On Random Subset Sum and some applications

Emanuele Natale

Université Côte d'Azur

Habilitation à Diriger des Recherches

préparée au Laboratoire d'Informatique, Signaux et Systèmes de
Sophia Antipolis, UCA, CNRS et Inria

Spécialité : **Informatique**

On Random Subset Sum and some applications

par

Emanuele Natale

Préalablement rapportée par :

Prof. David PELEG, Weizmann Institute of Science, Israel
Prof. Leszek GASIENIEK, Université de Liverpool, Royaume Uni
Prof. Christian SCHEIDELER, Université de Paderborn, Allemagne

Soutenue le 20 mars 2024 devant la jury composé de :

DR INRIA Giovanni NEGLIA (président), Université Côte d'Azur, France
DR INRIA Laurent VIENNOT, Université de Paris, France
Prof. Pierluigi CRESCENZI, Gran Sasso Science Institute, Italie
Prof. David PELEG, Weizmann Institute of Science, Israel
Prof. Leszek GASIENIEK, Université de Liverpool, Royaume Uni
Prof. Christian SCHEIDELER, Université de Paderborn, Allemagne

Contents

Chapter 1

10 Years of Research Later

Science advances one funeral at a time. - Max Planck

This HDR thesis focuses on the Random Subset Sum problem (RSSP) and its applications to Artificial Neural Networks (ANNs).

The HDR thesis should provide a general perspective on the researcher that is applying for the HDR title. Most scientists have a main research interest and it is thus clear what the main content of the thesis is going to be about. That's not my case, and the purpose of this first chapter is to delay our treatment of the RSSP until it can be framed in the right perspective within my application for the HDR. On the one hand, I considered surveying two different topics before resolving to the RSSP, but after some initial efforts, it simply felt that the result would not capture enough of the essence of my research over the last few years. In particular, I recently published the survey [BCN20a], in which my coauthors and I summarize the theoretical area in which I have been most active; that work is too recent to be expanded upon, and repeating most of its content felt of little service to the research community. On the other hand, trying to talk about everything in a more lightweight fashion was an option that I never really considered, as the result would have appeared too heterogeneous. I then set for the present compromise of presenting the main results in one of the research areas I have been contributing to, while providing a quick general overview of my scientific activity in the present chapter.

1.0.1 Thesis organization

The present chapter provides an informal and personal overview of my research over the last 10 years. In Section 1.6 I discuss my experience as a

PhD student supervisor. Chapters 2, 3 and 4 are technical and focus on the Random Subset Sum Problem and its connection with the so-called Lottery Ticket Hypothesis in artificial neural networks. Chapter 5 informally discusses some possible future research directions.

1.1 Alpha and beta releases: PhD years

Ten years ago, in 2013, I started my Ph.D. in Computer Science at Sapienza University of Rome. Five years before, in 2008, I had started studying Mathematics at the University of Rome Tor Vergata. One thing that become rapidly clear was that I would not become a professional mathematician in a classical sense, and I'd rather focus on other promising endeavors.

One surely was to contribute to what could be achieved using a computer. The other endeavor was to contribute to our understanding of life sciences, if not to the most mysterious byproduct of evolution: the human brain. Perhaps, in pursuing the first goal, I reasoned, I may also acquire useful tools to later contribute to the second one. The computer was, after all, a *thinking machine*. Or, if the reader would at this point find such a claim too cheap, I can quote:

> *If you will tell me precisely what it is that a machine cannot do, then I can always make a machine which will do just that.* - John von Neumann [JB03]

So it happened that I started my journey to become a theoretical computer scientist. Shortly after starting my Master's studies at the same university, in 2011, I switched to the Computer Science MSc degree. I was then very fortunate to be involved in the research group of Prof. Andrea Clementi. I had told him that, as Master's thesis project, I was looking for "a problem intersecting machine learning, distributed computing and network analysis". He then proposed to tackle the problem of community detection in temporal random networks. We later published our joint work in [CDIG+15].

Two years later, in 2013, I found myself with the important problem of choosing a Ph.D. topic. Distributed Computing was a possibility that would fit nicely with personal circumstances, and the hope that such a field could offer insights for progressing on the many open problems within the field of Complex Systems and Neuroscience was appealing. I then started my Ph.D. in Computer Science at Sapienza University of Rome.

During the aforementioned Master thesis project, I started analyzing some simple interaction rules among the agents of a system, that will be discussed next, in Section 1.2.

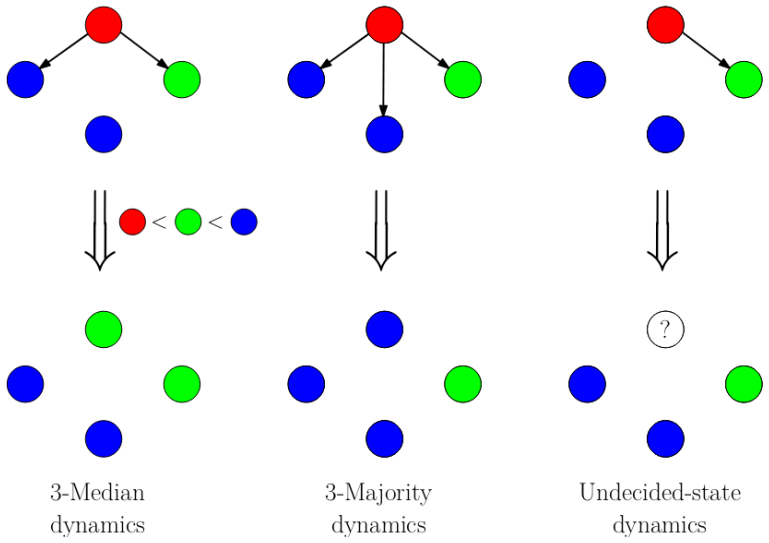

Figure 1.1: Examples of different *computational dynamics* that I investigated (see Section 1.2).

1.2 v1.0: Theoretical Computer Science and Multi-agent systems

During my Ph.D., which I obtained in 2017, I was fortunate enough to *discover* a research direction that we could later name *Computational Dynamics*, i.e., simple distributed probabilistic algorithms which allow multi-agent systems to solve global coordination tasks. Such line of works, that I summarized in my PhD thesis [Nat17] and, later, in a more up-to-date joint survey for ACM SIGACT News [BCN20a], concerned classes of algorithms that had been studied extensively from the perspective of computability theory. However, due to the lack of mathematical tools to rigorously model the behavior of these systems in the short term, efforts to explore these dynamics algorithmically had started to succeed only recently. My main contributions in this area had been on the fundamental distributed-computing problems of Consensus in some models of stochastic communication, under different flavors, which I briefly summarize in the coming Section 1.2.1.

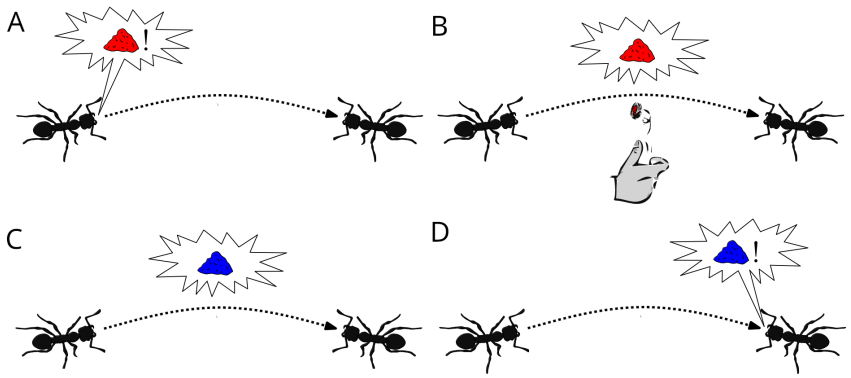

Figure 1.2: Illustration of noisy communication: A) The sender sends a given message m; B-C) m is changed to some message m' according to a probability distribution p_m which depends on m; D) the receiver receives the message m'.

1.2.1 Consensus Problems under Stochastic Communication

For the most part, I focused on the uniform PULL model in which every agent can only observe another agent sampled uniformly at random at each time step [DGH$^+$87]. The communication pattern is thus *stochastic* in the sense that agents have no control over with whom they communicate. This is typical of many real-world scenarios in biology, where multi-agent systems such as swarms of birds or insects can only get information from close neighbors, but the identity of the latter change unpredictably in a short time. Such *stochastic communication* property should not be confused with *noisy communication*, i.e. the fact that, when an agent u tries to convey some information to another agent v, the information that v receives can be different from what u originally intended to communicate (Figure 1.2).

The latter is the classical setting considered in Information Theory and has also played an important part in the work that I will mention in this section.

In general, we should think about the multi-agent system as a graph (that most of the time is *complete*), whose nodes/agents hold an *opinion* that can be represented with a color (see e.g. Figure 1.1).

Among the most fundamental problem in Distributed Computing and in the theory of Multi-agent Systems there's certainly the Consensus Problem, i.e. the problem of making agents agree on a value among a set of possible options.

How interesting could the Consensus Problem be? I will not argue about its importance for the Theory of Distributed Algorithms in general. In fact, I myself cannot offer more insight than a reader might get after reading for a few hours a standard reference [Lyn96]. As I mentioned above, in Section 1.1, my interest for the field has always been skewed toward aspects that could be of relevance for a biologist and, in that respect, I have been in rather a hurry to answer questions such as the following, which I hope will be captivating enough for most readers:

> Consider n agents, each one initially having a color out of a set of k possible ones. At each step, each agent u looks at the colors of two other agents v_1 and v_2 chosen independently and uniformly at random and, if v_1 and v_2 have the same color c, u's color becomes c as well.

As surprising as it could appear, back in 2013 the above question was both unanswered and interesting for the research community of distributed and parallel algorithms. So it was that I have been working on many variants of the Consensus Problem, which I list in the following.

In [BCN+16] I have been studying the Stable Consensus Problem, in which consensus should be maintained even when an adversary is corrupting the opinion of a limited number of agents in the system at each time step.

In [FN19], [BNFK18] and [dCN22], I have been studying the Noisy Consensus Problem, in which the communication is noisy (in the *noisy-communication* sense we just described). Notably, [BNFK18] appeared in PLOS Computational Biology, because of its interest for theoretical biologists, in particular concerning the study of the collective behaviors of biological systems. The latter work was thus a meaningful achievement to me as it represented a success with respect to my personal goal of contributing to the field of biology.

In the works [BCN+15, BCN+17b, CNNS18], I have been studying the Majority Consensus problem, in which we wish for the system to converge to the opinion which is initially held by a relative majority of the agents. The main object of investigation in the latter case are known as *majority dynamics*.

In [BCE+17], while the main object under investigation are still *majority dynamics*, I have been studying them in relation to the classical (Valid) Consensus Problem, in which the only requirement is that agents come to agree with a value that was initially present in the starting configuration. More specifically, the question being *what if we start with essentially every agent holding a different opinion*? In [BCE+17] we offer one of the first

rigorous ways to directly relate the behavior the 3-Majority and 2-Choices Dynamics, two of the most popular processes studied in the area.

Another joint work that is directly related to both the Valid and Noisy Consensus Problem has been [CGN+20] in which, in a precise sense, the Consensus Problem is shown to reduce to the Broadcast Problem, where an exponential gap between the two problems is shown when communication is affected by noise.

As mentioned above, [BCN20a] is a survey that I co-authored and which collects most of the relevant results (with the inevitable subjectivity of any relatively short survey). This section is thus kept as short as possible, and the reader interested in Computational Dynamics for the Consensus Problem is deferred to it. Related, older references that are still useful in many respects are [Sha07] and [MT17].

Next, in Section 1.2.2, I write about other works In which I tackled different problems other than variants of the Consensus one. However, the core mathematical techniques remain close to those employed in the above works. Further below, in Section 1.2.3, I dedicate a few lines to sharing my general opinion on the whole research area.

1.2.2 Opinion Dynamics and Community Detection

Back in Section 1.1, I mentioned how my journey as a researcher started in 2013 with my MSc thesis work [CDIG+15]. I have subsequently continued investigating the problem which goes under the names of Distributed Clustering or Distributed Community Detection, with the most beautiful result most likely being the very simple community detection algorithm for the stochastic block model in [BCN+20b] (Figure 1.3).

I provided an adaptation of [BCN+20b] in my joint work [BCM+18], where a variant of the clustering algorithm succeeds in performing community detection in the Population Protocol model, in which agents act asynchronously and there is no consistent notion of *global time*. I should remark here that the fact that the community detection problem becomes much harder in the Population Protocol model should not suggest to the inexperienced reader that problems are generally harder to solve in the asynchronous model. That is, in fact, quite not the case for the analysis of many computational dynamics such as those mentioned in Section 1.2.1: the behavior of a dynamics in the Population Protocol model gives rise to a much simpler Markov chain, as it involves a single interaction per transition.

Since 2013, while working on the conference version of [CIG+13], it was clear from simulations that the same result that our algorithm was achiev-

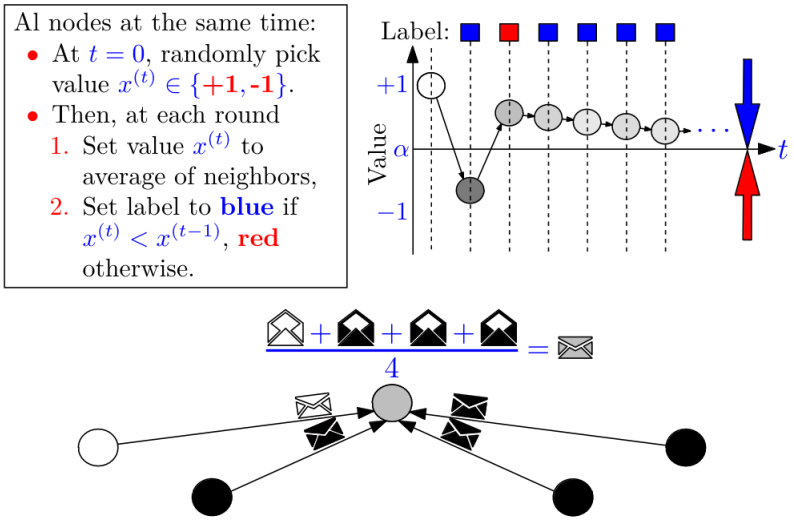

Figure 1.3: The Averaging Dynamics (see Section 1.2.2). On the top left, is the pseudocode of the process. On the bottom, illustration of one step of the dynamics. On the top right, is the representation of the evolution of the process: after an initial instability of the value held by a node, the value either decreases or increases stably, thus providing a common signal for identifying the node communities.

ing, could have been achieved through much simpler rules. Such rules were, and still are, extensively studied under the name of Label Propagation Algorithms, but almost all work on them is empirical. A good overview of some literature is provided by [Cru19]. Luckily, in 2018, a new technique to analyze one such dynamic - the 2-Choices Dynamics - had been found [CRRS17], allowing me and my collaborators to obtain the kind of rigorous result that I was striving for in 2013. The latter result appeared in [CNS19] and a related one that we obtained around the same time is [CNNS18].

1.2.3 So what?

The purpose of this chapter is to provide a bird's eye view of my past research, and given that the above works are not the focus of the present one, I will only share here my informed opinion for a researcher interested in knowing more about it.

In one way or another, all the above works consisted of a rigorous probabilistic analysis of discrete stochastic processes. The necessary mathematical background for understanding such analyses does not go too far beyond a solid background in the analysis of probabilistic algorithms, such that the one provided in [MU17]. The core techniques are a good grasp of concentration inequalities [CL06], probabilistic coupling [MU17, Ch. 12] and martingale techniques [Len17].

Unfortunately, most of the analyses one comes across in the literature turns out to be quite ad-hoc and, contrary to my hopes when I initially started to work in this area at the beginning of my Ph.D., the progress towards more general tools has been very limited. While on the practical side, there are surely many low-hanging results that the community would consider interesting and could thus be published in prestigious venues, I have to express my doubts regarding the possibility that, many years from now, such results that keep accumulating will crystallize into a more general understanding.

