notice that, if $d = 1$, the same bound holds as $\Pr\left[|\mathcal{S}\cap\mathcal{S}'| > \frac{k}{d}\right] = 0$. We now observe that, by the law of total probability

$$\Pr\left[|\mathcal{S}\cap\mathcal{S}'| \geq \frac{k}{d}\right] = \sum_{\tilde{\mathcal{S}}\in\binom{[n]}{k}} \Pr\left[\mathcal{S} = \tilde{\mathcal{S}}\right] \Pr\left[|\mathcal{S}\cap\mathcal{S}'| \geq \frac{k}{d} \,\middle|\, \mathcal{S} = \tilde{\mathcal{S}}\right]. \qquad \text{(J.44)}$$

Conditional on $\mathcal{S} = \tilde{\mathcal{S}}$, $|\mathcal{S}\cap\mathcal{S}'|$ is a hypergeometric random variable with

$$\mathbb{E}[|\mathcal{S}\cap\mathcal{S}'| \,|\, \mathcal{S} = \tilde{\mathcal{S}}] = \sum_{i\in\tilde{\mathcal{S}}} \Pr[i \in \mathcal{S}'] = k \Pr[1 \in \mathcal{S}] = \frac{k^2}{n}.$$

Since $n \geq k^2$, then $\frac{k^2}{n} \leq 1$. Hence, since Chernoff bounds holds for the hypergeometric distribution (Doerr, 2020, Theorem 1.10.25)

$$\begin{aligned}
\Pr\left[|\mathcal{S}\cap\mathcal{S}'| \geq \frac{k}{d} \,\middle|\, \mathcal{S} = \tilde{\mathcal{S}}\right] &\geq \Pr\left[|\mathcal{S}\cap\mathcal{S}'| \geq \frac{k^2}{n} + \left(\frac{k}{d} - 1\right) \,\middle|\, \mathcal{S} = \tilde{\mathcal{S}}\right] \\
&\leq \exp\left(-2\frac{\left(\frac{k}{d} - 1\right)^2}{k}\right) \\
&\leq \exp\left(-2\frac{k}{d^2}\left(1 - \frac{d}{k}\right)^2\right). \qquad \text{(J.45)}
\end{aligned}$$

Substituting equation (J.45) in equation (J.44) we get

$$\Pr\left[|\mathcal{S}\cap\mathcal{S}'| \geq \frac{k}{d}\right] \leq \exp\left(-2\frac{k}{d^2}\left(1 - \frac{d}{k}\right)^2\right). \qquad \text{(J.46)}$$

We can now keep bounding from above $\frac{\mathbb{E}[T_{n,k}^2]}{\mathbb{E}[T_{n,k}]^2}$ by plugging equation (J.46) in equation (J.43):

$$\frac{\Pr\left[|\mathcal{S}\cap\mathcal{S}'| \geq \frac{k}{d}\right]}{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(v)}\right]} + C' \leq \frac{\exp\left(-2\frac{k}{d^2}\left(1 - \frac{d}{k}\right)^2\right)}{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(v)}\right]} + C'.$$

By Lemma 5.4.1, and since $1 + 2\sqrt{c_d} + 2c_d \leq 2$, we have

$$\begin{aligned}
&\frac{\exp\left(-2\frac{k}{d^2}\left(1-\frac{d}{k}\right)^2\right)}{\Pr\left[\mathcal{E}_{\mathcal{S}_0}^{(v)}\right]} + C' \\
&\leq \frac{16\exp\left(-2\frac{k}{d^2}\left(1-\frac{d}{k}\right)^2\right)}{\left(\frac{2\varepsilon}{\sqrt{\pi 2k}}\right)^d} + C' \\
&= 16\exp\left(-2\frac{k}{d^2}\left(1-\frac{d}{k}\right)^2 + d\log\left(\frac{\sqrt{\pi 2k}}{2\varepsilon}\right)\right) + C'.
\end{aligned}$$

By the hypothesis, since $k \geq Cd^3 \log\frac{d}{\varepsilon}$ for a large enough C, it holds that

$$\begin{aligned}
&16\exp\left(-2\frac{k}{d^2}\left(1-\frac{d}{k}\right)^2 + d\log\left(\frac{\sqrt{\pi 2k}}{2\varepsilon}\right)\right) + C' \\
&\leq C' + 16\exp\left(-d\log\frac{d}{\varepsilon}\right) \\
&< C' + \frac{16}{e},
\end{aligned} \tag{J.47}$$

where the latter inequality holds as $\varepsilon \leq 1/4$.

Plugging the inverse of the expression in equation (J.47) in equation (J.36) we obtain the thesis. □

J.2 Kernel Pruning

Proof of Lemma 5.4.4. $\mathcal{S} \in \{0,1\}^{\text{shape}(V)}$ is such that $\mathcal{V} = V \odot \mathcal{S}$ contains only non-negative edges going from each input channel t to the output channels $(t-1)n+1, \ldots, tn$, and only non-positive[3] edges going from each input channel t to the output channels $tn+1, \ldots, 2tn$, while all remaining edges are set to zero. Let us define some convenient notations before proceeding with the proofs. By $[n, m]$ we denote the set $\{n, n+1, \ldots, m\}$ for each pair of integers $n \leq m \in \mathbb{N}$. In formulas, we obtain a tensor $\mathcal{V}$ such that, for each $(t,k) \in [c], \times[2nc]$:

$$(V \odot \mathcal{S})_{1,1,t,k} = \begin{cases} \boldsymbol{V}_{1,1,t,k} \cdot \mathbf{1}_{\boldsymbol{V}_{1,1,t,k}>0} & \text{if } k \in [(2t-2)n+1, (2t-1)n], \\ \boldsymbol{V}_{1,1,t,k} \cdot \mathbf{1}_{\boldsymbol{V}_{1,1,t,k}<0} & \text{if } k \in [(2t-1)n+1, 2tn], \\ 0 & \text{otherwise.} \end{cases} \tag{J.48}$$

To simplify the notation, we define the following indicator functions: for any $(t,k) \in [c] \times [2nc]$,

$$\begin{aligned}
&\mathbf{1}_{\frac{k}{2n} \in \left(t-1, t-\frac{1}{2}\right]} = 1 \text{ iff } k \in [(2t-2)n+1, (2t-1)n], \text{ and} \\
&\mathbf{1}_{\frac{k}{2n} \in \left(t-\frac{1}{2}, t\right]} = 1 \text{ iff } k \in [(2t-1)n+1, 2tn].
\end{aligned} \tag{J.49}$$

[3] We consider 0 to be both non-negative and non-positive.