1.2.4 Other more-or-less related problems

As I have been recounting in previous sections, until recently the main focus of my research has been the study of computational dynamics. Along the way, however, I investigated many other related problems, among which the following have been particularly interesting.

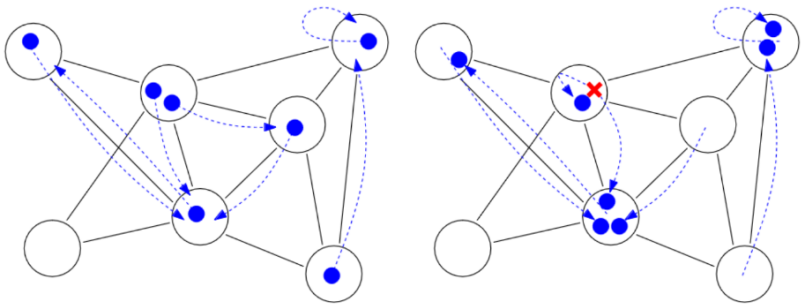

Figure 1.4: An illustration of one step of 7 tokens performing a random walks on a 7-node network in the GOSSIP Model. Each token selects its next destination independently and uniformly at random among the neighbors of its current node. Since in the GOSSIP Model, each node can communicate with only one neighbor at each step (see Section 1.2.4.1), if two or more tokens situated on the same node want to move to different neighbors, then they have to take turns moving.

1.2.4.1 Random walks in the GOSSIP Model

The analysis of random walks in networks subject to congestion constraints, such as the GOSSIP Model in which each node can communicate with at most one neighbor at each step, has been the subject of my joint work [BCN+17a]. The motivating scenario of random walks in the GOSSIP Model was actually proposed two years before, in my previous joint work [BCN+15] (see Fig. 1.4).

1.2.4.2 Clock synchronization in stochastic environments

Back in 2015, while visiting the IRIF Lab in Paris, I met for the first time Amos Korman. That was the beginning of one of the most important collaborations I carried on in the following years, and it was marked right away by a specific problem, the *Zealot Consensus Problem*. In the simplest form of the Zealot Consensus Problem, the system aims at reaching consensus *when one agent in the system is not going to change opinion*. We tackled the problem in the setting of a discrete-time multi-agent system in which agents interact stochastically in parallel (i.e. in the PULL model), and they can communicate only a few bits per interaction. One year later, in 2016, we started conceiving a solution that, after an additional year of refinement, resulted in the publication [BKN18]. In fact, the problem can be *reduced*,

in a classical sense, to that of synchronizing a clock; the intuition of such reduction is sketched in Figure 1.5.

In [BKN18], we manage to synchronize a clock of arbitrary cycle length using just 3 bits per interaction. Of course, the problem of further reducing the message length to 2 or even 1 bit was quite appealing. Later, in 2019, I started working with a student on a solution to the problem using only two bits. I successively stopped contributing to the project, but the student and other collaborators were able to successfully carry it on, publishing the 1-bit improvement in [BGS21][1]. The interesting fact that should be noted here is that, in fact, since the very few days after Amos had originally told me the problem, we came out with a 1-bit algorithm that was intuitively and empirically correct, but a rigorous analysis appeared beyond the grasp of current techniques. Recently, Amos Korman and a student of his managed to analyze a variant of such a natural algorithm in [KV22].

1.2.4.3 Fully-distributed Physarum dynamics

Physarum polycephalum is a *multinucleate coenocyte* which made headlines because, despite lacking a nervous system, behaves in a way that allows it to solve, *sensu lato*, instances of the Shortest Path Problem (see Fig. 1.6).

Its popularity spread among theoretical computer scientists when a discrete version of the aforementioned behavior started to be investigated. The interested reader is deferred to [Bon20] and references therein.

An important aspect that appeared to be lacking in such algorithmic models of the behavior of Physarum was that, in practice, the behavior of the organism is merely local: what the organism is doing in some spacial point is not influenced by points that are far away. Without getting into the details, let it be said that the algorithmic models tacitly assumed that the *solution* that Physarum was computing could leverage some kind of global information. In [BBN18], we address such a problem by providing a fully-local version of the famous algorithmic model, which leverages the nice theory bridging random walks and electric potentials.

[1] I was quite flattered by the acknowledgement the authors wrote in [BGS21]: "We are deeply indebted to Emanuele Natale for introducing the problem to us, for helping us to devise the binary clock protocol, for pointing out related work, and for helpful discussions through the course of this project."

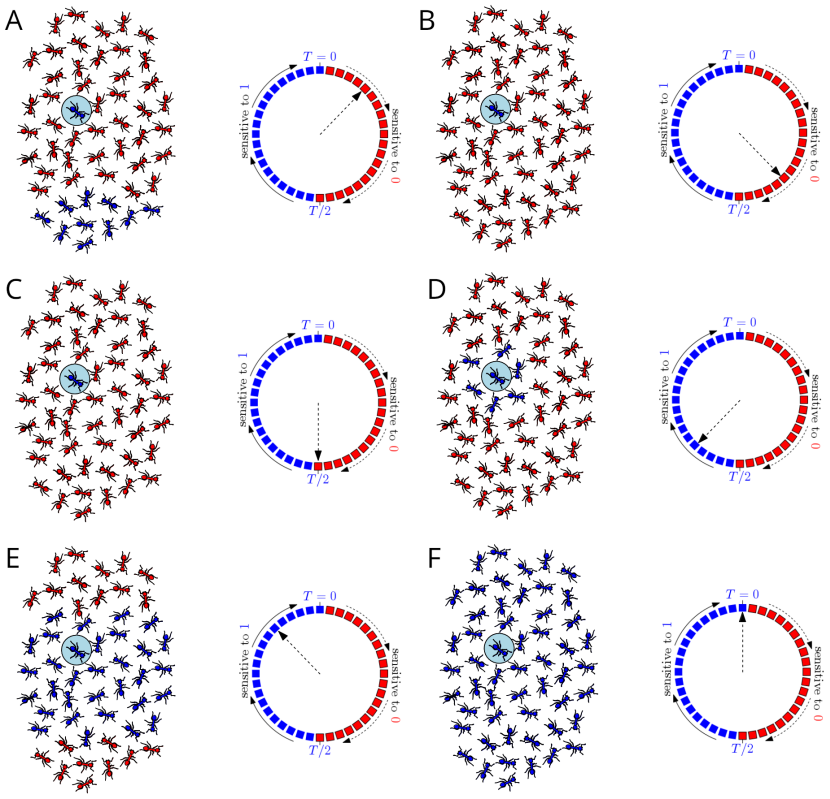

Figure 1.5: Illustration of how the binary Zealot Consensus Problem can be solved under the assumption that agents share a clock modulo $\mathcal{O}(\log n)$ where n is the number of agents. The shared clock is divided into two phases: one in which agents only adopt the first opinion if they communicate with an agent holding it, and a second one in which they only adopt the second opinion if they see it. In subfigures A-C, the agents are in the red phase, hence the red opinion spreads in the system and is eventually supported by all agents except the zealot agent who always hold the blue opinion. Successively, in subfigures D-F, the clock is in the blue phase and all agents copy the blue opinion if they see agents supporting it. Eventually, in subfigure F, all agents support the blue opinion (including the zealot agent), and when the clock enters again the red phase there is no red agent that can start propagating again the red opinion.

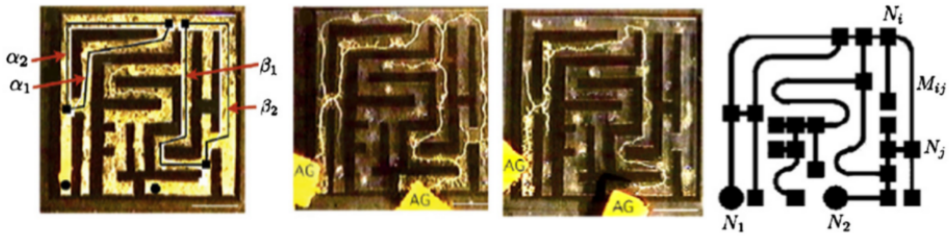

Figure 1.6: Illustration of the plasmodial maze-solving process (see Section 1.2.4.3), from left to right: (1) Physarum polycephalum is initially spread all over the maze. (2) Nutrients are placed in locations marked with AG; Physarum starts concentrating in a vascular-like system according to the model $\dot{x}_e = |q_e| - x_e$ where e a segment of the maze (an *edge*), x_e is the density of Physarum present in that segment, and q_e is the flow of nutrients being transported by Physarum along that segment. (c) The final state that Physarum reaches, in which the organism is concentrated on the shortest path between the two food sources. (d) Network representation of the maze, where the source node N_1 and the sink node N_2 are indicated by solid circles and other nodes are shown by solid squares.

1.2.4.4 Sampling random expander graphs and load balancing in distributed networks

In [BCN+19], we analyzed simple distributed processes for creating random expander graphs. The more general result is about sapling a random expander subgraph from a sufficiently dense one. If we consider the special case of a complete graph as the starting topology, we can think about such special case as n agents that wants to construct an expander graph in a decentralized manner. The underlying process is as simple as *asking a random neighbor to connect in the expander, and accepting the first k requests that you receive*. The analysis presented [BCN+19] is very technical and leverages the interesting technique of *encoding arguments*, a pedagogical exposition of which is provided in [MMR17].

In subsequent work, with some colleagues, we refined the analysis of [BCN+19] and obtained a result that appeared particularly relevant in the context of load balancing in distributed networks, which we published in [CNZ21].

1.2.4.5 Levy walks and the optimal foraging hypothesis

After my work on the Physarum dynamics that I described in Section 1.2.4.3, I started to become interested in the more general question of *how animals move*. There is a myriad of patterns that animal species follow, for a variety of reasons. These reasons, which can range from the optimal foraging to the fastest way to reach a nest site, are the main motivations why researchers are interested in animal movement models.

Lévy walks are one such pattern that has attracted a lot of attention in the biological and physical communities in the last twenty years, as they are optimal search strategies. The investigation of this type of random process dates back to work by Paul Lévy in 1937 [Lev37], from which the process gets its name. In the bi-dimensional case, we can think of a Lévy walk as follows: an agent repeatedly chooses a direction uniformly at random and then moves in that direction for a distance that is drawn from a power-law distribution with parameter α. Lévy walks are thus characterized by the power-law parameter α that tunes how often the walks exhibit short steps in contrast to longer ones. Of particular relevance is the range $\alpha \in (2, 3)$; outside this interval, the behavior of a Lévy walk is similar to those of other known movement models, namely, the Brownian motion ($3 \leq \alpha$), and the ballistic walk ($\alpha \leq 2$), which are well understood (see also Figure 1.7).

Starting with Shlesinger and Klafter in 1986 [SK86], over past decades, several studies have highlighted how the movement of many animal species and living organisms exhibit a pattern resembling that of Lévy walks [Rey18]. Nevertheless, the main reason Lévy walks have been the object of an intense investigation is that they have been mathematically proven to be optimal foraging strategies: in particular, when a single forager is searching for food in an environment that exhibits a "uniform" distribution of food locations, the Lévy walk with parameter $\alpha = 2$ outperforms other search strategies [VBH$^+$99]. The fact that Lévy walks rarely revisit previously explored areas seems to play a fundamental role in this respect. For such reasons, researchers have introduced the so-called Lévy flight foraging hypothesis, which states that, since Lévy walks can optimize search efficiency, then natural selection must have led to adaptations for Lévy walk foraging.

But what about animals that search collectively, like ants or bees? Together with my former student Francesco D'Amore and other colleagues, in [CdGN21] I have analyzed the search efficiency of a group of entities that must find a single target in the infinite two-dimensional grid. We consider the case in which all entities start together from the same position (e.g., a nest site) and must find as efficiently as possible a target that is placed at

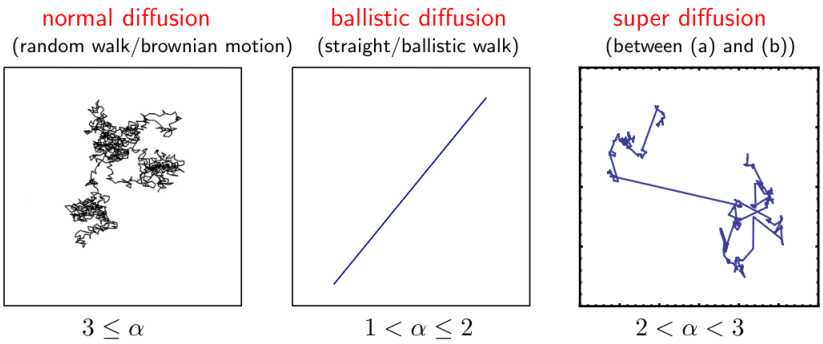

Figure 1.7: The three fundamental regimes of Lévy walks: on the left, the diffusive regime for the range of values of α for which the variance of the step length is finite ($\alpha > 3$); in the middle the ballistic regime for the range of values of α for which the expected length of the steps is infinite ($\alpha < 2$); on the right is the super-diffusive regime for the range of values of α for which the step length has finite expectation but infinite variance ($2 < \alpha < 3$).

some distance from the nest. In distributed computing, the above problem has been known as the *ANTS problem* [FK17]. We show that Lévy walks are the optimal search strategy; more precisely, the hitting time (i.e., the first time an entity finds the target) matches the smallest time that is required to find the target with constant probability. Depending on the target distance and the number of individuals composing the search group, the parameter α characterizing the search pattern needs to be accurately tuned to reach optimality. Nevertheless, if each entity in the nest chooses uniformly at random its parameter in the interval $\alpha \in (2, 3)$ and then performs a Lévy walk with that parameter, optimality is still achieved. This search algorithm is very simple, homogeneous, and performs as well as more artificial search strategies which were previously designed for this search problem [FK17].

Moreover, the aforementioned strategy, which yields optimal search efficiency for (almost) all distance scales, requires different members of the same group to follow different search patterns. The mathematical evidence for such variation in the search patterns among individuals of the same group raises interesting questions which would require experimental validation.

Overall, our result has been in line with other recent findings [GK20], showing that Lévy walks are surprisingly efficient movement patterns, thus offering new mathematical grounds for the Lévy flight foraging hypothesis.

1.3 v2.0: Computational Neuroscience

Let me recall how one of the mysteries inspiring me to pursue an academic career was the functioning of the human brain (Section 1.1). In 2016, I was very fortunate to have the opportunity to participate to a semester at the Simons Institute for the Theory of Computing as a visiting graduate student, and even more fortunate to know that the Institute was considering organizing a program that would foster collaborations between theoretical computer scientists and theoretical and computational neuroscientists. The prospect of such a unique possibility revamped my enthusiasm and I tried to stir with more determination my current research in a direction that could increase my chances to be granted a Simons Fellowship for such a program. The work [BNFK18], which I already mentioned in Section 1.2.1, grew largely out of that desire, as one can hardly attend several talks by theoretical neuroscientists without the role of noise being called into question. In that regard, there is usually an attempt to propose that noise plays a constructive role; in [BNFK18], on the contrary and less surprisingly, we investigated instead the limit that a stochastic and noisy biological system has to face to propagate information reliably.

One year later, in 2017, the program was officially announced and I was granted a fellowship. I can say that my research activity in theoretical neuroscience officially began right then, during the 2018 Brain and Computation Program of the Simons Institute for the Theory of Computing, although the first work that could be considered properly pertaining to the area I did not publish until 2022. This has been a collaboration with one of the program organizers, Christos Papadimitriou. In his words, the program would give me the chance to acquire, in few months, a birds-eye vision of the field of theoretical neuroscience comparable to what he got after several years of study. The reason for such a time gap in grasping the fundamentals of the discipline was indeed one of the main takeaways, which I find best summarized in the provocative question that was asked during the Simons workshop "What Is Missing in Current Theories of Brain Computation" on April 17th, 2018:

> What do all theoretical neuroscientists agree on, other than there are neurons that fire in the brain? - Abbas El Gamal.