For each $(i,j,k) \in [D] \times [D] \times [2nc]$, applying equation (J.48) and Definition 5.4.2, it then holds

$$\begin{aligned}
&\left(\text{relu}\left((V \odot \mathcal{S}) * X\right)\right)_{i,j,k} \\
&= \text{relu}\left(\sum_{t=1}^{c_0} \mathcal{V}_{1,1,t,k} X_{i,j,t}\right) \\
&= \text{relu}\left(\sum_{t=1}^{c_0} \left(V_{1,1,t,k} X_{i,j,t} \cdot \mathbf{1}_{\boldsymbol{V}_{1,1,t,k}>0} \mathbf{1}_{\frac{k}{2n} \in \left(t-1, t-\frac{1}{2}\right]}\right.\right. \\
&\quad \left.\left. + V_{1,1,t,k} X_{i,j,t} \cdot \mathbf{1}_{\boldsymbol{V}_{1,1,t,k}<0} \mathbf{1}_{\frac{k}{2n} \in \left(t-\frac{1}{2}, t\right]}\right)\right) \\
&= \text{relu}\left(\sum_{t=1}^{c_0} \left(V^+_{1,1,t,k} X_{i,j,t} \mathbf{1}_{\frac{k}{2n} \in \left(t-1, t-\frac{1}{2}\right]} - V^-_{1,1,t,k} X_{i,j,t} \mathbf{1}_{\frac{k}{2n} \in \left(t-\frac{1}{2}, t\right]}\right)\right) \\
&= \text{relu}\left(\sum_{t=1}^{c_0} \left(V^+_{1,1,t,k} (X^+_{i,j,t} - X^-_{i,j,t}) \mathbf{1}_{\frac{k}{2n} \in \left(t-1, t-\frac{1}{2}\right]}\right.\right. \\
&\quad \left.\left. + V^-_{1,1,t,k} (X^-_{i,j,t} - X^+_{i,j,t}) \mathbf{1}_{\frac{k}{2n} \in \left(t-\frac{1}{2}, t\right]}\right)\right). \qquad \text{(J.50)}
\end{aligned}$$

Observe that only one term survives in the summation in equation (J.50), as there exists only one $t \in [c_0]$ such that $k \in [(2t-2)n+1, 2tn]$, say $t^\star$. Moreover, out of the four additive terms in the expression

$$V^+_{1,1,t^\star,k} (X^+_{i,j,t^\star} - X^-_{i,j,t^\star}) \mathbf{1}_{\frac{k}{2n} \in \left(t^\star-1, t^\star-\frac{1}{2}\right]} + V^-_{1,1,t^\star,k} (X^-_{i,j,t^\star} - X^+_{i,j,t^\star}) \mathbf{1}_{\frac{k}{2n} \in \left(t^\star-\frac{1}{2}, t^\star\right]},$$

at most one is non-zero, due to Definition 5.4.2. The ReLU cancels out negative ones, implying that equation (J.50) can be rewritten without the ReLU as a sum of only non-negative terms (out of which, at most one is non-zero) as follows

$$\begin{aligned}
&\text{relu}\left(\sum_{t=1}^{c_0} \left(V^+_{1,1,t,k} (X^+_{i,j,t} - X^-_{i,j,t}) \mathbf{1}_{\frac{k}{2n} \in \left(t-1, t-\frac{1}{2}\right]}\right.\right. \\
&\quad \left.\left. + V^-_{1,1,t,k} (X^-_{i,j,t} - X^+_{i,j,t}) \mathbf{1}_{\frac{k}{2n} \in \left(t-\frac{1}{2}, t\right]}\right)\right) \\
&= \sum_{t=1}^{c_0} \left(V^+_{1,1,t,k} X^+_{i,j,t} \mathbf{1}_{\frac{k}{2n} \in \left(t-1, t-\frac{1}{2}\right]} + V^-_{1,1,t,k} X^-_{i,j,t} \mathbf{1}_{\frac{k}{2n} \in \left(t-\frac{1}{2}, t\right]}\right). \qquad \text{(J.51)}
\end{aligned}$$

Finally, by equations (J.48) and (J.49), $\mathcal{V}^+_{1,1,t,k} = 0$ if $\frac{k}{2n} \notin \left(t-1, t-\frac{1}{2}\right]$, and $\mathcal{V}^-_{1,1,t,k} = 0$ if $\frac{k}{2n} \in \left(t-\frac{1}{2}, t\right]$, which means that in equation (J.51) we can ignore the indicator functions and

further simplify the expression as

$$\begin{aligned}
&\sum_{t=1}^{c_0}\left(V^+_{1,1,t,k}X^+_{i,j,t}\mathbf{1}_{\frac{k}{2n}\in\left(t-1,t-\frac{1}{2}\right]}+V^-_{1,1,t,k}X^-_{i,j,t}\mathbf{1}_{\frac{k}{2n}\in\left(t-\frac{1}{2},t\right]}\right)\\
&=\sum_{t=1}^{c_0}\left(\mathcal{V}^+_{1,1,t,k}X^+_{i,j,t}+\mathcal{V}^-_{1,1,t,k}X^-_{i,j,t}\right)\\
&=\left(\sum_{t=1}^{c_0}\mathcal{V}^+_{1,1,t,k}X^+_{i,j,t}+\sum_{t=1}^{c_0}\mathcal{V}^-_{1,1,t,k}X^-_{i,j,t}\right)\\
&=\left(\mathcal{V}^+ * X^+ + \mathcal{V}^- * X^-\right)_{i,j,k}.
\end{aligned}$$

□

Proof of Lemma 5.4.5. Adopting the same definitions as in Lemma 5.4.4 (and equation (J.48)), for each $(r,s,t_1)\in[d]\times[d]\times[c_1]$ we have, by Lemma 5.4.4,

$$\begin{aligned}
&\left(U * \operatorname{relu}\left((V\odot\mathcal{S})*X\right)\right)_{r,s,t_1}\\
&=\left(U*\left(\left(\tilde{V}^+ * \boldsymbol{X}^+\right)+\left(\tilde{V}^- * \boldsymbol{X}^-\right)\right)\right)_{r,s,t_1}\\
&=\sum_{i,j\in[d],k\in[2nc_0]}U_{i,j,k,t_1}\cdot\left(\left(\tilde{V}^+ * \boldsymbol{X}^+\right)+\left(\tilde{V}^- * \boldsymbol{X}^-\right)\right)_{r-i+1,s-j+1,k}\\
&=\sum_{i,j\in[d],k\in[2nc_0]}U_{i,j,k,t_1}\cdot\sum_{t_0\in[c_0]}\Big(\tilde{V}^+_{1,1,t_0,k}\cdot\boldsymbol{X}^+_{r-i+1,s-j+1,t_0}\\
&\quad+\tilde{V}^-_{1,1,t_0,k}\cdot\boldsymbol{X}^-_{r-i+1,s-j+1,t_0}\Big)\\
&=\sum_{t_0\in[c_0]in[d],k\in[2nc_0]}\left(U_{i,j,k,t_1}\cdot\tilde{V}^+_{1,1,t_0,k}\right)\cdot\boldsymbol{X}^+_{r-i+1,s-j+1,t_0}\\
&\quad+\sum_{t_0\in[c_0]}\sum_{i,j\in[d],k\in[2nc_0]}\left(U_{i,j,k,t_1}\cdot\tilde{V}^-_{1,1,t_0,k}\right)\cdot\boldsymbol{X}^-_{r-i+1,s-j+1,t_0}\\
&=\sum_{i,j\in[d],t_0\in[c_0]}\left(\sum_{k\in[2nc_0]}U_{i,j,k,t_1}\cdot\tilde{V}^+_{1,1,t_0,k}\right)\cdot\boldsymbol{X}^+_{r-i+1,s-j+1,t_0}\\
&\quad+\sum_{i,j\in[d],t_0\in[c_0]}\left(\sum_{k\in[2nc_0]}U_{i,j,k,t_1}\cdot\tilde{V}^-_{1,1,t_0,k}\right)\cdot\boldsymbol{X}^-_{r-i+1,s-j+1,t_0}.
\end{aligned}$$

Define $\boldsymbol{L}^{+}_{i,j,t_0,t_1} = \sum_{k\in[nc]} U_{i,j,k,t_1} \cdot \tilde{V}^{+}_{1,1,t_0,k}$ and, similarly, $\boldsymbol{L}^{-}_{i,j,t_0,t_1} = \sum_{k\in[nc]} U_{i,j,k,t_1} \cdot \tilde{V}^{-}_{1,1,t_0,k}$. Then,

$$\begin{aligned}
&\sum_{i,j\in[d],t_0\in[c_0]} \left(\sum_{k\in[nc_0]} U_{i,j,k,t_1} \cdot \tilde{V}^{+}_{1,1,t_0,k} \right) \cdot \boldsymbol{X}^{+}_{r-i+1,s-j+1,t_0} \\
&\quad + \sum_{i,j\in[d],t_0\in[c_0]} \left(\sum_{k\in[nc_0]} U_{i,j,k,t_1} \cdot \tilde{V}^{-}_{1,1,t_0,k} \right) \cdot \boldsymbol{X}^{-}_{r-i+1,s-j+1,t_0} \\
= &\sum_{i,j\in[d],t_0\in[c_0]} \boldsymbol{L}^{+}_{i,j,t_0,t_1} \cdot \boldsymbol{X}^{+}_{r-i+1,s-j+1,t_0} + \sum_{i,j\in[d],t_0\in[c_0]} \boldsymbol{L}^{-}_{i,j,t_0,t_1} \cdot \boldsymbol{X}^{-}_{r-i+1,s-j+1,t_0}.
\end{aligned}$$

We now show that, for each $t_0 \in [c_0]$, $K_{:,:,t_0,:}$ can be ε-approximated by $\boldsymbol{L}^{+}_{:,:,t_0,:}$ by suitably pruning $\tilde{V}^{+}$, i.e., by further zeroing entries of $\mathcal{S}$, and that such pruning corresponds to solving an instance of MRSS according to Theorem 5.3.2. The same reasoning applies to $\boldsymbol{K}^{-}$ and $\boldsymbol{L}^{-}$.

For each $t_0 \in [c_0]$, let

$$I^{(t_0)}_{+} = \{k \in \{(2t_0-2)n+1, \dots, (2t_0-1)n\} : \mathcal{S}_{1,1,t_0,k} = 1\}.$$

Observe that $I^{(t_0)}_{+}$ consists of the strictly positive entries of $\tilde{V}^{+}_{1,1,t_0,:}$.[4] Since the entries of $\boldsymbol{V}$ follow a standard normal distribution, each entry is positive with probability $1/2$. By a standard application of Chernoff bounds (Lemma I.3 in appendix I), we then have

$$\Pr\left[\left|I^{(t_0)}_{+}\right| > \frac{n}{3}\right] \geq 1 - \frac{\varepsilon}{4},$$

provided that the constant C in the bound on n is sufficiently large.