After the program, Christos started publishing joint works on the Assembly Calculus, a theoretical framework that seeks to explain the emergence of high-level cognition from the low-level behavior of neurons and synapses through an algorithmic formalization of Hebbian learning [PVM$^+$20]. In

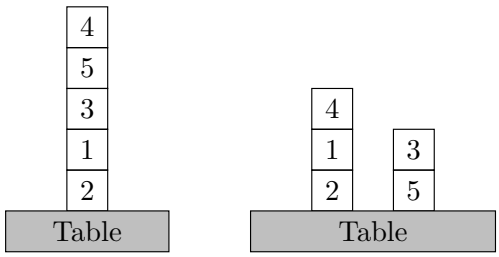

Figure 1.8: An instance of a Blocks World puzzle.

[MCP21], it is shown how the model can allow neurons to effectively implement a parser. In [dMC+22], we further show how such a framework can allow us to solve planning problems. Specifically, we consider the famous task of solving Blocks World puzzles [Win71, ST01], in which the disposition of blocks (which we can think of as child-toy blocks on a table) is modified via simple operations until a given configuration is reached (see e.g. Fig. 1.8). In [dMC+22] we show how simple programs solving Blocks World can be implemented in Assembly Calculus.

As I said, [dMC+22] was the first *theoretical neuroscience* project I finally published since taking part to the Simons Program in 2018. Since the Program, however, I've been trying to tackle El Gamal's question by making efforts to ground some theoretical ideas on real data. Unfortunately, the theoretical part is still far from becoming concrete, but the experimental part leads to three works. Let me spend a few words on the experimental part before sharing a brief consideration of the theoretical one.

The first one is still unpublished and consists in investigating whether the phylogenetic tree of more than one hundred mammals can be reconstructed based on the neural connectivity of their brains.

The second one is [FCC+21], in which we investigate a network alignment algorithm (i.e. a relaxed version of graph isomorphism, also known as graph matching [CFSV04]), and apply it to assess the robustness of brain atlases (see Fig. 1.9), which are canonical ways to partition the brain into regions according to various anatomical and functional criteria [FZB16]. The third one is also unpublished, but the large dataset has already been made publicly available in [RDN23]: it consists of a series of graphs representing the correlation of the activity among brain regions during fMRI recordings of the subject in a resting state (i.e. performing no task).

Back to theoretical considerations, theoretical neuroscience is a field that has historically been led by physicists. I do share Christos Papadimitriou's

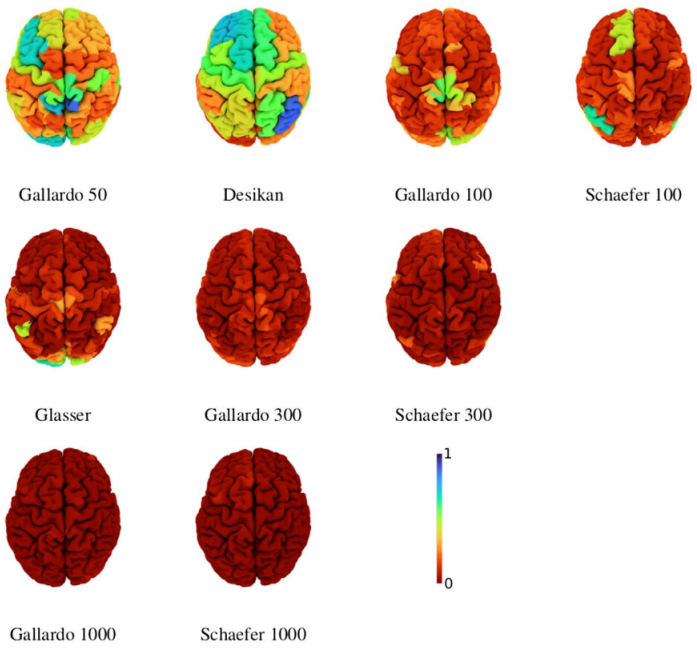

Figure 1.9: Illustration of the success rate with which regions for different atlases are correctly labeled by the methodology proposed in Frigo et al. (2021) after the regions are shuffled according to a uniform random permutation. Atlases with 100 regions or fewer are illustrated in the first row. The second row illustrates atlases with approximately 300 regions, and the third row those with 1,000 regions (see Section 1.3).

view, which he expressed in his joint survey [MPVL19], that theoretical computer science has much to contribute to the development of the field. Many theoretical computer scientists are indeed devoting energies in this direction.

As for myself, my contribution has been limited to the aforementioned work on the Assembly Calculus (which is also experimental). The reason for that is a variant of El Gamal's question: What are the conditions for making other theoretical neuroscientists accept a new theory? That is a problem that Science has solved, in principle, since its incipit, by forcing us to face empirical observations and experiments. Unfortunately, we can see how, in many domains, it is way easier for a theorist to base theory on theory, and much harder to base it on reality. Today, I am still struggling to get certain empirical observations on the brain, but in the last section, I will share more on a theoretical neuroscience topic I've worked on, Section 1.5, which is devoted to the main subject of the present thesis, namely my work in the theory of ANNs.

1.4 Digression in Integrated Assessment Modeling

Before introducing the subject that will be the main object of study of this work, let me mention another project that has absorbed a good deal of my time over the last two years. Since 2021, I've also been working on an integrated assessment modeling library, WorldDynamics.jl, to apply scientific machine learning to develop models related to sustainable development goals, which can be found at `https://github.com/worlddynamics/WorldDynamics.jl`. Such research interest emerged during the COVID quarantine periods in 2020, when the general atmosphere drifted my attention towards sustainability and ecological societal issues.

Given the importance of predicting how the global socioeconomic system interacts with major ecological aspects of the planet, I reasonably expected that substantial modeling effort had been done in that direction. I learned about the seminal work that Jay Wright Forrester and others carried out in the '70, which culminated in the World3 Model, giving birth to the area of Integrated Assessment Models [For73]. To my great surprise, however, the following decades didn't see the emergence of a rigorous and systematic approach. Forrester et al.'s modeling methodology crystallized in what became known as System Dynamics [BAN17], which is a modeling framework in which dependencies among variables fall in a limited set of mathematical relations. Many other models were proposed since World3, e.g. Nobel

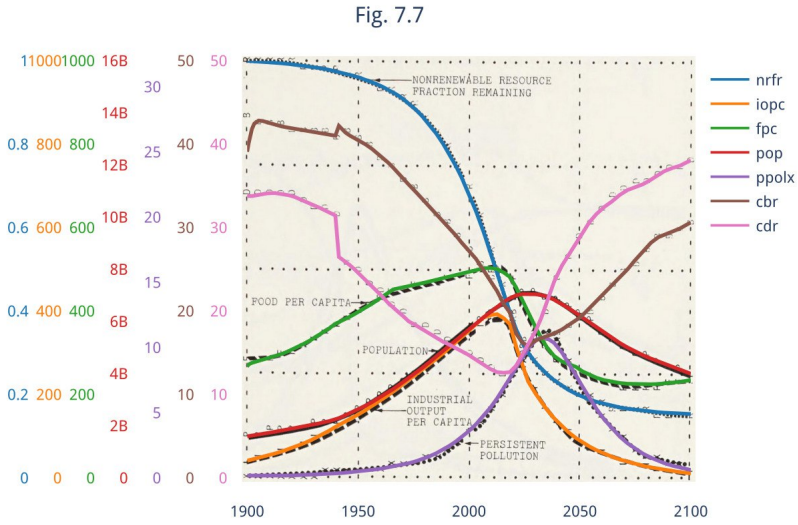

Figure 1.10: Evolution of some variables in a simulation of the World3 Model performed with the WorldDynamics.jl library, overlaid on the original Figure 7.7 from the famous book by Meadows et al. (see Section 1.4).

Laureate William Nordhaus' DICE Model [Nor18], but the relations among them remained debatable. As a side note, the lack of systematic scientific progress in Integrated Assessment Modeling is an interesting fact from the point of view of the sociology of science, and could have a direct relationship with the aforementioned fragmentation in theoretical neuroscience (Section 1.3).

A recent effort to provide a systematic framework to organize and compare models aimed at estimating carbon emissions is the MIMI Framework [MRL+18]. However, in 2020, a framework including World3 and many famous IAM was still missing. Besides that, the emerging Julia programming language had been developing an ecosystem of libraries on techniques at the intersection of machine learning and scientific computing that got recently grouped under the umbrella term of *Scientific Machine Learning [BAB+19]*. The perspective of applying such techniques for developing new IAMs seemed to have the potential to be scientifically significant.

When I shared the above considerations with Pierluigi Crescenzi in 2020, his shared enthusiasm for the endeavor quickly led us to embark on a re-implementation of the World3 model. Let me only say that the model had

been developed in 1970 before even the C language had been created; Forrester and his collaborators had thus developed a specific language called DYNAMO to implement and simulate the Wolrd3 model, which is today a dead language; we thus had to turn into software archaeologists and study old manuals of the language. Today, we are continuing to develop the library with the hope that the project could have a positive impact outside academia. The reader who's interested in some of the scientific aspects of the library is deferred to [CLN+23, CNRS23].

1.5 v3.0: Sparsification in ANNs

Artificial neural networks (ANNs) are a research topic that is often claimed to be backed up by little theory: they work in practice, and we don't know why. Given such an atmosphere, contributing to their understanding had been an appealing challenge since my PhD years. Such desire grew stronger when, in 2015, I attended for the first time the Biological Distributed Algorithms Workshop (BDA) to present my work on Noisy Consensus (mentioned in Section 1.2.1). There, Turing Award Leslie Valiant held an invited talk titled "*A computational model and theory of cortex*".

Starting from the empirical fact that the connections between neurons in our brains are much denser at birth and become progressively sparser throughout our lives, Valiant shared some considerations on the extent to which learning might be a sparsification process: Theoretically, starting with a complete graph, we could encode information by simply removing edges, like a sculptor carving stone. Already interested in sparsification techniques in theoretical computer science (which later led to the works mentioned in Section 1.2.4.4), the desire of exploring Valiant's speculation implanted in my brain and started growing when I was finally attending the Simons Program three years later.

Unfortunately, with the great kind of frustration discussed in Section 1.3, to date I cannot find sufficient experimental details on sparsification to justify some theoretical ideas. On the another hand, the investigation led me to the study of sparsification algorithms for ANNs, also known as pruning algorithms. So it happened that, since 2019 ANN pruning has been a major part of my research [BNV19]. Along with the investigation of the latter, I should also mention some joint work on security aspects of ANNs [dCNV23], and on a patent that we filed on a neuromorphic computing approach that emerged from some of the ideas discussed in the next chapters ([DCNV17]).

Finally, we have arrived at what will be the technical focus of this work. I

hope that the reader will be able to appreciate, as mentioned at the beginning of the chapter, the heterogeneity of my research, and the consequent difficulty in isolating a topic that can be the subject of a sufficiently cohesive treatise. I hope even more strongly that the final choice will prove useful to those interested in the topic.

1.6 Supervising PhD students

I trace back the start of my student supervision activity to 2017 when, as a postdoctoral fellow of the Max Planck Institute for Informatics, I started supervising interns. During my Ph.D., I already carried on some projects with other students, thus without the guidance of more senior scientists, such as [BN16] (later published in [BN19]). At the Max Planck, however, I was for the first time *responsible* for the research activity of undergraduate students, which were now less experienced than me. I was very fortunate to work with bright students. With one of them, Iliad Ramezani, we worked during the summer of 2018 on proving upper and lower bounds on the necessary memory for the problem of computing the relative majority in the famous distributed-computing model known as Population Protocols, which falls within the research topics discussed in Section 1.2. Our work was later published in [RN19].

Even deeper was my collaboration with two other Ph.D. students, Giacomo Scornavacca and Emilio Cruciani, which were visiting the Max Planck for a few months. I have already mentioned several works we co-authored in Section 1.2.2, in particular, [CNNS18, CNS19] with both Emilio Cruciani and Giacomo Scornavacca, and [CGPS17, CGG$^+$18, CGN$^+$20] with Giacomo Scornavacca. Given the extent of our collaboration, I was kindly offered to become a co-supervisor of Emilio Cruciani, who then became my first PhD student, defending his thesis in 2019. Emilio later joined the COATI project-team, as a postdoctoral fellow and played a key role in the interdisciplinary collaboration on the topic of *brain alignment* discussed in Section 1.3, which resulted in the publication [FCC$^+$21].

My second Ph.D. student has been Francesco D'Amore, who started his Ph.D. in 2019 in COATI under the direction of Nicolas Nisse (HDR). I have written about our several joint works in sections 1.2, 1.2.4.5 and 1.3. With Francesco D'Amore, my share of responsibility was even higher than with Emilio Cruciani, as I was also the one who proposed the topic of his Ph.D. thesis. However, his mathematical talent and his kind personality made working with him a real pleasure. He brilliantly defended his thesis in 2022,

and nowadays we are continuing to interact closely while he is continuing to pursue his academic career as a postdoctoral fellow in prestigious international research groups in theoretical computer science.

My third Ph.D. student has been Arthur Carvalho Walraven Da Cunha[2], for which I got the derogation to be his supervisor without having the HDR. Unfortunately, the first part of his thesis has been carried on during the COVID-19 pandemic, which made our collaboration difficult for some time. Despite that, we managed to carry on the technical topic that constitutes the main part of this document, namely chapters 2, 3 and 4. Several of our works have initially been rejected by top-tier conferences, but we did not give up and we managed to publish several of them in the end, namely [dCNV22], [dCNV23] and [dCdG$^+$22][3], while others are currently under review.

Already when I was a Ph.D. student, I could see the great diversity of philosophies and approaches to student supervision. Some supervisors are distant, quite formal and only meet with their students a few hours per week (or even months). Others are informal, meet with their students every day and are very close to them. The latter has been the case for my Ph.D. supervisor, Prof. Andrea Clementi, and I have tried to follow his example. Today, I think that the distant approach is easier to follow, but the close approach is more rewarding for both the supervisor and the student. On the other hand, it is fundamental to select the student quite carefully, as a closer relationship does not allow to follow too many students at the same time, and it can even be counterproductive if the student is not motivated enough. Luckily for me, when I compare my experience to that of many colleagues, I can see that I have been lucky with my students. Of course, as Luis Pasteur once said, "Chance only favors the mind which is prepared"; I very much hope to continue having opportunities to supervise motivated and brilliant students in the future, and to seize them by being the best supervisor I can be in helping them achieve their potential as researchers.

[2]At the time of writing, he just submitted the manuscript of his Ph.D. thesis and is expected to defend in September 2023.

[3]At the time of writing, the paper has been accepted to the European Symposium on Algorithms 2023.

Chapter 2

Introduction to RSS

On voit, par cet Essai, que la théorie des probabilités n'est, au fond, que le bon sens réduit au calcul; elle fait apprécier avec exactitude ce que les esprits justes sentent par une sorte d'instinct, sans qu'ils puissent souvent s'en rendre compte. - Pierre-Simon Laplace [Lap18]

This chapter introduces the Random Subset Sum Problem (RSSP) and the results that will be used in the following chapter. Our interest for the problem in the context of machine learning is motivated by its application to the Strong Lottery Ticket Hypothesis, which will be presented in Chapter 4, after presenting a simplified proof of the main result about the RSSP that had been leveraged in this context.

First, let us recall the classical (deterministic) Subset Sum Problem.

Definition 1 (Subset Sum Problem (SSP))**.** Given n integers $x_1, ..., x_n$ and a target value z, the SSP is the decision problem of determining whether there exists a subset $S \subseteq \{x_1, ..., x_n\}$ such that $\sum_{x \in S} x = z$.

To appreciate the depth of the SSP, we also need to remark on how it is essentially equivalent to another fundamental decision problem in complexity theory, the Number Partition Problem (NPP). We defer such important digression on the relation between the two problems in Section 2.0.1. Next, we give two important definitions for approaching the *approximate* variant of SSP that we are interested in in this work.

Definition 2 (ϵ-approximation)**.** We say that x ϵ-approximates z if $|x - z| \leq \epsilon$.