For each $k \in I^{(t_0)}_{+}$, up to reshaping the tensor as a one dimensional vector, $U_{:,:,k,:} \cdot \tilde{V}^{+}_{1,1,t_0,k}$ is an NSN vector (Definition 5.3.2) by Lemma I.5 (appendix J). Thus, for each $t_0 \in [c_0]$ and a sufficiently-large value of C, since the target filter K is such that $\|K_{:,:,t_0,:}\|_1 \leq 1$ and we have $n \geq Cd^{12}c_1^6 \log^3 \frac{d^2c_1c_0}{\varepsilon}$, then we can apply an amplified version of Theorem 5.3.2 (i.e., Corollary I.6 in appendix J with vectors of dimension d^2c_1) to show that, with probability $1 - \frac{\varepsilon}{4c_0}$ there exists a way to zero the entries indexed by $I^{(t_0)}_{+}$ of $\mathcal{S}$ (and thus $\tilde{V}^{+}_{1,1,t_0,:}$), so that the pruned version of $\boldsymbol{L}^{+}_{:,:,t_0,:} = \sum_{k\in[nc_0]} U_{:,:,k,:} \cdot \tilde{V}^{+}_{1,1,t_0,k}$ approximates $K_{:,:,t_0,:}$. In particular, there exists another binary mask $\hat{S}^{+} \in \{0,1\}^{\text{shape}\,\tilde{S}}$ such that $\hat{L}^{+}_{:,:,t_0,:} = \sum_{k\in[nc_0]} U_{:,:,k,:} \cdot \hat{V}^{+}_{1,1,t_0,k}$ approximates $K_{:,:,t_0,:}$, where $\hat{V}^{+} = \tilde{V}^{+} \odot \hat{S}^{+}$. An analogous argument carries on for a binary mask $\hat{S}^{-}$ and $-\hat{L}^{-}_{:,:,t_0,:}$.[5] More formally, let

$$\begin{aligned}
\mathcal{E}^{(\text{kernel})}_{t_0,+} &= \left\{ \exists \hat{S}^{+} \in \{0,1\}^{\text{shape}\,\tilde{S}} \;\; \|\hat{L}^{+}_{:,:,t_0,:} - K_{:,:,t_0,:}\|_\infty \leq \frac{\varepsilon}{2d^2c_1c_0} \right\}, \\
\mathcal{E}^{(\text{kernel})}_{t_0,-} &= \left\{ \exists \hat{S}^{-} \in \{0,1\}^{\text{shape}\,\tilde{S}} \;\; \|\hat{L}^{-}_{:,:,t_0,:} + K_{:,:,t_0,:}\|_\infty \leq \frac{\varepsilon}{2d^2c_1c_0} \right\}, \text{ and} \\
\mathcal{E}^{(\text{kernel})} &= \left(\bigcap_{t_0\in[c_0]} \mathcal{E}^{(\text{kernel})}_{t_0,+} \right) \bigcap \left(\bigcap_{t_0\in[c_0]} \mathcal{E}^{(\text{kernel})}_{t_0,-} \right).
\end{aligned}$$

[4] Notice that excluding zero entries implies conditioning on the event that the entry is not zero. However, such an event has zero probability and thus doesn't impact the analysis.

[5] The negative sign in front of $\hat{L}^{-}_{:,:,t_0,:}$ does not affect the random subset sum result as each entry is independently negative or positive with the same probability.

Then, by Corollary I.6,

$$\Pr\left[\mathcal{E}^{(\text{kernel})}_{t_0,+} \;\middle|\; |I^{(t_0)}_+| > \frac{n}{3}\right] \geq 1 - \frac{\varepsilon}{4c_0}, \text{ and}$$
$$\Pr\left[\mathcal{E}^{(\text{kernel})}_{t_0,-} \;\middle|\; |I^{(t_0)}_-| > \frac{n}{3}\right] \geq 1 - \frac{\varepsilon}{4c_0}.$$

By the union bound, we have the following:

$$\begin{aligned}
&\Pr\left[\mathcal{E}^{(\text{kernel})} \;\middle|\; |I^{(t_0)}_+|, |I^{(t_0)}_-| > \frac{n}{3}\right] \\
&= 1 - \Pr\left[\left(\bigcup_{t_0\in[c_0]} \overline{\mathcal{E}}^{(\text{kernel})}_{t_0,+}\right) \bigcup \left(\bigcup_{t_0\in[c_0]} \overline{\mathcal{E}}^{(\text{kernel})}_{t_0,-}\right) \;\middle|\; \left|I^{(t_0)}_+\right|, \left|I^{(t_0)}_-\right| > \frac{n}{3}\right] \\
&\geq 1 - \sum_{t_0\in[c_0]} \left[\Pr\left[\overline{\mathcal{E}}^{(\text{kernel})}_{t_0,+} \;\middle|\; \left|I^{(t_0)}_+\right|, \left|I^{(t_0)}_-\right| > \frac{n}{3}\right] + \Pr\left[\overline{\mathcal{E}}^{(\text{kernel})}_{t_0,-} \;\middle|\; \left|I^{(t_0)}_+\right|, \left|I^{(t_0)}_-\right| > \frac{n}{3}\right]\right] \\
&\geq 1 - 2\sum_{t_0\in[c_0]} \frac{\varepsilon}{4c_0} \\
&\geq 1 - \frac{\varepsilon}{2}.
\end{aligned}$$

Since $\Pr\left[\left|I^{(t_0)}_+\right|, \left|I^{(t_0)}_-\right| > \frac{n}{3}\right] \geq 1 - \frac{\varepsilon}{2}$, then we can remove the conditional event obtaining

$$\begin{aligned}
\Pr\left[\mathcal{E}^{(\text{kernel})}\right] &\geq \Pr\left[\mathcal{E}^{(\text{kernel})} \;\middle|\; \left|I^{(t_0)}_+\right|, \left|I^{(t_0)}_-\right| > \frac{n}{3}\right] \Pr\left[\left|I^{(t_0)}_+\right|, \left|I^{(t_0)}_-\right| > \frac{n}{3}\right] \\
&\geq \left(1 - \frac{\varepsilon}{2}\right)^2 \\
&\geq 1 - \varepsilon. \qquad \text{(J.52)}
\end{aligned}$$

To rewrite the latter in terms of the filter K and a mask S, we notice that pruning $L^+_{:,:,t_0,:}$ and $L^-_{:,:,t_0,:}$ separately, with two binary masks, is equivalent to say that there exists a single binary mask $\hat{S} \in \{0,1\}^{\text{shape}\,\tilde{S}}$ such that, $\hat{L}_{:,:,t_0,:}$ can be written as $\hat{L}_{:,:,t_0,:} = \sum_{k\in[nc_0]} U_{:,:,k,:} \cdot \hat{V}_{1,1,t_0,k}$, where $\hat{V} = \tilde{V} \odot \hat{S}$. Equation (J.52) implies that, with probability $1-\varepsilon$, such $\hat{S}$ exists and hence,

$$\|K - \hat{L}^+\|_\infty + \|K + \hat{L}^-\|_\infty \leq \frac{\varepsilon}{d^2 c_1 c_0}. \qquad \text{(J.53)}$$