Definition 3 (ϵ-approximable)**.** Given a set of values $\omega_n = \{x_1, ..., x_n\}$, we say that z is ϵ-approximable with ω_n if there exists a subset $S_z \subseteq \omega_n$ such that the sum of its elements $\sum_{x \in S_z} x$ ϵ-approximates z.

We are now ready to provide a formal statement of the *Random* SSP.

Problem 4 (Random Subset Sum Problem (RSSP))**.** Let $\Omega_n = \{X_1, ..., X_n\}$ be i.i.d. uniform random variables over $[-1, 1]$. Given $\epsilon > 0$ and $z \in [-1, 1]$, is there a subset $S \subseteq \Omega_n$ such that the sum of its elements $\sum_{x \in S} x$ ϵ-approximates z?

The study of RSSP intensified after Lueker applied it to the estimation of the integrality gap (i.e. the difference between the values of the integer and relaxed solutions) for random instances of the Knapsack Problem, a classical NP-complete problem defined by the following integer linear program (ILP).

Definition 5 (0-1 Knapsack Problem [Kar72])**.** Given $2n + 1$ positive reals $a_1, ..., a_n, b_1, ..., b_n$ and B, an instance of the 0-1 Knapsack Problem is defined by the following ILP:

$$\text{maximize } \sum_{i=1}^{n} z_i a_i$$

$$\text{subject to } \begin{cases} \sum_{i=1}^{n} z_i b_i \leq B, \\ z_i \in \{0, 1\}. \end{cases}$$

To explain the efficacy of fast backtracking algorithms for Knapsack, in [Lue82] Lueker showed that RSSP naturally appears in the analysis of a greedy procedure for the problem, and managed to prove, under relatively mild assumptions, that the integrality gap is at most $O\left(\frac{\log n}{n}\right)$ and at least $\Omega\left(\frac{1}{n}\right)$. His proof relied on a second moment method [AS16], and was later generalized by Dyer and Frieze to random packing integer programs (IP) [DF89] and very recently by Borst et al. to random binary IP [BDHT].

In this work, we are going to focus on one of several refined results that Lueker proved in 1998 on RSSP, which we state next.

Theorem 6 ([Lue98],[dCdG$^+$22])**.** *Let $\Omega_n = \{X_1, ..., X_n\}$ be i.i.d. uniform random variables over $[-1, 1]$ and $\epsilon \in \left(0, \frac{1}{3}\right)$. There is a universal constant*[1] *C such that if $n \geq C \log \frac{1}{\epsilon}$ then with probability at least $1 - \epsilon$ any $z \in [-1, 1]$ is ϵ-approximable with Ω_n.*

Lueker proved a slight variant of Theorem 6 by constructing a clever martingale and applying the Azuma-Hoeffding inequality to it [DP09]. In

[1] I.e. a constant independent from n and ϵ.

Chapter 3, we will propose a more elementary, alternative proof that does not require familiarity with such relatively advanced tools and should thus be more accessible.

Remark 7. In [Lue98], it is remarked that $C \leq 2\,(1 + \log e)$. In its present form, the approach that we present in Chapter 3 leads to a much larger bound, namely $742.27 \log \frac{1}{2\epsilon}$, but we did not try to optimize the resulting constant.

Before presenting some results related to Theorem 6 that will be useful in applying the latter in the context of artificial neural network pruning in Chapter 4, in the next section we are going to briefly discuss the relationship between SSP and NPP.

2.0.1 The Subset Sum Problem and the Number Partition Problem

An instance of the SSP is easily transformed into an instance of NPP, one of the six basic NP-complete problems of historical importance in Computational Complexity Theory [GJ79]. Let us define NPP and show next how to reduce SSP to NNP.

Definition 8 (Number Partition Problem (NPP))**.** Given n integers $x_1, ..., x_n$, the NPP is the decision problem of determining whether there exists a subset $S \subseteq \{x_1, ..., x_n\}$ such that[2] $\sum_{x \in S} x - \sum_{x \notin S} x = 0$.

Given an instance of SSP with n integers $x_1, ..., x_n$ and a target value z, an equivalent instance of NPP is obtained by considering the set of $n + 1$ integers $\{x_1, ..., x_n, \sum_{i=1}^{n} x_i - 2z\}$. Indeed, we can assume without loss of the generality that the solution to such NPP will be of the form

$$\left(\sum_{i=1}^{n} x_i - 2z\right) + \sum_{x \in S} x - \sum_{x \notin S} x = 0 \tag{2.1}$$

for some $S \subseteq \{x_1, ..., x_n\}$. Using the fact that $\sum_{i=1}^{n} x_i - \sum_{x \notin S} x = \sum_{x \in S} x$, we can rewrite which we can rewrite Eq. 2.1 as $\sum_{x \in S} x = z$, which is a solution to the SSP.

For completeness, we observe that by a similar reasoning an instance of the NPP problem with input $x_1, ..., x_n$ can be transformed into an instance of the SSP by considering the same set of input values $\{x_1, ..., x_n\}$ and target value $\frac{1}{2} \sum_i x_i$.

[2] We denote with $\bar{A}$ the complement of the set A.

Hence, the SSP is equivalent to the NPP. We remark here that, while in this work, we focus on a random version of SSP, some works have investigated random versions of NPP. The above argument shows that results for one of them can often be directly translated into results for the other. We will not leverage results obtained on the NPP in this work, but for the sake of completeness, we mention that random versions of NPP have attracted a lot of interest in Statistical Physics [MM09], in particular concerning the study of phase transitions in random combinatorial structures [Mer01, BCP01, BCMP04].

2.1 Useful corollaries

In his section, we show how we can easily extend Theorem 6 to more general distributions that *contain* a uniform distribution. We will see an application of such a corollary in Chapter 4.

Let us begin by making precise the notion of containing a uniform distribution.

Definition 9 (Super-uniform variable)**.** Given $a > 0$ and $b \in \left(0, \frac{1}{2a}\right)$, a random variable X is (a,b)-super-uniform if its density f_X satisfies $f_X(x) \geq b$ for each $x \in [-a,a]$. We simply say that X is super-uniform if there exist a and b such that X is (a,b)-super-uniform.

We can now state and prove the following corollary of Theorem 3.

Corollary 10 (RSS for super-uniform variables)**.** *Let $\Omega_n = \{X_1, ..., X_n\}$ be independent (a,b)-super-uniform random variables over and $\epsilon \in \left(0, \frac{1}{3}\right)$. There is a constant $C_{a,b}$, depending only on a and b, such that if $n \geq C_{a,b} \log \frac{1}{\epsilon}$ then with probability at least $1-\epsilon$ any $z \in [-a,a]$ is ϵ-approximable with Ω_n.*

Proof. Let f_i be the density of X_i and $U_1, ..., U_n$ be n i.i.d. standard uniform random variables. Consider a sample $\Omega_n = \{X_1, ..., X_n\}$ of independent (a,b)-super-uniform random variables and define the *pruned* sample

$$\tilde{\Omega}_n = \left\{X_i : (X_i \in \Omega_n) \wedge (X_i \in [-a,a]) \wedge (U_i \cdot f(X_i) \leq b)\right\}_{i \in [n]}.$$

One can verify that for each i and each $x \in [-a,a]$

$$\begin{aligned} &\Pr\left(X_i = x \,\middle|\, X_i \in \tilde{\Omega}_n\right) \\ &= \frac{\Pr\left((X_i = x) \wedge (X_i \in [-a,a]) \wedge (U_i \cdot f_i(X_i) \leq b)\right)}{\Pr\left((X_i \in [-a,a]) \wedge (U_i \cdot f_i(X_i) \leq b)\right)}, \end{aligned}$$

$$
\begin{aligned}
&= \frac{\Pr\left(U_i \cdot f_i\left(x\right) \leq b \,|\, X_i = x\right) \Pr\left(X_i = x\right)}{\int_{-a}^{a} \Pr\left(U_i \cdot f_i\left(t\right) \leq b\right) f_i\left(t\right) dt} \\
&\quad \text{since } f_i\left(x\right) \geq b \text{ for } x \in [-a, a] \text{ and } \Pr\left(X_i = x\right) = f_i\left(x\right) \\
&= \frac{\frac{b}{f_i(x)} f_i\left(x\right)}{\int_{-a}^{a} \frac{b}{f_i(t)} f_i\left(t\right) dt} \\
&= \frac{1}{2a}
\end{aligned}
$$

which means that each $X_i \in \tilde{\Omega}_n$ has distribution $\text{Unif}\left([-a, a]\right)$. Moreover, since

$$\Pr\left(X_i \in \tilde{\Omega}_n\right) \geq 2ab,$$

we can choose $C_{a,b}$ large enough so that a straightforward application of a Chernoff bound yields

$$\Pr\left(\left|\tilde{\Omega}_n\right| \geq C \log \frac{1}{\epsilon} \,\middle|\, |\Omega_n| \geq C_{a,b} \log \frac{1}{\epsilon}\right) \geq 1 - \frac{\epsilon}{2},$$

where C is the universal constant of Theorem 6. We can thus apply Theorem 6 to $\tilde{\Omega}_n$ with error $\epsilon' = \frac{\epsilon}{2}$, which implies that the overall probability that $\tilde{\Omega}_n$ is sufficiently large and that all values are ϵ-approximable is $\left(1 - \frac{\epsilon}{2}\right)^2 > 1 - \epsilon$. □

A case of special interest, because of its occurrence in applications, is when the random samples follow the distribution of the product of two uniform random variables, which we provide in the next lemma.

Lemma 11 (Product of uniforms)**.** *If X_1 and X_2 are independent* $\text{Unif}\left([-1, 1]\right)$ *random variables then $X_1 \cdot X_2$ is $\left(\frac{1}{2}, \frac{\log 2}{2}\right)$-super-uniform.*

As we mentioned above, in Chapter 4 we are going to leverage Corollary 10 and Lemma 11 to prove some interesting results on random convolutional neural networks. Before that, in the next Chapter, we are going to provide an alternative proof of Theorem 6 to that originally given in [Lue98].

Chapter 3

An Elementary Proof of Random Subset Sum

> I am going to give what I will call an elementary demonstration. But elementary does not mean easy to understand. Elementary means that very little is required to know ahead of time to understand it, except to have an infinite amount of intelligence - R. P. Feynman, [FGG96]

This chapter contains a more streamlined presentation of the proof of Theorem 6 given in [dCdG+22], which leverages a common analysis technique for Rumor Spreading protocols (see e.g. [DK]).

Compared to the original proof in [Lue98], this new proof does require familiarity with the concept of martingale and the Azuma-Hoeffding inequality and it involves easier calculations. It is thus more accessible and should be easier to adapt to variants of the original theorem. The initial setup will be the same as that of the original proof. We recall that our goal is to be able to ϵ-approximate all values in $[-1, 1]$. We can thus estimate our progress towards the latter goal by measuring the fraction of values $z \in [-1, 1]$ which are ϵ-approximable with a sample of a given size. We make the latter notion precise in the next definition.

Definition 12. The density of approximable values Z_{Ω_n} with a sample $\Omega_n = \{X_1, ..., X_n\}$ is

$$Z_{\Omega_n,\epsilon} = \frac{1}{2}\int_{-1}^{1} Y_{\Omega_n}(z)\, dz, \tag{3.1}$$

where $Y_{\Omega_n,\epsilon}(z)$ is the indicator random variable which is 1 if and only if z is ϵ-approximable with Ω_n.

We can then look at a sequence of samples of increasing size $\Omega_1 \subseteq \Omega_2 \subseteq ... \subseteq \Omega_n \subseteq ...$ where $\Omega_i = \{X_1, ..., X_i\}$ for each i, and at the corresponding sequence $Y_{\Omega_1,\epsilon}(z), ..., Y_{\Omega_n,\epsilon}(z), ...$ as if it were a discrete-time stochastic process.

Notation 13. For simplicity's sake, in the following, we often abuse notation and, instead of $Z_{\Omega_n,\epsilon}$ and $Y_{\Omega_n,\epsilon}(z)$, we write $Z_{n,\epsilon}$ and $Y_{n,\epsilon}(z)$ respectively, or even just Z_n and $Y_n(z)$.

A key observation is then that the process $Y_1(z), ..., Y_n(z), ...$ satisfies the following recurrence equation.

Lemma 14. *Given* $\Omega_n = \{X_1, ..., X_n\}$, $\Omega_{n+1} = \Omega_n \cup \{X_{n+1}\}$ *and* $z \in \mathbb{R}$, Y_n *satisfies the recurrence*

$$Y_{n+1}(z) = Y_n(z) + (1 - Y_n(z)) Y_n(z - X_{n+1}). \tag{3.2}$$

Proof. When trying to approximate z with the sum of a subset of $\Omega_{n+1} = \{X_1, ..., X_{n+1}\}$, we can distinguish two cases

1. There is a subset achieving that without including X_{n+1}, in which case $Y_n(z) = 1$, or
2. It is necessary to include X_{n+1}, which implies $Y_{n+1}(z) = Y_n(z - X_{n+1})$.

□

Our main goal is then to exploit Eq. 3.2 to prove that all values will be ϵ-approximable with a certain probability and sample size. In particular, we will leverage the following fact and focus on proving that, for some n, with high probability $Z_n \geq 1 - \epsilon$.

Lemma 15. *If* $Z_{n,\epsilon} \geq 1 - \frac{\epsilon}{2}$, *then* $Z_{n,2\epsilon} = 1$.

Proof. Let z' be a value which is **not** ϵ-approximable with Ω_n. Since $Z_{n,\epsilon} \geq 1 - \frac{\epsilon}{2}$, then z' is at distance at most ϵ from a value z which **is** ϵ-approximable with Ω_n. Let x be the sum of the subset which ϵ-approximates z. By the triangle inequality

$$\left|z' - x\right| \leq \left|z' - z\right| + |z - x| \leq \epsilon.$$

□

3.1 Exponential Increase Phase

Still following [Lue98], we are now going to look at the expectation of the random variable Z_n. However, we are going to depart from his analysis by providing different bounds and leveraging them in a different way.