Let $S = \tilde{S} \odot \hat{S}$: S is a $2n$-channel blocked basks. Furthermore, for such an S, notice that the following holds.

$$\begin{aligned}
&\sup_{X:\|X\|_\infty\leq M} \|K * X - N_0^{(S)}(X)\|_\infty \\
&= \sup_{X:\|X\|_\infty\leq M} \|K * X - U * \text{relu}\left((V \odot S) * X\right)\|_\infty \\
&= \sup_{X:\|X\|_\infty\leq M} \|K * X - U * \text{relu}\left((V \odot \tilde{S} \odot \hat{S}) * X\right)\|_\infty \\
&= \sup_{X:\|X\|_\infty\leq M} \|K * \left(\boldsymbol{X}^+ - \boldsymbol{X}^-\right) - U * \left((\hat{V}^+ * \boldsymbol{X}^+\right) + \left(\hat{V}^- * \boldsymbol{X}^-\right)\|_\infty,
\end{aligned}$$

where the latter holds by Lemma 5.4.4.[6] Then, by the distributive property of the convolution and the triangle inequality,

$$\begin{aligned}
&\sup_{X:\|X\|_\infty\le M} \|K * \left(\boldsymbol{X}^+ - \boldsymbol{X}^-\right) - U * \left((\hat{V}^+ * \boldsymbol{X}^+\right) + \left(\hat{V}^- \odot \hat{S} * \boldsymbol{X}^-\right)\|_\infty \\
= &\sup_{X:\|X\|_\infty\le M} \|K * \boldsymbol{X}^+ - U * \left(\hat{V}^+ * \boldsymbol{X}^+\right) - K * \boldsymbol{X}^- - U * \left(\hat{V}^- * \boldsymbol{X}^-\right)\|_\infty \\
\le &\sup_{X:\|X\|_\infty\le M} \|K * \boldsymbol{X}^+ - U * \left(\hat{V}^+ * \boldsymbol{X}^+\right)\|_\infty \\
&+ \sup_{X:\|X\|_\infty\le M} \|K * \boldsymbol{X}^- + U * \left(\hat{V}^- * \boldsymbol{X}^-\right)\|_\infty.
\end{aligned}$$

One can now apply the Tensor Convolution Inequality (Lemma I.7) and obtain

$$\begin{aligned}
&\sup_{X:\|X\|_\infty\le M} \|K * \boldsymbol{X}^+ - U * \left(\hat{V}^+ * \boldsymbol{X}^+\right)\|_\infty \\
&+ \sup_{X:\|X\|_\infty\le M} \|K * \boldsymbol{X}^- + U * \left(\hat{V}^- * \boldsymbol{X}^-\right)\|_\infty \\
\le &\sup_{X:\|X\|_\infty\le M} \|\boldsymbol{X}^+\|_\infty \cdot \|K - U * \hat{V}^+\|_1 \\
&+ \sup_{X:\|X\|_\infty\le M} \|\boldsymbol{X}^-\|_\infty \cdot \|K + U * \hat{V}^-\|_1 \\
= &\, M \cdot \|K - U * \hat{V}^+\|_1 + M \cdot \|K + U * \hat{V}^-\|_1.
\end{aligned}$$

Now, observing that the number of entries of the two tensors in the expression above is $d^2 c_1 c_0$, and using equation (J.53) (which holds with probability $1 - \varepsilon$), we get that

$$\begin{aligned}
&M \cdot \|K - U * \hat{V}^+\|_1 + M \cdot \|K - U * \hat{V}^-\|_1 \\
\le &\, d^2 c_1 c_0 \left(\|K - U * \hat{V}^+\|_\infty + \|K - U * \hat{V}^-\|_\infty\right) \\
\le &\, d^2 c_1 c_0 M \frac{\varepsilon}{d^2 c_1 c_0} \\
= &\, \varepsilon M.
\end{aligned}$$

proving the thesis. □

Proof of Theorem 5.3.1. In order to bound the error propagation across layers, we define the layers' outputs

$$\begin{aligned}
&\boldsymbol{X}^{(0)} = X, \\
&\boldsymbol{X}^{(i)} = \text{relu}\left(\boldsymbol{K}^{(i)} * \boldsymbol{X}^{(i-1)}\right) \quad \text{for } 1 \le i \le \ell.
\end{aligned} \tag{J.54}$$

Notice that $\boldsymbol{X}^{(\ell)}$ is the output of the target function, i.e., $f(X) = \boldsymbol{X}^{(\ell)}$.

For brevity's sake, given masks $\boldsymbol{S}^{(1)}, \ldots, \boldsymbol{S}^{(2\ell)}$, let us denote

$$\tilde{\boldsymbol{L}}^{(i)} = \boldsymbol{L}^{(i)} \odot \boldsymbol{S}^{(i)}.$$

[6] The presence of $\hat{S}$ does not influence the proof of Lemma 5.4.4.

Since the ReLU function is 1-Lipschitz, for all $\boldsymbol{X}^{(i-1)}$ it holds

$$\begin{aligned}&\|\text{relu}\left(\boldsymbol{K}^{(i)} * \boldsymbol{X}^{(i-1)}\right) - \text{relu}\left(\tilde{\boldsymbol{L}}^{(2i)} * \text{relu}\left(\tilde{\boldsymbol{L}}^{(2i-1)} * \boldsymbol{X}^{(i-1)}\right)\right)\|_\infty \\ &\leq \|\boldsymbol{K}^{(i)} * \boldsymbol{X}^{(i-1)} - \tilde{\boldsymbol{L}}^{(2i)} * \text{relu}\left(\tilde{\boldsymbol{L}}^{(2i-1)} * \boldsymbol{X}^{(i-1)}\right)\|_\infty. \end{aligned} \tag{J.55}$$