Proposition 16 (Expected growth)**.** *It holds*

$$\mathbb{E}\left[Z_{n+1} \mid X_1, ..., X_n\right] \geq Z_n \left(1 + \frac{1}{4}\left(1 - Z_n\right)\right).$$

Proof. We have

$$\mathbb{E}\left[Z_{n+1} \mid X_1, ..., X_n\right]$$

from Definition 12

$$= \mathbb{E}\left[\frac{1}{2}\int_{-1}^{1} Y_{n+1}\left(z\right) dz \,\middle|\, X_1, ..., X_n\right]$$

bringing the integral outside

$$= \frac{1}{2}\int_{-1}^{1} \mathbb{E}\left[Y_{n+1}\left(z\right) \mid X_1, ..., X_n\right] dz$$

from Lemma 14

$$= \frac{1}{2}\int_{-1}^{1} \mathbb{E}\left[Y_n\left(z\right) + \left(1 - Y_n\left(z\right)\right) Y_n\left(z - X_{n+1}\right) \mid X_1, ..., X_n\right] dz$$

making expectation explicit

$$= \frac{1}{2}\int_{-1}^{1} \frac{1}{2}\int_{-1}^{1} \left(Y_n\left(z\right) + \left(1 - Y_n\left(z\right)\right) Y_n\left(z - x\right)\right) dxdz$$

distributing integrals

$$= \frac{1}{2}\int_{-1}^{1} Y_n\left(z\right)\left(\frac{1}{2}\int_{-1}^{1} dx\right) dz + \frac{1}{2}\int_{-1}^{1} \left(1 - Y_n\left(z\right)\right) \frac{1}{2}\int_{-1}^{1} Y_n\left(z - x\right) dxdz$$

recognizing Z_n and changing variable $z - x = y$

$$= Z_n + \frac{1}{4}\int_{-1}^{1} \left(1 - Y_n\left(z\right)\right) \int_{z-1}^{z+1} Y_n\left(y\right) dydz. \tag{3.3}$$

In order to provide a lower bound on $\int_{-1}^{1}\left(1 - Y_n\left(z\right)\right)\int_{z-1}^{z+1} Y_n\left(y\right) dxd$, notice that $Y_n\left(z\right)$ and $1 - Y_n\left(z\right)$ are non-negative, hence for each $u \in \left[-\frac{1}{2}, \frac{1}{2}\right]$ and $z \in \left[u - \frac{1}{2}, u + \frac{1}{2}\right]$ we can just restrict the domain of integration, yielding

$$\int_{-1}^{1} \left(1 - Y_n\left(z\right)\right) \int_{z-1}^{z+1} Y_n\left(y\right) dydz$$

$$\begin{aligned}
&\text{since } \left[u-\frac{1}{2},u+\frac{1}{2}\right]\subseteq[-1,1]\\
&\geq \int_{u-\frac{1}{2}}^{u+\frac{1}{2}}\left(1-Y_n\left(z\right)\right)\int_{z-1}^{z+1}Y_n\left(y\right)dydz\\
&\text{since } \left[u-\frac{1}{2},u+\frac{1}{2}\right]\subseteq[z-1,z+1]\\
&\geq \int_{u-\frac{1}{2}}^{u+\frac{1}{2}}\left(1-Y_n\left(z\right)\right)\int_{u-\frac{1}{2}}^{u+\frac{1}{2}}Y_n\left(y\right)dydz. \qquad (3.4)
\end{aligned}$$

We would like to express the latter lower bound in terms of Z_n. Using the intermediate value theorem, we can prove that there is a value $u^*\in\left[-\frac{1}{2},\frac{1}{2}\right]$ for which $\int_{u^*-\frac{1}{2}}^{u^*+\frac{1}{2}}Y_n\left(y\right)dy=Z_n$. To see why, consider the function $h\left(u\right)=\frac{1}{2}\int_{u-\frac{1}{2}}^{u+\frac{1}{2}}Y_n\left(y\right)dy$. h is continuous with $h\left(-\frac{1}{2}\right)=\int_{-1}^{0}Y_n\left(y\right)dy$ and $h\left(\frac{1}{2}\right)=\int_{0}^{1}Y_n\left(y\right)dy$. Since $h\left(-\frac{1}{2}\right)+h\left(\frac{1}{2}\right)=\int_{-1}^{1}Y_n\left(y\right)dy=2Z_n$, then

$$\max\left\{h\left(-\frac{1}{2}\right),h\left(\frac{1}{2}\right)\right\}\geq Z_n \quad\text{and}\quad \min\left\{h\left(-\frac{1}{2}\right),h\left(\frac{1}{2}\right)\right\}\leq Z_n.$$

Hence, as announced, from the intermediate value theorem there exists a $u^*\in\left[-\frac{1}{2},\frac{1}{2}\right]$ such that $h\left(u^*\right)=Z_n$. We can then lower bound Eq. 3.4 with $\left(1-Z_n\right)Z_n$, and Eq. 3.3 with $Z_n\left(1+\frac{1}{4}\left(1-Z_n\right)\right)$, concluding the proof. □

Proposition Eq. 16 shows that, as long as the fraction of approximable values does not exceed a constant fraction, it grows exponentially in expectation. Next, in Section 3.1.1, we are going to turn such expected growth into a bound on the probability that $Z_{n+1}\geq\left(1+\beta\right)Z_n$.

Before doing that, it is instructive to derive a bound on the *expected time* for Z_n to exceed $\frac{1}{2}$.

Proposition 17. *18If* $n\geq\frac{\log\frac{1}{2\varepsilon}}{\log\frac{13}{12}}$ *then*

$$\mathbb{E}\left[Z_n\right]>\frac{1}{2}.$$

Proof. We have

$$\begin{aligned}
&\mathbb{E}\left[Z_{n+1}\right]\\
&\text{by the law of total expectation}\\
&=\mathbb{E}\left[\mathbb{E}\left[Z_{n+1}\mid X_1,...,X_n\right]\right]
\end{aligned}$$

by the law of total probability

$$= \mathbb{E}\left[\mathbb{E}\left[Z_{n+1}\,\middle|\,X_1,...,X_n,Z_n<\frac{2}{3}\right]\Pr\left(Z_n<\frac{2}{3}\right)\right.$$
$$\left.+\,\mathbb{E}\left[Z_{n+1}\,\middle|\,X_1,...,X_n,Z_n\geq\frac{2}{3}\right]\Pr\left(Z_n\geq\frac{2}{3}\right)\right]$$

lower bounding the second conditional expectation

$$\geq \mathbb{E}\left[\mathbb{E}\left[Z_{n+1}\,\middle|\,X_1,...,X_n,Z_n<\frac{2}{3}\right]\Pr\left(Z_n<\frac{2}{3}\right)+\frac{2}{3}\Pr\left(Z_n\geq\frac{2}{3}\right)\right]$$

by Proposition 16

$$\geq \mathbb{E}\left[Z_n\left(1+\frac{1}{12}\right)\Pr\left(Z_n<\frac{2}{3}\right)+\frac{2}{3}\Pr\left(Z_n\geq\frac{2}{3}\right)\right]$$
$$= \mathbb{E}\left[Z_n\right]\left(1+\frac{1}{12}\right)\Pr\left(Z_n<\frac{2}{3}\right)+\frac{2}{3}\Pr\left(Z_n\geq\frac{2}{3}\right)$$
$$= \mathbb{E}\left[Z_n\right]\left(1+\frac{1}{12}\right)\left(1-\Pr\left(Z_n\geq\frac{2}{3}\right)\right)+\frac{2}{3}\Pr\left(Z_n\geq\frac{2}{3}\right)$$
$$= \mathbb{E}\left[Z_n\right]\left(1+\frac{1}{12}\right)+\left(\frac{2}{3}-\mathbb{E}\left[Z_n\right]\left(1+\frac{1}{12}\right)\right)\Pr\left(Z_n\geq\frac{2}{3}\right)$$
$$\geq \mathbb{E}\left[Z_n\right]\left(1+\frac{1}{12}\right)\mathbf{1}_{\left[\mathbb{E}[Z_n]\left(1+\frac{1}{12}\right)\leq\frac{2}{3}\right]}$$
$$\geq \mathbb{E}\left[Z_{n-1}\right]\left(1+\frac{1}{12}\right)^2\mathbf{1}_{\left[\mathbb{E}[Z_{n-1}]\left(1+\frac{1}{12}\right)\leq\frac{2}{3}\right]}\mathbf{1}_{\left[\mathbb{E}[Z_n]\left(1+\frac{1}{12}\right)\leq\frac{2}{3}\right]}$$

since $Z_n \geq Z_{n-1}$ we can drop further $\mathbf{1}_{[\cdot]}$ functions and keep unrolling

$$\geq \mathbb{E}\left[Z_0\right]\left(1+\frac{1}{12}\right)^{n+1}\mathbf{1}_{\left[\mathbb{E}[Z_n]\left(1+\frac{1}{12}\right)\leq\frac{2}{3}\right]}$$

since $\mathbb{E}\left[Z_0\right] \geq \epsilon$

$$\geq \epsilon\left(1+\frac{1}{12}\right)^{n+1}\mathbf{1}_{\left[\mathbb{E}[Z_n]\left(1+\frac{1}{12}\right)\leq\frac{2}{3}\right]}.$$

If $n \geq \frac{\log\frac{1}{2\epsilon}}{\log\frac{13}{12}}$, the last inequality implies $\mathbb{E}\left[Z_n\right] \geq \frac{1}{2}\mathbf{1}_{\left[\mathbb{E}[Z_{n-1}]\left(1+\frac{1}{12}\right)\leq\frac{2}{3}\right]}$, which in turn implies that either $\mathbb{E}\left[Z_n\right] \geq \frac{1}{2}$ (if $\mathbb{E}\left[Z_{n-1}\right]\left(1+\frac{1}{12}\right) \leq \frac{2}{3}$) or $\mathbb{E}\left[Z_{n-1}\right] > \frac{8}{13} > \frac{1}{2}$. □

Unfortunately, trying to derive a bound with high probability on the event "$Z_n > \frac{1}{2}$" by directly applying a Chernoff-Hoeffding bound to Propo-

sition 18 would yield an extra $\log \frac{1}{\epsilon}$ time factor. Saving the latter factor is the goal of the more careful analysis, that is given in the next section.

3.1.1 Interval partition with geometric coupling

In the next proposition, we are going to convert the expected growth estimated in Proposition 16 to a probabilistic bound.

Proposition 18. *Given $\beta \in \left(0, \frac{1}{8}\right)$, it holds*

$$\Pr\left(Z_{n+1} \geq Z_n\left(1+\beta\right) \middle| X_1, ..., X_n, Z_n \leq \frac{1}{2}\right) \geq 1 - \frac{7}{8\left(1-\beta\right)}.$$

Proof. Notice that if $Z_n \leq \frac{1}{2}$, Proposition 16 implies

$$\mathbb{E}\left[Z_{n+1} \middle| X_1, ..., X_n, Z_n \leq \frac{1}{2}\right] \geq Z_n\left(1+\frac{1}{8}\right). \tag{3.5}$$

To prove the thesis, we need an upper bound on the value of Z_{n+1}. To that end, we are going to use a reversed form of Markov's inequality, stated in the following lemma.

Lemma 19 (Reverse Markov's inequality [BGPS06])**.** *If X is a random variable such that $0 \leq X \leq B$, then for any $t \in (0, B)$ it holds*

$$\Pr\left(X \geq t\right) \geq \frac{\mathbb{E}\left[X\right] - t}{B - t}.$$

Proof. We have

$$\begin{aligned}\mathbb{E}\left[X\right] &\leq t \Pr\left(X < t\right) + B \Pr\left(X \geq t\right) \\ &= t + \left(B - t\right) \Pr\left(X \geq t\right).\end{aligned}$$

□

While $Z_{n+1} \leq 1$ with probability 1, such a bound would not serve the purpose: as we will see shortly, the upper bound is going to appear in the denominator of the reverse Markov's inequality, and we need it to have Z_n as a factor to cancel a corresponding factor in the numerator. To achieve that, we consider a random variable $\hat{Z}_n$ which is always smaller than Z_n, by defining it as $\hat{Z}_n = \frac{1}{2}\int_{-1}^{1} \hat{Y}_n\left(z\right) dz$ where

$$\hat{Y}_{n+1}\left(z\right) = Y_n\left(z\right) + \left(1 - Y_n\left(z\right)\right) Y_n\left(z - X_{n+1}\right) \mathbf{1}_{z - X_{n+1} \in [-1,1]}. \tag{3.6}$$

By comparing Eq. 3.6 with Eq. 3.2, we see that

$$\Pr\left(\hat{Z}_n \leq Z_n\right) = 1. \tag{3.7}$$

Moreover

$$\begin{aligned}
\hat{Z}_{n+1} &= \frac{1}{2}\int_{-1}^{1} \hat{Y}_{n+1}(z)\, dz \\
&\text{by Eq. 3.6} \\
&= \frac{1}{2}\int_{-1}^{1} \left(Y_n(z) + (1 - Y_n(z))\, Y_n(z - X_{n+1})\, \mathbf{1}_{z - X_{n+1} \in [-1,1]}\right) dz \\
&\text{distributing the integral} \\
&= Z_n + \frac{1}{2}\int_{-1}^{1} (1 - Y_n(z))\, Y_n(z - X_{n+1})\, \mathbf{1}_{z - X_{n+1} \in [-1,1]} dz \\
&\text{discarding } 1 - Y_n(z) \\
&\leq Z_n + \frac{1}{2}\int_{-1}^{1} Y_n(z - X_{n+1})\, \mathbf{1}_{z - X_{n+1} \in [-1,1]} dz \\
&\text{substituting } y = z - X_{n+1} \\
&= Z_n + \frac{1}{2}\int_{-1-X_{n+1}}^{1-X_{n+1}} Y_n(y)\, \mathbf{1}_{y \in [-1,1]} dy \\
&\text{making the indicator function implicit in the integration domain} \\
&= Z_n + \frac{1}{2}\int_{\max\{-1,-1-X_{n+1}\}}^{\min\{1,1-X_{n+1}\}} Y_n(y)\, dy \\
&\leq Z_n + \frac{1}{2}\int_{-1}^{1} Y_n(y)\, dy \\
&\leq 2Z_n. \qquad (3.8)
\end{aligned}$$

Hence, we can compute

$$\Pr\left(Z_{n+1} \geq Z_n(1+\beta) \,\middle|\, X_1, ..., X_n, Z_n \leq \frac{1}{2}\right)$$

by Eq. 3.7

$$\Pr\left(\hat{Z}_{n+1} \geq Z_n(1+\beta) \,\middle|\, X_1, ..., X_n, Z_n \leq \frac{1}{2}\right)$$

by the reverse Markov's inequality (Lemma 19) and Eq. 3.8

$$\geq \frac{\mathbb{E}\left[\hat{Z}_{n+1} \,\middle|\, X_1, ..., X_n, Z_n \leq \frac{1}{2}\right] - Z_n(1+\beta)}{2Z_n - Z_n(1+\beta)}$$

by Eq. 3.5

$$\begin{aligned}
&\geq \frac{Z_n\left(1+\frac{1}{8}\right) - Z_n\left(1+\beta\right)}{2Z_n - Z_n\left(1+\beta\right)} \\
&= \frac{\frac{1}{8}-\beta}{1-\beta} \\
&= 1 - \frac{7}{8\left(1-\beta\right)}.
\end{aligned}$$

□

We can now leverage Proposition 18 to upper bound the probability of the duration of the first phase of the process, in which Z_n exceeds $\frac{1}{2}$.

Proposition 20. *Let* $\beta \in \left(0, \frac{1}{8}\right)$, $p_\beta = 1 - \frac{7}{8(1-\beta)}$ *and* $\alpha_{\beta,\epsilon} = \left\lceil \frac{\log \frac{1}{2\epsilon}}{\log(1+\beta)} \right\rceil$. *If* $n \geq \frac{\alpha_{\beta,\epsilon}+1}{p_\beta}$, *it holds*

$$\Pr\left(Z_n > \frac{1}{2}\right) > 1 - e^{-\frac{2p_\beta^2}{n}\left(n - \frac{\alpha_{\beta,\epsilon}}{p_\beta}\right)^2}.$$

Proof. In order to leverage Proposition 18, we partition the interval $[0,1]$ as follows:

$$\begin{aligned}
I_0 &= (0, \epsilon] \\
I_i &= \left(\epsilon\left(1+\beta\right)^{i-1}, \epsilon\left(1+\beta\right)^{i}\right] \quad \text{for } i \in \{1, ..., \alpha_{\beta,\epsilon} - 1\} \\
I_{\alpha_{\beta,\epsilon}} &= \left(\epsilon\left(1+\beta\right)^{\alpha_{\beta,\epsilon}-1}, \frac{1}{2}\right] \\
I_{\alpha_{\beta,\epsilon}+1} &= \left(\frac{1}{2}, 1\right]
\end{aligned}$$

By the law of total probability, it then directly follows from Proposition 18 that

$$\Pr\left(Z_{n+1} \in I_{i+1} \,|\, Z_n \in I_i\right) \geq p_\beta.$$

Hence, defining the geometric random variables $Z_i^{(\uparrow)} \sim \text{Geom}\left(p_\beta\right)$, we can consider a coupling such that, for any k,

$$Z_i^{(\uparrow)} = k \implies \text{“}Z_{n+k} \in \left(\bigcup_{j=i+1}^{\alpha_{\beta,\epsilon}+1} I_{i+1}\right) \mid Z_n \in I_i\text{”},$$

that is

$$\sum_{i=0}^{\alpha_{\beta,\epsilon}+1} Z_i^{(\uparrow)} \leq n \implies Z_n > \frac{1}{2}. \tag{3.9}$$

From Eq. 3.9, we can stochastically dominate the original process:

$$\Pr\left(Z_n \leq \frac{1}{2}\right) \leq \Pr\left(\sum_{i=0}^{\alpha_{\beta,\epsilon}} Z_i^{(\uparrow)} > n\right). \tag{3.10}$$

We can bound the r.h.s. of Eq. 3.10 using a standard coupling between a sum of geometric i.i.d. random variable and an appropriate binomial distribution, which can be expressed with the following abuse of notation

$$\begin{aligned}\Pr\left(\sum_{i=0}^{\alpha_{\beta,\epsilon}+1} Z_i^{(\uparrow)} > n\right) &= \Pr\left(\sum_{i=0}^{\alpha_{\beta,\epsilon}+1} \mathrm{Geom}_i\left(p_\beta\right) > n\right). \\ &= \Pr\left(\mathrm{Bin}\left(n, p_\beta\right) \leq \alpha_{\beta,\epsilon}+1\right)\end{aligned} \tag{3.11}$$

We now recall a Chernoff-Hoeffding bound, which are going to apply next.

Theorem 21 (Chernoff–Hoeffding bound [DP09])**.** *Let $X = \sum_{i=1}^{n} X_i$ where the X_i are i.i.d. in $[0,1]$. For all $t > 0$ and $\epsilon > 0$, it holds*

$$\Pr\left(X < \mathbb{E}\left[X\right] - t\right) \leq e^{-\frac{2t^2}{n}}.$$

By combining Eq. 3.10 and Eq. 3.11, and using Theorem 21 with $\mathbb{E}\left[X\right] - t = np_\beta - t = \alpha_{\beta,\epsilon} + 1$, we get

$$\begin{aligned}\Pr\left(Z_n \leq \frac{1}{2}\right) &\leq \Pr\left(\mathrm{Bin}\left(n, p_\beta\right) \leq \alpha_{\beta,\epsilon}+1\right) \\ &\leq e^{-\frac{2\left(np_\beta - \alpha_{\beta,\epsilon} - 1\right)^2}{n}},\end{aligned} \tag{3.12}$$

which holds as long as $np_\beta - \alpha_{\beta,\epsilon} - 1 > 0$, namely $n > \frac{\alpha_{\beta,\epsilon}+1}{p_\beta}$. The thesis follows by considering the complementary event $Z_n > \frac{1}{2}$ in Eq. 3.12. □

Proposition 20 concludes the analysis of a *first phase* of the process, in which the expected growth of Z_n is exponential. The exponential increase is allowed by the fact that a relatively large fraction of not-yet-approximable values (i.e. $1 - Z_n$) can become approximable, but such an assumption ceases to apply when Z_n becomes large. This is a common situation in the analysis of analogous stochastic processes in other contexts, such as the

analysis of opinion dynamics [BCN20a], and the key observation is that when the exponential growth is coming to an end, an exponential decay phase for the complementary variable $1 - Z_n$ is about to begin, as we will see in the next section.

3.2 Exponential Decrease Phase

Our analysis will now deal with a *second phase* of the process, in which the complement of Z_n, i.e. the quantity $\bar{Z}_n = 1 - Z_n$ (namely the fraction of values that are not ϵ-approximable) shrinks exponentially fast in expectation.

Corollary 22 (of Proposition 16)**.** *It holds*

$$\mathbb{E}\left[\bar{Z}_{n+1} \,\middle|\, X_1, ..., X_n\right] \leq \bar{Z}_n \left(1 + \frac{1}{4}\left(1 - \bar{Z}_n\right)\right).$$

Proof. We have

$$\begin{aligned}
&\mathbb{E}\left[\bar{Z}_{n+1} \,\middle|\, X_1, ..., X_n\right] \\
&= 1 - \mathbb{E}\left[Z_{n+1} \,|\, X_1, ..., X_n\right] \\
&\qquad \text{from Proposition 16} \\
&\leq 1 - Z_n \left(1 + \frac{1}{4}\left(1 - Z_n\right)\right) \\
&= \bar{Z}_n - \frac{1}{4}\bar{Z}_n\left(1 - \bar{Z}_n\right).
\end{aligned}$$

□

In contrast to Proposition 20, translating Corollary 22 into a probability bound is relatively straightforward.

Lemma 23. *Let τ be such that $Z_\tau > \frac{1}{2}$, then for $n > \tau$ it holds*

$$\Pr\left(Z_n \leq 1 - \frac{\epsilon}{2} \,\middle|\, Z_\tau > \frac{1}{2}\right) \leq \left(\frac{7}{8}\right)^{n-\tau}.$$

Proof. For any $\tau < n$, it holds

$$\begin{aligned}
&\mathbb{E}\left[\bar{Z}_n \,\middle|\, \bar{Z}_\tau \leq \frac{1}{2}\right] \\
&\qquad \text{by the law of iterated expectation}
\end{aligned}$$

$$
\begin{aligned}
&= \mathbb{E}\left[\mathbb{E}\left[\bar{Z}_n \,\middle|\, X_1, ..., X_{n-1}, \bar{Z}_\tau \leq \frac{1}{2}\right]\right] \\
&\quad \text{by Corollary 22} \\
&\leq \frac{7}{8}\mathbb{E}\left[\bar{Z}_{n-1} \,\middle|\, \bar{Z}_\tau \leq \frac{1}{2}\right]. \qquad (3.13)
\end{aligned}
$$

Hence

$$
\begin{aligned}
&\Pr\left(Z_n \leq 1 - \frac{\epsilon}{2} \,\middle|\, Z_\tau > \frac{1}{2}\right) \\
&= \Pr\left(\bar{Z}_n \geq \frac{\epsilon}{2} \,\middle|\, \bar{Z}_\tau \leq \frac{1}{2}\right) \\
&\quad \text{by Markov's inequality} \\
&\leq 2\frac{\mathbb{E}\left[\bar{Z}_n \,\middle|\, \bar{Z}_\tau \leq \frac{1}{2}\right]}{\epsilon} \\
&\quad \text{by Eq. 3.13} \\
&\leq \left(\frac{7}{8}\right)^{n-\tau} 2\mathbb{E}\left[\bar{Z}_\tau \,\middle|\, \bar{Z}_\tau \leq \frac{1}{2}\right] \\
&\leq \left(\frac{7}{8}\right)^{n-\tau}.
\end{aligned}
$$

□

We can finally combine Lemma 23 and Proposition 20 to prove Theorem 6.

Proof of Theorem 6. It holds

$$
\begin{aligned}
&\Pr\left(Z_n > 1 - \frac{\epsilon}{2}\right) \\
&\geq \Pr\left(Z_n > 1 - \frac{\epsilon}{2} \,\middle|\, Z_k > \frac{1}{2}\right) \Pr\left(Z_k > \frac{1}{2}\right) \\
&\quad \text{by Lemma 23 and Proposition 20} \\
&\geq \left(1 - \left(\frac{7}{8}\right)^{n-k}\right)\left(1 - e^{-\frac{2p_\beta^2}{k}\left(k - \frac{\alpha_{\beta,\epsilon}}{p_\beta}\right)^2}\right) \\
&\quad \text{for } n, k \geq C_\beta \log\frac{1}{2\epsilon} \text{ for a large enough } C_\beta \\
&\geq \left(1 - \frac{\epsilon}{2}\right)\left(1 - \frac{\epsilon}{2}\right)
\end{aligned}
$$

$$\geq 1 - \epsilon.$$

□

Chapter 4

The Strong Lottery Ticket Hypothesis for Convolutional Networks

> [T]he set of possible people allowed by our DNA so massively outnumbers the set of actual people. In the teeth of these stupefying odds it is you and I, in our ordinariness, that are here. We privileged few, who won the lottery of birth against all odds[.] - Richard Dawkins [Daw06]

In this chapter, we are going to apply the results on RSS that we have discussed and re-proved in chapters 2 3 to a seemingly unrelated problem in the theory of artificial neural networks (ANNs), known as the Strong Lottery Ticket Hypothesis (SLTH). In particular, after recalling the history of the SLTH and briefly surveying related results, we are going to provide a simplified version of our original contribution to the topic, namely a proof of the SLTH in the case of Convolutional Neural Networks (CNNs) [dCNV22]. Specifically, compared to the original version [dCNV22], this chapter provides a more streamlined proof of one of the main technical ingredients, namely Proposition 32.

4.1 The SLTH

The SLTH is a *strong* version of the Lottery Ticket Hypothesis (LTH), a statement motivated by empirical observations by [FC19]. The LTH is closely related to one of the simplest compression strategies for neural networks,

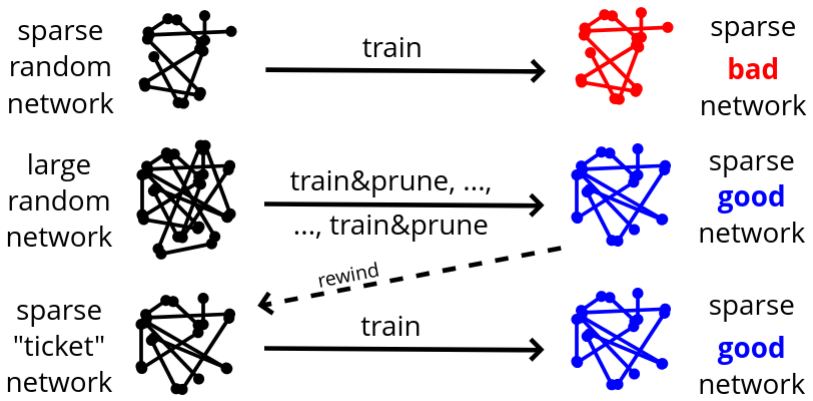

Figure 4.1: Diagram illustrating key facts related to the LTH. While sparse networks do not generally achieve good accuracy when trained, IMP succeeds in creating a sparse network that performs well. A variant of IMP "with rewind" allows to identify *lottery tickets*, i.e. sparse subnetworks that, when re-initialized with the original weights, successfully train to reach good accuracy.

i.e. Iterative Magnitude Pruning (IMP), in which edges of the network are progressively removed (pruned) during training (Figure 4.2). Despite its simplicity, IMP is still nowadays one of the most effective strategies for reducing the number of parameters of ANNs [BGOFG20]. Before explaining the experiments in [FC19] on a variant of IMP that motivated the LTH, let us first informally recall a key empirical fact on ANNs (see also Figure 4.1): generally speaking, training sparse architectures with relatively few parameters doesn't work, in the sense that it doesn't allow to reach good accuracy; training large architectures with many parameters, on the contrary, generally succeeds in reaching good accuracy; on the other hand IMP allows to reach the accuracy of large architectures while producing, in the end, a sparse network. In this context, [FC19] considers a variant of IMP with rewind, in which after each pruning step the weights of the remaining edges are re-initialized to their original value before training. This way, they manage to obtain sparse networks that successfully train to good accuracy. The success of their approach lead them to formulate the LTH:

> "Dense, randomly-initialized, feed-forward networks contain subnetworks (winning tickets) that—when trained in isolation reach test accuracy comparable to the original network in a similar

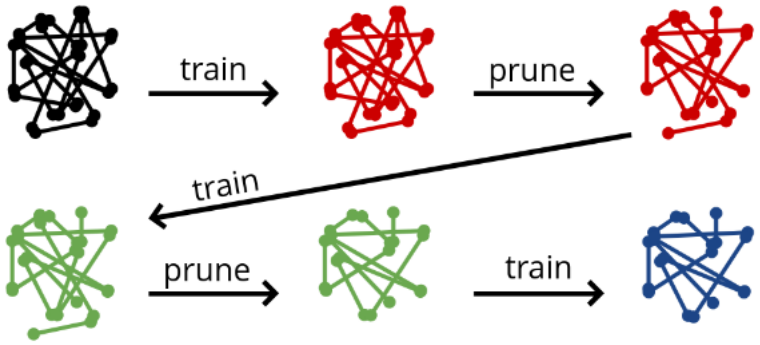

Figure 4.2: Schematic representation of Iterative Pruning techniques, in which an initial network is repeatedly trained according to some schedule (e.g. a predetermined number of epochs or until reaching some accuracy) and then *pruned* according to some rule (e.g. the edges with the smallest absolute value are removed).

number of iterations."

Let us now establish some notation to rephrase the LTH and prepare the ground for introducing its stronger version, the SLTH.

Let us denote by N_0 a dense randomly initialized network with 2ℓ layers[1], by N_L a subnetwork obtained by pruning N_0 and by N_T the best ℓ-layer *target* network for the task at hand among those which are *sufficiently smaller* than N_0 (see also Figure 4.3).

We can rephrase the LTH as follows:

> Given a network N_T, a sufficiently large network N_0 contains a subnetwork N_L such that the accuracy of N_L after training is at least as good as that of N_T.

Later works then explored an even more extreme case of pruning, by introducing methods to sparsify the network *as a training strategy*. Specifically, [ZLLY19] presents an algorithm that learns an associated probability p for each weight w in the network. On the forward pass, they include the weight w with probability p and otherwise zero it out. Equivalently, they use weight˜ $w = wX$ where X is a Bernoulli(p) random variable. The probabilities p

[1]In the dense network case, the 2ℓ layers of N_0 can be reduced to $\ell + 1$ as shown by [Bur22]. Here we keep the factor 2 since it is not clear whether the argument carries on to the CNN case we are going to consider in this chapter.

are the output of a sigmoid, and are learned through stochastic gradient descent. Subsequently, [RWK$^+$20] proposed a more accurate pruning-without-training strategy with the EDGE-POPUP algorithm, which assigns to each edge (u, v) a *score* s_{uv} and deterministically select the top-k edges in the network with the highest scores. The scores are updated naively with gradient descent: ignoring momentum and weight decay, $s_{uv}^{(t+1)} = s_{uv}^{(t)} - \alpha \frac{\partial \mathcal{L}}{\partial \mathcal{I}_v^{(t)}} w_{uv} \mathcal{Z}_u^{(t)}$ where $\mathcal{I}_v$ is the input of node v, $\mathcal{Z}_u$ is the output of node u (i.e. $\mathcal{Z}_u = \sigma(\mathcal{I}_u)$), w_{uv} is the weight of the edge (u, v) (which doesn't change), α is the step size of the gradient descent and $\mathcal{L}$ is the loss function. [RWK$^+$20] formally proves that the above pruning rule and score update do indeed optimize the loss across mini-batches.

The above works thus motivated a stronger version of the LTH, the **Strong** Lottery Ticket Hypothesis (SLTH):

> Given a network N_T, a sufficiently large network N_0 contains a subnetwork N_L such that the accuracy of N_L is approximately the same as that of N_T.

From a theoretical point of view, the SLTH presents the big advantage of being easier to turn in a quantitative statement, since training is not involved (see Figure 4.3). The SLTH thus spawned various theoretical results.

The first, [MYSSS], presented a way to prune random feed-forward dense neural networks that was progressively refined across subsequent works.

Intuitively, if we would like to obtain a certain network N_T, we can imaging generating a random network N_0 which is identical in structure to N_T, except for having M random edges between each pair of neurons (i.e. being a random multi-graph); for a large enough value of M, there would then be a large enough probability that, for each edge, one of the M random edges would be close enough to the weight of the corresponding edge in N_T.

The construction by [MYSSS], represented in Figure 4.4, can be seen as a way to simulate the previous idea, by essentially pre-prune the random network N_0 so that each pair of pairs of layers $(2\ell, 2\ell - 1)$ and $(2\ell - 1, 2\ell)$ correspond to a pair of layers $(\ell, \ell + 1)$ of an ideal target network N_T that we wish to approximate, and each weight w between ℓ and $\ell + 1$ corresponds to a *gadget* shown in part (c) of Figure 4.4. The idea is then to further prune the aforementioned gadgets to approximate w. In subfigure (c), it is also hinted that [MYSSS] leverage the identity $x = \sigma(x) - \sigma(-x)$ in order to approximate the output xw for any sign of x and w, when intermediate neurons in the construction are subject to ReLU activations.