The key step of the proof is that, for each layer i, since $n_i \geq Cd^{12}c_i^6 \log^3 \frac{d^2 c_i c_{i-1}\ell}{\varepsilon}$ for a suitable constant C, we can apply Lemma 5.4.5 to get that, with probability at least $1 - \frac{\varepsilon}{2\ell}$, for all $\boldsymbol{X}^{(i-1)} \in \mathbb{R}^{D\times D\times c_0}$ it holds

$$\|\boldsymbol{K}^{(i)} * \boldsymbol{X}^{(i-1)} - \tilde{\boldsymbol{L}}^{(2i)} * \text{relu}\left(\tilde{\boldsymbol{L}}^{(2i-1)} * \boldsymbol{X}^{(i-1)}\right)\|_\infty < \frac{\varepsilon}{2\ell} \cdot \|\boldsymbol{X}^{(i-1)}\|_\infty. \tag{J.56}$$

Hence, combining equations (J.55) and (J.56) we get that, with probability at least $1 - \frac{\varepsilon}{2\ell}$, for all $\boldsymbol{X}^{(i-1)} \in \mathbb{R}^{D\times D\times c_0}$,

$$\begin{aligned}&\|\text{relu}\left(\boldsymbol{K}^{(i)} * \boldsymbol{X}^{(i-1)}\right) - \text{relu}\left(\tilde{\boldsymbol{L}}^{(2i)} * \text{relu}\left(\tilde{\boldsymbol{L}}^{(2i-1)} * \boldsymbol{X}^{(i-1)}\right)\right)\|_\infty \\ &< \frac{\varepsilon}{2\ell} \cdot \|\boldsymbol{X}^{(i-1)}\|_\infty. \end{aligned} \tag{J.57}$$

By a union bound, we get that equation (J.57) holds for all layers with probability at least $1 - \varepsilon$.

Analogously, we can define the pruned layers' outputs

$$\begin{aligned}\tilde{\boldsymbol{X}}^{(0)} &= X, \\ \tilde{\boldsymbol{X}}^{(i)} &= \text{relu}\left(\tilde{\boldsymbol{L}}^{(2i)} * \text{relu}\left(\tilde{\boldsymbol{L}}^{(2i-1)} * \tilde{\boldsymbol{X}}^{(i-1)}\right)\right) \quad \text{for } 1 \leq i \leq \ell. \end{aligned} \tag{J.58}$$

Notice that $\tilde{\boldsymbol{X}}^{(\ell)}$ is the output of the pruned network, i.e., $N_0^{(S^{(1)},\dots,S^{(2\ell)})}(X) = \tilde{\boldsymbol{X}}^{(\ell)}$.

By the same reasoning employed to derive equations (J.56) and (J.57) we have that, with probability $1 - \varepsilon$, the output of all pruned layers satisfies

$$\begin{aligned}&\|\text{relu}\left(\boldsymbol{K}^{(i)} * \tilde{\boldsymbol{X}}^{(i-1)}\right) - \text{relu}\left(\tilde{\boldsymbol{L}}^{(2i)} * \text{relu}\left(\tilde{\boldsymbol{L}}^{(2i-1)} * \tilde{\boldsymbol{X}}^{(i-1)}\right)\right)\|_\infty \\ &< \frac{\varepsilon}{2\ell} \cdot \|\tilde{\boldsymbol{X}}^{(i-1)}\|_\infty. \end{aligned} \tag{J.59}$$

Moreover, for each $1 \leq i \leq \ell - 1$, by the triangle inequality and by equation (J.59),

$$\begin{aligned}\|\tilde{\boldsymbol{X}}^{(i)}\|_\infty &= \|\tilde{\boldsymbol{X}}^{(i)} - \text{relu}\left(\boldsymbol{K}^{(i)} * \tilde{\boldsymbol{X}}^{(i-1)}\right) + \text{relu}\left(\boldsymbol{K}^{(i)} * \tilde{\boldsymbol{X}}^{(i-1)}\right)\|_\infty \\ &\leq \|\tilde{\boldsymbol{X}}^{(i)} - \text{relu}\left(\boldsymbol{K}^{(i)} * \tilde{\boldsymbol{X}}^{(i-1)}\right)\|_\infty + \|\text{relu}\left(\boldsymbol{K}^{(i)} * \tilde{\boldsymbol{X}}^{(i-1)}\right)\|_\infty \\ &\leq \frac{\varepsilon}{2\ell} \cdot \|\tilde{\boldsymbol{X}}^{(i-1)}\|_\infty + \|\text{relu}\left(\boldsymbol{K}^{(i)} * \tilde{\boldsymbol{X}}^{(i-i)}\right)\|_\infty. \end{aligned}$$

By the Lipschitz property of relu and Lemma I.7

$$\begin{aligned}
&\frac{\varepsilon}{2\ell} \cdot \|\tilde{\boldsymbol{X}}^{(i-1)}\|_\infty + \|\mathrm{relu}\left(\boldsymbol{K}^{(i)} * \tilde{\boldsymbol{X}}^{(i-i)}\right)\|_\infty \\
\leq\ &\frac{\varepsilon}{2\ell} \cdot \|\tilde{\boldsymbol{X}}^{(i-1)}\|_\infty + \|\boldsymbol{K}^{(i)} * \tilde{\boldsymbol{X}}^{(i-i)}\|_\infty \\
\leq\ &\frac{\varepsilon}{2\ell} \cdot \|\tilde{\boldsymbol{X}}^{(i-1)}\|_\infty + \|\boldsymbol{K}^{(i)}\|_1 \|\tilde{\boldsymbol{X}}^{(i-i)}\|_\infty \\
=\ &\|\tilde{\boldsymbol{X}}^{(i-1)}\|_\infty \left(1 + \frac{\varepsilon}{2\ell}\right).
\end{aligned}$$

By unrolling the recurrence, we get that, with probability $1 - \varepsilon$,

$$\|\tilde{\boldsymbol{X}}^{(i)}\|_\infty \leq \|\tilde{\boldsymbol{X}}^{(0)}\|_\infty \left(1 + \frac{\varepsilon}{2\ell}\right)^i. \tag{J.60}$$