In [MYSSS], the above construction follows the above intuition of looking

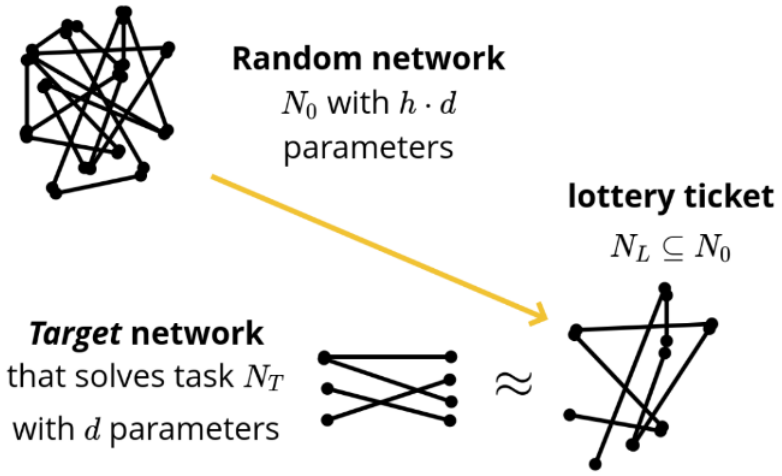

Figure 4.3: Diagram of the relation between the initial random network N_0, the target network N_T and the lottery ticket subnetwork N_L.

for a single random edge that ϵ-approximate w, which then requires that N_0 is larger than N_T by a factor of order $\frac{1}{\epsilon}$.

[PRN$^+$20] and, independently, [OHR], improved the factor $\frac{1}{\epsilon}$ in [MYSSS] to $\log \frac{1}{\epsilon}$, which can be seen to be tight by a packing argument. In particular, when considering the construction in [MYSSS], [PRN$^+$20] recognized that if one aims at combining several random weights instead of picking just the best one, then the problem can be traced back to the famous Random Subset Sum Problem that we discussed in detail in the previous chapters.

Later work generalized the class of functions for which the result of [PRN$^+$20] holds and, for approximating an ℓ-layer network N_T, reduced the number of layers of the random network N_0 from 2ℓ to $\ell+1$ [BLMG22, FB21].

This chapter is devoted to our contribution to the SLTH, namely the first proof for CNNs. Next, we provide an informal version of our result.

Theorem 24 (Informal version of Theorem 31)**.** *Let $\epsilon > 0$, and N_T be any CNN with k parameters, ℓ layers, and such that all its filters have ℓ_1 norm at most 1; finally, let N_0 be a random CNN with $O\left(k \log \frac{k\ell}{\epsilon}\right)$ parameters and 2ℓ layers. With probability $1-\epsilon$, we can approximate N_T within an error ϵ by suitably pruning N_0.*

The rest of this chapter presents a revised version of the proof of 24, originally given in [dCNV22].

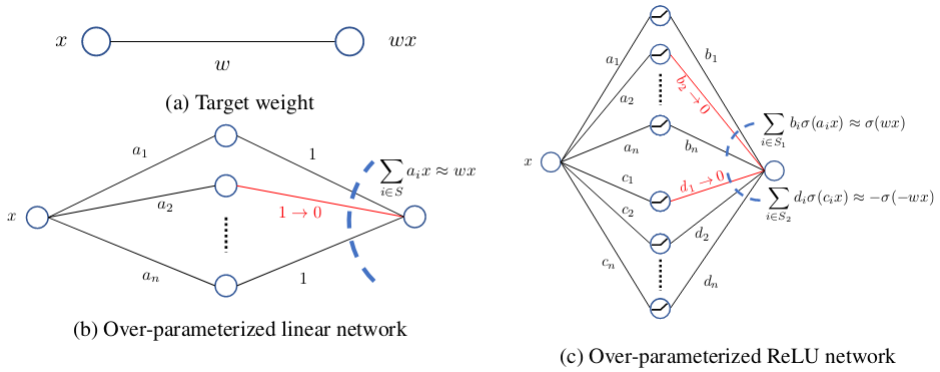

Figure 4.4: Diagram by Pensia et al. (2020) illustrating the construction by Malach et al. (2020) (see Section 4.1). (a) Suppose that there exists a target network N_T that we wish to approximate (see also Figure 4.3), and let w be any weight of such network. (b) Consider a random network N_0 (again, see Figure 4.3), such that the layers ℓ and $\ell + 1$ at the endpoints of w correspond to layers $2\ell - 2$ and 2ℓ of N_0, so that there is an intermediate layer $2\ell - 1$ in N_0 with many intermediate neurons; if we assume that the activation of the intermediate neurons is the identity and all weights between layer $2\ell - 1$ and 2ℓ are 1, we can prune edges between 2ℓ and $2\ell - 1$, and between $2\ell - 1$ and 2ℓ, so that the resulting topology computes $\sum_{i \in S} a_i x$ where a_i is the weight of the remaining edges connecting the left neuron to a set of intermediate neurons Ω, and S is the subset of intermediate neurons Ω which are connected to the right neuron. (c) In the previous construction, we can remove the assumption of weight 1 between layers $2\ell - 1$ and 2ℓ and further assume that intermediate neurons have ReLU activations.

4.2 CNN notation and definitions

Let σ be the ReLU activation function, i.e. $\sigma(x) = \max\{0, x\}$. We say that a tensor is *non-negative* if all its entries are non-negative. Given any tensor K, we denote with $\|K\|_{max}$ the maximum absolute value of all entries of K,namely $\|K\|_{max} = \max_{\vec{i} \in \text{indices}(K)} \left|K_{\vec{i}}\right|$. We often make use of the *slice* notation ":", which indicates that the full range of the index where the colon appears has to be considered; for example, if $U \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times d_4}$, then for any $i_3 \in d_3$ we denote with $U_{:,:,i,:}$ the 3-dimensional tensor in $\mathbb{R}^{d_1 \times d_2 \times d_3 \times d_4}$ such that

$$\left(U_{:,:,i_3,:}\right)_{i_1,i_2,i_4} = U_{i_1,i_2,i_3,i_4}. \tag{4.1}$$

Since we are interested in randomly-initialized CNNs, it will be handy to formally define the notion of a tensor with uniformly random entries.

Definition 25 (Uniform tensor)**.** We say that $U \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times d_4}$ is a *uniform tensor* and write $U \sim \text{Unif}\left([-1,1]\right)^{d_1 \times d_2 \times d_3 \times d_4}$ if its entries are i.i.d. with distribution $\text{Unif}\left([-1,1]\right)$.

The SLTH being about pruning, we also need to define the notion of a mask, which allows us to represent pruning algebraically via component-wise multiplication between tensors.

Definition 26 (Tensor mask)**.** A *mask* S of a tensor L is a binary tensor with the same size as L.

Next, we define the two main operations used when deadline with convolutional neural networks: the aforementioned component-wise multiplication of tensors and discrete (finite) convolution.

Definition 27 (Component-wise multiplication $\odot$)**.** The component-wise multiplication (also known as Hadamard product) of two tensors L and L', denoted with $L \odot L'$, is defined as the component-wise product of the two tensors. In formulas, given indices I

$$\left(L \odot L'\right)_I = \left(L\right)_I \cdot \left(L'\right)_I.$$

Given two tensors $K \in \mathbb{R}^{d \times d \times c}$ and $X \in \mathbb{R}^{D \times D \times c}$, their discrete convolution is the $D \times D$ matrix such that, for each $i, j \in [D]$

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

where all missing entries of X are assumed to be 0, i.e. X is zero-padded. Analogously, when $K \in \mathbb{R}^{d\times d\times c_0\times c_1}$ and $X \in \mathbb{R}^{D\times D\times c_0}$, $K * X$ is the $D \times D \times c_1$ tensor such that, for each $i, j \in [D]$ and $\ell \in [c_1]$

$$(K * X)_{i,j,\ell} = \sum_{i',j'\in[d],k\in[c_0]} K_{i',j',k,\ell} \cdot X_{i-i'+1,j-j'+1,k}. \tag{4.2}$$

For convenience, we often make use of the *slice* notation (Eq. 4.1) when referring to the output of a 4-dimensional convolution; for example, for each ℓ we can rewrite Eq. 4.2 as

$$(K * X)_{:,:,\ell} = K_{:,:,:,\ell} * X.$$

Next, we formally define the type of convolutional neural networks (CNNs) which we consider in our proofs. Compared to classical CNNs, the convolutions have no bias and are suitably padded with zeros.

Definition 28 (Simple CNN)**.** A simple CNN $N : [0,1]^{D\times D\times c_0} \to \mathbb{R}^{D\times D\times c_\ell}$ is a neural network that it can be written as

$$N(X) = K^{(\ell)} * \sigma\left(K^{(\ell-1)} * \sigma\left(\cdots * \sigma\left(K^{(1)} * X\right)\right)\right)$$

where $K^{(i)} \in \mathbb{R}^{d_i\times d_i\times c_{i-1}\times c_i}$ for $i \in [\ell]$ and for some $c_0, ..., c_\ell$ and $d_1, ..., d_\ell$ are the network *filters*.

We call a random simple CNN a simple CNN such that $K^{(i)} \sim \text{Unif}\left([-1,1]\right)^{d_i\times d_i\times c_{i-1}\times c_i}$.

Remark 29. The restrictions on tensor sizes and the exclusion of bias terms are for the sake of simplicity. The proof we provide also works with biases, yielding an equivalent RSS problem, with the only modification of replacing d_i^2 terms with $d_i^2 + 1$ terms.

Finally, since the terms kernel and filter are not consistently used in the literature, we provide a formal definition of our convention.

Definition 30 (Kernels and filters)**.** Given a simple CNN $K^{(\ell)} * \sigma\left(K^{(\ell-1)} * \sigma\left(\cdots * \sigma\left(K^{(1)} * X\right)\right)\right)$, we call a *filter* each tensor $K^{(i)} \in \mathbb{R}^{d_i\times d_i\times c_{i-1}\times c_i}$ $(i \in [\ell])$, and each slice $K^{(i)}_{:,:,t,k}$ $(t \in [c_{i-1}], k \in [c_i])$ a *kernel*.

4.3 Proof of the SLTH for CNNs

We start the section by providing the full rigorous statement of Theorem 24.

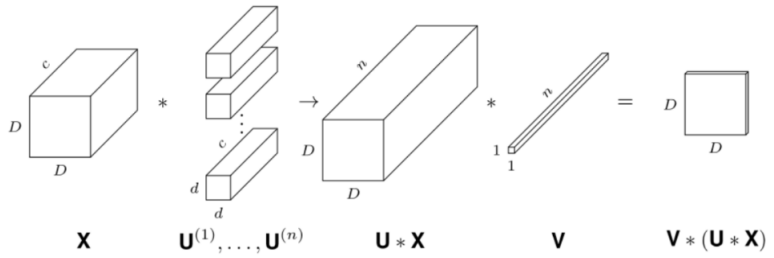

Figure 4.5: Scheme of the random CNN employed in Theorem 31.

Theorem 31. *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 $L^{(2i-1)} \sim \text{Unif}\left([-1,1]\right)^{d_i \times d_i \times c_{i-1} \times n_i}$ and $L^{(2i)} \sim \text{Unif}\left([-1,1]\right)^{1 \times 1 \times n_i \times c_i}$ with $n_i \geq C c_i \log \frac{c_{i-1} c_i d_i^2 \ell}{\epsilon}$ for some positive integers n_i and c_i. Let then N_0 be a random CNN of the form*

$$N_0\left(X\right) = L^{(2\ell)} * \sigma\left(\cdots \sigma\left(L^{(1)} * X\right)\right).$$

Given any mask $S^{(i)}$ for each tensor $L^{(i)}$, let

$$N_0^{\left(S^{(1)},\ldots,S^{(2\ell)}\right)} = \left(S^{(2\ell)} \odot L^{(2\ell)}\right) * \sigma\left(\cdots \sigma\left(\left(S^{(1)} \odot L^{(1)}\right) * X\right)\right)$$

be the CNN resulting from pruning N_0 by applying mask $S^{(i)}$ to each tensor $L^{(i)}$. Finally, let $\mathcal{F}$ be the class of functions $f : [0,1]^{D \times D \times c_0} \to \mathbb{R}^{D \times D \times c_\ell}$ which can be written in the form

$$f\left(X\right) = K^{(\ell)} * \sigma\left(\cdots \sigma\left(K^{(1)} * X\right)\right)$$

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

There exists a universal value of C such that, with probability $1 - \epsilon$, for every $f \in \mathcal{F}$

$$\inf_{\substack{S^{(1)},\ldots,S^{(2\ell)} \\ \forall i, S^{(i)} \in \{0,1\}^{\text{size}\left(L^{(i)}\right)}}} \sup_{X \in [-1,1]^{D \times D \times c_0}} \left\| f\left(X\right) - N_0^{\left(S^{(1)},\ldots,S^{(2\ell)}\right)}\left(X\right) \right\|_{max} \leq \epsilon.$$

We break the proof down into three sections: Section 4.3.1 where we bound the approximation error for a single filter, Section 4.3.2 where we bound the approximation error over an entire layer, and finally Section 4.3.3 where we conclude the proof by combining the previous results.

4.3.1 Approximation of a filter

The main ingredient for proving Theorem 31 is provided by the following lemma, which shows how to approximate a convolution with a single filter using convolutions with random filters after suitably pruning them.

Proposition 32 (Unstructured filter pruning)**.** *Let $D, d, c, n \in \mathbb{N}$ be positive integers and $\epsilon, C \in \mathbb{R}_{>0}$ with $n \geq C \log \frac{d^2 c}{\epsilon}$, and let $U \sim \text{Unif}\left([-1,1]\right)^{d\times d\times c\times n}$, $V \sim \text{Unif}\left([-1,1]\right)^{1\times 1\times n\times 1}$ and $S \in \{0,1\}^{\text{size}(U)}$. We define $N_0\left(X\right) = V * \sigma\left(U * X\right)$ where $X \in [0,1]^{D\times D\times c}$, and its pruned version $N_0^{(S)}\left(X\right) = V * \sigma\left((U \odot S) * X\right)$. There is a universal constant C such that, with probability $1-\epsilon$, for all $K \in [-1,1]^{d\times d\times c\times 1}$ with $\|K\|_1 \leq 1$, there exists an S such that*

$$\sup_{X\in[0,1]^{D\times D\times c}} \left\| K * X - N_0^{(S)}\left(X\right)\right\|_{max} < \epsilon.$$

To prove Proposition 32, we use the following inequality.

Proposition 33 (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*

$$\|K * X\|_{max} \leq \|K\|_1 \cdot \|X\|_{max}.$$

Proof. We have

$$\begin{aligned}
&\|K * X\|_{max} \\
&\leq \max_{i,j\in[D],\ell\in[c_1]} \sum_{i',j'\in[d],k\in[c]} \left|K_{i',j',k,\ell} X_{i-i'+1,j-j'+1,k}\right| \\
&\leq \max_{i,j\in[D],\ell\in[c_1]} \left(\sum_{i',j'\in[d],k\in[c]} \left|K_{i',j',k,\ell}\right|\right) \|X\|_{max} \\
&\leq \max_{i,j\in[D],\ell\in[c_1]} \|K\|_1 \cdot \|X\|_{max} \\
&= \|K\|_1 \cdot \|X\|_{max}.
\end{aligned}$$

□

We can now proceed with the proof of Proposition 32.