Thus, combining equations (J.59) and (J.60), with probability $1 - \varepsilon$ we get that, for each $i \in [\ell]$,

$$\begin{aligned}
&\|\boldsymbol{K}^{(i)} * \tilde{\boldsymbol{X}}^{(i-1)} - \tilde{\boldsymbol{L}}^{(2i)} * \mathrm{relu}\left(\tilde{\boldsymbol{L}}^{(2i-1)} * \tilde{\boldsymbol{X}}^{(i-1)}\right)\|_\infty \\
<\ &\frac{\varepsilon}{2\ell} \cdot \left(1 + \frac{\varepsilon}{2\ell}\right)^{i-1} \|\tilde{\boldsymbol{X}}^{(0)}\|_\infty.
\end{aligned} \tag{J.61}$$

We then see that with probability $1 - \varepsilon$, for $1 \leq i \leq \ell$ and all $X \in [-1, 1]^{D \times D \times c_0}$, by equations (J.54) and (J.58), and by the triangle inequality,

$$\begin{aligned}
&\|\boldsymbol{X}^{(\ell)} - \tilde{\boldsymbol{X}}^{(\ell)}\|_\infty \\
=\ &\|\mathrm{relu}\left(\boldsymbol{K}^{(\ell)} * \boldsymbol{X}^{(\ell-1)}\right) - \mathrm{relu}\left(\tilde{\boldsymbol{L}}^{(2\ell)} * \mathrm{relu}\left(\tilde{\boldsymbol{L}}^{(2\ell-1)} * \tilde{\boldsymbol{X}}^{(\ell-1)}\right)\right)\|_\infty \\
\leq\ &\|\mathrm{relu}\left(\boldsymbol{K}^{(\ell)} * \boldsymbol{X}^{(\ell-1)}\right) - \mathrm{relu}\left(\boldsymbol{K}^{(\ell)} * \tilde{\boldsymbol{X}}^{(\ell-1)}\right)\|_\infty \\
&+ \|\mathrm{relu}\left(\boldsymbol{K}^{(\ell)} * \tilde{\boldsymbol{X}}^{(\ell-1)}\right) - \mathrm{relu}\left(\tilde{\boldsymbol{L}}^{(2\ell)} * \mathrm{relu}\left(\tilde{\boldsymbol{L}}^{(2\ell-1)} * \tilde{\boldsymbol{X}}^{(\ell-1)}\right)\right)\|_\infty.
\end{aligned}$$

Again by the 1-Lipschitz property of the ReLU activation function, and by the distributive property of the convolution operation,

$$\begin{aligned}
&\|\mathrm{relu}\left(\boldsymbol{K}^{(\ell)} * \boldsymbol{X}^{(\ell-1)}\right) - \mathrm{relu}\left(\boldsymbol{K}^{(\ell)} * \tilde{\boldsymbol{X}}^{(\ell-1)}\right)\|_\infty \\
&+ \|\mathrm{relu}\left(\boldsymbol{K}^{(\ell)} * \tilde{\boldsymbol{X}}^{(\ell-1)}\right) - \mathrm{relu}\left(\tilde{\boldsymbol{L}}^{(2\ell)} * \mathrm{relu}\left(\tilde{\boldsymbol{L}}^{(2\ell-1)} * \tilde{\boldsymbol{X}}^{(\ell-1)}\right)\right)\|_\infty \\
\leq\ &\|\boldsymbol{K}^{(\ell)} * \boldsymbol{X}^{(\ell-1)} - \boldsymbol{K}^{(\ell)} * \tilde{\boldsymbol{X}}^{(\ell-1)}\|_\infty \\
&+ \|\boldsymbol{K}^{(\ell)} * \tilde{\boldsymbol{X}}^{(\ell-1)} - \tilde{\boldsymbol{L}}^{(2\ell)} * \mathrm{relu}\left(\tilde{\boldsymbol{L}}^{(2\ell-1)} * \tilde{\boldsymbol{X}}^{(\ell-1)}\right)\|_\infty \\
=\ &\|\boldsymbol{K}^{(\ell)} * \left(\boldsymbol{X}^{(\ell-1)} - \tilde{\boldsymbol{X}}^{(\ell-1)}\right)\|_\infty \\
&+ \|\boldsymbol{K}^{(\ell)} * \tilde{\boldsymbol{X}}^{(\ell-1)} - \tilde{\boldsymbol{L}}^{(2\ell)} * \mathrm{relu}\left(\tilde{\boldsymbol{L}}^{(2\ell-1)} * \tilde{\boldsymbol{X}}^{(\ell-1)}\right)\|_\infty.
\end{aligned}$$

Lemma I.7 and the hypothesis $\|\boldsymbol{K}^{(\ell)}\|_1 \leq 1$ imply that

$$\begin{aligned}
&\|\boldsymbol{K}^{(\ell)} * \left(\boldsymbol{X}^{(\ell-1)} - \tilde{\boldsymbol{X}}^{(\ell-1)}\right)\|_\infty \\
&\quad + \|\boldsymbol{K}^{(\ell)} * \tilde{\boldsymbol{X}}^{(\ell-1)} - \tilde{\boldsymbol{L}}^{(2\ell)} * \mathrm{relu}\left(\tilde{\boldsymbol{L}}^{(2\ell-1)} * \tilde{\boldsymbol{X}}^{(\ell-1)}\right)\|_\infty \\
&\leq \|\boldsymbol{K}^{(\ell)}\|_1 \cdot \|\left(\boldsymbol{X}^{(\ell-1)} - \tilde{\boldsymbol{X}}^{(\ell-1)}\right)\|_\infty \\
&\quad + \|\boldsymbol{K}^{(\ell)} * \tilde{\boldsymbol{X}}^{(\ell-1)} - \tilde{\boldsymbol{L}}^{(2\ell)} * \mathrm{relu}\left(\tilde{\boldsymbol{L}}^{(2\ell-1)} * \tilde{\boldsymbol{X}}^{(\ell-1)}\right)\|_\infty \\
&\leq \|\left(\boldsymbol{X}^{(\ell-1)} - \tilde{\boldsymbol{X}}^{(\ell-1)}\right)\|_\infty \\
&\quad + \|\boldsymbol{K}^{(\ell)} * \tilde{\boldsymbol{X}}^{(\ell-1)} - \tilde{\boldsymbol{L}}^{(2\ell)} * \mathrm{relu}\left(\tilde{\boldsymbol{L}}^{(2\ell-1)} * \tilde{\boldsymbol{X}}^{(\ell-1)}\right)\|_\infty.
\end{aligned}$$

Now, we first apply equation (J.61) and then we unroll the recurrence for all layers (as, with probability $1-\varepsilon$, equation (J.61) holds for all layers), obtaining

$$\begin{aligned}
&\|\left(\boldsymbol{X}^{(\ell-1)} - \tilde{\boldsymbol{X}}^{(\ell-1)}\right)\|_\infty \\
&\quad + \|\boldsymbol{K}^{(\ell)} * \tilde{\boldsymbol{X}}^{(\ell-1)} - \tilde{\boldsymbol{L}}^{(2\ell)} * \mathrm{relu}\left(\tilde{\boldsymbol{L}}^{(2\ell-1)} * \tilde{\boldsymbol{X}}^{(\ell-1)}\right)\|_\infty \\
&\leq \|\boldsymbol{X}^{(\ell-1)} - \tilde{\boldsymbol{X}}^{(\ell-1)}\|_\infty + \frac{\varepsilon}{2\ell} \cdot \left(1 + \frac{\varepsilon}{2\ell}\right)^{\ell-1} \\
&\leq \sum_{j=1}^{\ell} \frac{\varepsilon}{2\ell} \cdot \left(1 + \frac{\varepsilon}{2\ell}\right)^{j-1}.
\end{aligned}$$

By summing the geometric series and observing that $\varepsilon < 1$, we conclude that

$$\begin{aligned}
\sum_{j=1}^{\ell} \frac{\varepsilon}{2\ell} \cdot \left(1 + \frac{\varepsilon}{2\ell}\right)^{j-1} &= \left(1 + \frac{\varepsilon}{2\ell}\right)^{\ell} - 1 \\
&\leq e^{\frac{\varepsilon}{2}} - 1 \\
&\leq \varepsilon.
\end{aligned}$$

Hence, with probability $1-\varepsilon$, for all $X \in [-1,1]^{D \times D \times c_0}$, for all $\ell \in [c]$ it holds that

$$\|\boldsymbol{X}^{(\ell)} - \tilde{\boldsymbol{X}}^{(\ell)}\|_\infty \leq \varepsilon,$$

yielding the thesis. □

Élagage des structures aléatoires

Arthur Carvalho Walraven da Cunha

Résumé

La *Strong Lottery Ticket Hypothesis (SLTH)* stipule que les réseaux de neurones contiennent, lors de l'initialisation aléatoire, des sous-réseaux qui fonctionnent bien sans aucun entraînement. Le réseau aléatoire doit cependant être sur-paramétré : avoir plus de paramètres qu'il n'en aurait besoin. La SLTH a d'abord été prouvée pour les réseaux entièrement connectés et suppose une sur-paramétrisation polynomiale. Puis, cela a été amélioré pour ne nécessiter qu'un surplus logarithmique, ce qui est essentiellement optimal. Ce fort résultat a tiré parti d'un beau théorème sur le *Subset Sum Problem (SSP)*. Il considère une version aléatoire du SSP dans laquelle on cherche à approximer une valeur cible en sommant des sous-ensembles d'un échantillon aléatoire donné. Le théorème affirme que garantir l'existence d'une solution avec une haute probabilité ne nécessite qu'une taille d'échantillon logarithmique par rapport à la précision des approximations. Nous présentons une preuve plus simple et plus directe pour ce résultat. Ensuite, en tirant parti du théorème sur le SSP, nous étendons le SLTH aux *Convolutional Neural Networks (CNNs)* : nous montrons que les CNN aléatoires contiennent des sous-CNN clairsemés qui n'ont pas besoin d'entraînement pour obtenir de bonnes performances. Nous avons également obtenu le résultat en supposant une sur-paramétrisation logarithmique. Bien que le surplus imposé par le SLTH puisse être compensé par la rareté des sous-réseaux obtenus, exploiter la rareté en pratique est très difficile si elle n'est pas structurée. Étendre les résultats sur le SLTH pour produire des sous-réseaux structurés nécessiterait une version multidimensionnelle du théorème sur le SSP. Nous prouvons la véracité d'une telle version et nous l'utilisons pour montrer que le SLTH est toujours valable pour les CNN si nous exigeons que les sous-réseaux soient structurés. Enfin, nous proposons une application des idées de cette thèse à la conception de circuits : nous exploitons l'aléatoire inhérent aux spécifications des composants électroniques intégrés pour obtenir des composants programmables hautement précis à partir de composants statiques de faible précision.

Mots-clés : Réseau de neurones, Algorithmes des graphes, Compression de modèles, Élagage

Abstract

The *Strong Lottery Ticket Hypothesis (SLTH)* states that neural networks contain, at random initialisation, sub-networks that perform well without any training. The random network needs, however, to be over-parameterized: to have more parameters than it would otherwise need. The SLTH was first proved for fully-connected networks and assumed polynomial over-parameterization. Soon after, this was improved to only require a logarithmic overhead, which is essentially optimal. This strong result leveraged a theorem on the *Subset Sum Problem (SSP)*. It considers a randomised version of the SSP in which one seeks to approximate a target value by summing subsets of a given random sample. The theorem asserts that ensuring the existence of a solution with high probability only requires a logarithmic sample size relative to the precision of the approximations. We present a simpler, more direct proof for this result. Then, leveraging the theorem on the SSP, we extend the SLTH to *Convolutional Neural Networks (CNNs)*: we show that random CNNs contain sparse sub-CNNs that do not require training to achieve good performance. We also obtained the result assuming a logarithmic over-parameterization. Even though the overhead imposed by the SLTH could be offset by the sparsity of the sub-networks obtained, exploiting sparsity in practice is very difficult if it is not structured. Extending the results on the SLTH to produce structured sub-networks would require a multidimensional version of the theorem on SSP. We prove such a version and use it to show that the SLTH still holds for CNNs if we require the sub-networks to be structured. Finally, we propose an application of the ideas in this thesis to the design of circuits: We harness the inherent randomness in the specs of integrated electronic components to obtain highly accurate programmable components from low-precision static ones.

Keywords: Neural network, Graph algorithms, Model compression, Pruning