Proof of Proposition 32. The first step of the proof is to get rid of the non-linearity so that we can work with $V * (U * X)$ rather than $V * \sigma\left(U * X\right)$. Given that $\sigma\left(x\right) = x$ for all positive x, this is easily achieved by pruning

all negative entries of U, yielding $U^+ = \max.\{0, U\}$ where max. denotes an entry-wise application of max. Since the input tensor X is non-negative, we then have

$$V * \sigma\left(U^+ * X\right) = V * \left(U^+ * X\right). \tag{4.3}$$

We then compute

$$\begin{aligned}
&\left(V * \left(U^+ * X\right)\right)_{r,s,1} \\
&= \sum_{t=1}^{n} V_{1,1,t,1} \cdot \left(U^+ * X\right)_{r,s,t} \\
&= \sum_{t=1}^{n} V_{1,1,t,1} \cdot \left(\sum_{i,j\in[d],k\in[c]} U^+_{i,j,k,t} \cdot X_{r-i+1,s-j+1,k} \right)_{r,s,t} \\
&= \sum_{t=1}^{n} \sum_{i,j\in[d],k\in[c]} \left(V_{1,1,t,1} \cdot U^+_{i,j,k,t}\right) \cdot X_{r-i+1,s-j+1,k} \\
&= \sum_{i,j\in[d],k\in[c]} \left(\sum_{t=1}^{n} V_{1,1,t,1} \cdot U^+_{i,j,k,t} \right) \cdot X_{r-i+1,s-j+1,k} \\
&\qquad \text{defining } L_{i,j,k,1} = \sum_{t=1}^{n} V_{1,1,t,1} \cdot U^+_{i,j,k,t} \\
&= \sum_{i,j\in[d],k\in[c]} L_{i,j,k,1} \cdot X_{r-i+1,s-j+1,k}
\end{aligned}$$

which shows that $V * (U^+ * X)$ is equivalent to $L * X$ for a suitable $L \in \mathbb{R}^{d\times d\times c\times 1}$. We now observe that, for each tuple of indices (i, j, k), we can control the value of $L_{i,j,k,1}$ by suitably pruning the entries of the vector $U^+_{i,j,k,:}$. In fact, for each target value $K_{i,j,k,1} \in [-1, 1]$, the problem of pruning $U^+_{i,j,k,:}$ so that $|K_{i,j,k,1} - L_{i,j,k,1}| < \epsilon$ is an instance of RSS.

More precisely, we can define the event

$$\mathcal{E}^{(\text{filter})}_{i,j,k,1} = \text{“}\forall z \in [-1, 1], \exists S \subseteq [n] : \left| z - \sum_{t\in S} V_{1,1,t,1} \cdot U^+_{i,j,k,t} \right| < \epsilon\text{”}$$

and their union $\mathcal{E}^{(\text{filter})} = \bigcap_{i,j,k\in[d]} \mathcal{E}^{(\text{filter})}_{i,j,k,1}$. By Corollary 11, for every i, j, k and t, the random variable $V_{1,1,t,1} \cdot U^+_{i,j,k,t}$ is $\left(\frac{1}{2}, \frac{\log 2}{2}\right)$-super-uniform. By Corollary 10 and the hypothesis $n \geq C \log \frac{d^2 c}{\epsilon}$, it thus follows that

$\Pr\left(\bar{\mathcal{E}}^{(\text{filter})}_{i,j,k,1}\right) \leq \frac{\epsilon}{d^2 c}$, where $\bar{\mathcal{E}}^{(\text{filter})}_{i,j,k,1}$ denotes the complement of $\mathcal{E}^{(\text{filter})}_{i,j,k,1}$. By a union bound, we then see that

$$\begin{aligned}
&\Pr\left(\mathcal{E}^{(\text{filter})}\right) \\
&= 1 - \Pr\left(\bigcup_{i,j,k\in[d]} \bar{\mathcal{E}}^{(\text{filter})}_{i,j,k,1}\right) \\
&\geq 1 - \sum_{i,j,k\in[d]} \Pr\left(\bar{\mathcal{E}}^{(\text{filter})}_{i,j,k,1}\right) \\
&\geq 1 - \sum_{i,j,k\in[d]} \frac{\epsilon}{d^2 c} \\
&\geq 1 - \epsilon. \qquad (4.4)
\end{aligned}$$

Thus, when $\mathcal{E}^{(\text{filter})}$ holds, we have

$$\sup_{K\in[-1,1]^{d\times d\times 1\times 1}} \inf_{S\in\{0,1\}^{\text{size}(U)}} \left\|K - V * (U^+ \odot S)\right\|_{max} < \frac{\epsilon}{d^2 c}. \qquad (4.5)$$

It follows that, when $\mathcal{E}^{(\text{filter})}$ holds (which, by Eq. 4.4, happens with probability $1-\epsilon$), we can compute

$$\begin{aligned}
&\sup_{K\in[-1,1]^{d\times d\times 1\times 1}} \inf_{S\in\{0,1\}^{\text{size}(U)}} \sup_{X\in[0,1]^{D\times D\times c}} \left\|K * X - N_0^{(S)}(X)\right\|_{max} \\
&= \sup_{K\in[-1,1]^{d\times d\times 1\times 1}} \inf_{S\in\{0,1\}^{\text{size}(U)}} \sup_{X\in[0,1]^{D\times D\times c}} \|K * X - V * \sigma((U \odot S) * X)\|_{max} \\
&\quad \text{restricting } U \text{ to positive entries} \\
&= \sup_{K\in[-1,1]^{d\times d\times 1\times 1}} \inf_{S\in\{0,1\}^{\text{size}(U)}} \sup_{X\in[0,1]^{D\times D\times c}} \left\|K * X - V * \sigma\left((U^+ \odot S) * X\right)\right\|_{max} \\
&\quad \text{by a slight adaptation of Eq. 4.3} \\
&= \sup_{K\in[-1,1]^{d\times d\times 1\times 1}} \inf_{S\in\{0,1\}^{\text{size}(U)}} \sup_{X\in[0,1]^{D\times D\times c}} \left\|K * X - V * \left((U^+ \odot S) * X\right)\right\|_{max} \\
&\quad \text{from the distributivity property of convolution} \\
&= \sup_{K\in[-1,1]^{d\times d\times 1\times 1}} \inf_{S\in\{0,1\}^{\text{size}(U)}} \sup_{X\in[0,1]^{D\times D\times c}} \left\|\left(K - V * (U^+ \odot S)\right) * X\right\|_{max} \\
&\quad \text{by the Tensor Convolution Inequality (Proposition 33)} \\
&= \sup_{K\in[-1,1]^{d\times d\times 1\times 1}} \inf_{S\in\{0,1\}^{\text{size}(U)}} \sup_{X\in[0,1]^{D\times D\times c}} \left\|K - V * (U^+ \odot S)\right\|_1 \cdot \|X\|_{max} \\
&\quad \text{since } \|X\|_{max} \leq 1
\end{aligned}$$

$$
\begin{aligned}
&= \sup_{K \in [-1,1]^{d \times d \times 1 \times 1}} \inf_{S \in \{0,1\}^{\text{size}(U)}} \left\| K - V * (U^+ \odot S) \right\|_1 \\
&\quad \text{since } K - V * (U^+ \odot S) \text{ has } d^2 c \text{ entries} \\
&= d^2 c \sup_{K \in [-1,1]^{d \times d \times 1 \times 1}} \inf_{S \in \{0,1\}^{\text{size}(U)}} \left\| K - V * (U^+ \odot S) \right\|_{max} \\
&\quad \text{by Eq. 4.5} \\
&= d^2 c \frac{\epsilon}{d^2 c} \\
&= \epsilon,
\end{aligned}
$$

proving the thesis. □

4.3.2 Approximation of a convolution layer

The next lemma extends the approximation provided by Proposition 32 from a single filter to an entire convolution layer.

Lemma 34 (Layer-wise approximation)**.** *Let $D, d, c_0, c_1, n \in \mathbb{N}$ and $\epsilon', C \in \mathbb{R}_{>0}$ with $n \geq C c_1 \log \frac{d^2 c_0 c_1}{\epsilon'}$, $U \sim \text{Unif}\left([-1,1]\right)^{d \times d \times c_0 \times n}$ and $V \sim \text{Unif}\left([-1,1]\right)^{1 \times 1 \times n \times c_1}$. Let $N_0 : [0,1]^{D \times D \times c_0} \to \mathbb{R}^{D \times D \times c_1}$ be the random 2-layer CNN*

$$N_0(X) = V * \sigma(U * X),$$

and given any two masks S and T for U and V respectively, let

$$N_0^{(S,T)}(X) = (V \odot T) * \sigma((U \odot S) * X).$$

There exists a value for the above constant C such that, independently from all other parameters, with probability $1-\epsilon$ it holds that for all $K \in [-1,1]^{d \times d \times c_0 \times c_1}$ there exist masks S and T which satisfy

$$\sup_{X \in [0,1]^{D \times D \times c_0}} \left\| K * X - N_0^{(S,T)}(X) \right\|_{max} \leq \epsilon'.$$

Proof. Intuitively, to approximate each output channel of a given layer, we are going to prune the second convolution filter V of N_0 in a way that allows us to treat each output channel independently. Specifically, we choose T to be the block diagonal matrix with c_1 blocks of size $n' = \frac{n}{c_1}$, namely

$$(T)_{1,1,t,\ell} = \begin{cases} 1 & \text{if } (\ell - 1) n' \leq t \leq \ell n', \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

The rest of the proof is devoted to showing how to choose S, which we do independently for each kernel in an analogous way to the proof of Proposition 32. For each $\ell \in [c_1]$ let us denote $K^{(\ell)}$ the ℓ-th kernel of K, namely$K^{(\ell)} = K_{:,:,:,\ell}$. Moreover, as we show shortly, the block structure of T given in Eq. 4.6 motivates the definition of the following sub-tensors

$$U^{(\ell)} = U_{:,:,:,(\ell-1)n'<t\leq \ell n'}, \qquad S^{(\ell)} = S_{:,:,:,(\ell-1)n'<t\leq \ell n'}, \qquad V^{(\ell)} = V_{:,:,(\ell-1)n'<t\leq \ell n',:}. \tag{4.7}$$

Similarly to the proof of Proposition 32, we now choose S so that we can zero the negative entries of U and compute

$$\begin{aligned}
&((V \odot T) * \sigma((U \odot S) * X))_{r,s,\ell} \\
&\quad \text{by choosing } S \text{ so that } U \odot S = U^+ \odot S \\
&= \left((V \odot T) * \sigma\left(\left(U^+ \odot S\right) * X\right)\right)_{r,s,\ell} \\
&\quad \text{since } \left(U^+ \odot S\right) * X \text{ is positive} \\
&= \left((V \odot T) * \left(\left(U^+ \odot S\right) * X\right)\right)_{r,s,\ell} \\
&\quad \text{making convolutions explicit} \\
&= \sum_{(\ell-1)n'<t\leq \ell n'} V_{1,1,t,\ell} \sum_{i,j\in[d],k\in[c]} \left(U^+ \odot S\right)_{i,j,k,t} \cdot X_{r-i+1,s-j+1,k} \\
&\quad \text{substituting the definitions of Eq. 4.7} \\
&= \left(V^{(\ell)} * \left(\left(U^{(\ell)} \odot S^{(\ell)}\right) * X\right)\right)_{r,s}.
\end{aligned}$$

We now apply Proposition 32 with $\epsilon = \frac{\epsilon'}{c_1}$to each output channel of $\sigma((U \odot S) * X)$ by defining the events $\mathcal{E}_\ell^{(\text{layer})}$ as the fact that for any $K^{(\ell)} \in [-1,1]^{d\times d\times c_0\times c_1}$ and $X \in [0,1]^{D\times D\times c_0}$ it holds

$$\left\| K^{(\ell)} * X - V^{(\ell)} * \left(\left(U^{(\ell)} \odot S^{(\ell)}\right) * X\right)\right\|_{max} \leq \frac{\epsilon'}{c_1}.$$

We thus get that the intersection of all those events $\mathcal{E}^{(\text{layer})} = \bigcap_\ell \mathcal{E}_\ell^{(\text{layer})}$ (which corresponds to the thesis), holds with probability

$$\begin{aligned}
&\Pr\left(\mathcal{E}^{(\text{layer})}\right) \\
&= 1 - \Pr\left(\bar{\mathcal{E}}^{(\text{layer})}\right) \\
&= 1 - \Pr\left(\bigcup_\ell \bar{\mathcal{E}}_\ell^{(\text{layer})}\right)
\end{aligned}$$

$$
\begin{aligned}
&\text{by the union bound} \\
&\geq 1 - \sum_{\ell} \Pr\left(\bar{\mathcal{E}}_{\ell}^{(\text{layer})}\right) \\
&\text{by Proposition 32} \\
&\leq 1 - \sum_{\ell} \frac{\epsilon'}{c_1} \\
&= 1 - \epsilon'.
\end{aligned}
$$

□

4.3.3 Approximation of a CNN 31

We are now ready to prove Theorem 31, by closely following the same proof given in [**?**, Appendix B].

Proof of Theorem 31. To bound the error propagation across layers, let us define the layers' outputs[2]

$$
\begin{aligned}
&X^{(0)} = X, \\
&X^{(i)} = \sigma\left(K^{(i)} * X^{(i-1)}\right) \quad \text{for } 1 \leq i \leq \ell - 1, \\
&X^{(\ell)} = K^{(\ell)} * X^{(\ell-1)}.
\end{aligned} \tag{4.8}
$$

Notice that $X^{(\ell)}$ is the output of the target function, i.e.

$$
f(X) = X^{(\ell)} = K^{(\ell)} * X^{(\ell-1)}. \tag{4.9}
$$

For brevity's sake, given masks $S^{(1)}, ..., S^{(2\ell)}$, let us denote

$$
\tilde{L}^{(i)} = L^{(i)} \odot S^{(i)}. \tag{4.10}
$$

Since the ReLU function is 1-Lipschitz, for all $X^{(i-1)}$ it holds

$$
\begin{aligned}
&\left\| \sigma\left(K^{(i)} * X^{(i-1)}\right) - \sigma\left(\tilde{L}^{(2i)} * \sigma\left(\tilde{L}^{(2i-1)} * X^{(i-1)}\right)\right) \right\|_{\max} \\
&\qquad \leq \left\| K^{(i)} * X^{(i-1)} - \tilde{L}^{(2i)} * \sigma\left(\tilde{L}^{(2i-1)} * X^{(i-1)}\right) \right\|_{\max}.
\end{aligned} \tag{4.11}
$$

[2]We remark that some technicalities can be avoided by applying a ReLU to the last layer as well, so that it is analogous to other layers. However, we keep the last layer linear for consistency with the common practical use of analogous architectures.

Moreover, for each layer i, Lemma 34 implies that with probability at least $1-\frac{\epsilon}{2\ell}$ for all $X^{(i-1)} \in \mathbb{R}^{D\times D\times c_0}$ it holds

$$\begin{aligned}
&\left\|K^{(i)} * X^{(i-1)} - \tilde{L}^{(2i)} * \sigma\left(\tilde{L}^{(2i-1)} * X^{(i-1)}\right)\right\|_{\max} \\
&\quad \text{multiplying and dividing by } \left\|X^{(i-1)}\right\|_{max} \\
&= \left\|K^{(i)} * \frac{X^{(i-1)}}{\left\|X^{(i-1)}\right\|_{max}} - \tilde{L}^{(2i)} * \sigma\left(\tilde{L}^{(2i-1)} * \frac{X^{(i-1)}}{\left\|X^{(i-1)}\right\|_{max}}\right)\right\| \cdot \left\|X^{(i-1)}\right\|_{max} \\
&\quad \text{by Lemma 34} \\
&< \frac{\epsilon}{2\ell} \cdot \left\|X^{(i-1)}\right\|_{max}. \qquad (4.12)
\end{aligned}$$

Hence, combining Eq. 4.11 and Eq. 4.12 we get that with probability at least $1-\frac{\epsilon}{2\ell}$ for all $X^{(i-1)} \in \mathbb{R}^{D\times D\times c_{i-1}}$

$$\left\|\sigma\left(K^{(i)} * X^{(i-1)}\right) - \sigma\left(\tilde{L}^{(2i)} * \sigma\left(\tilde{L}^{(2i-1)} * X^{(i-1)}\right)\right)\right\|_{\max} < \frac{\epsilon}{2\ell} \cdot \left\|X^{(i-1)}\right\|_{max}. \qquad (4.13)$$

By a union bound, we get that Eq. 4.13 holds for all layer with probability at least $1-\epsilon$.

We now define the pruned layers' outputs

$$\begin{aligned}
&\tilde{X}^{(0)} = X, \\
&\tilde{X}^{(i)} = \sigma\left(\tilde{L}^{(2i)} * \sigma\left(\tilde{L}^{(2i-1)} * \tilde{X}^{(i-1)}\right)\right) \quad \text{for } 1 \le i \le \ell-1, \qquad (4.14) \\
&\tilde{X}^{(\ell)} = \tilde{L}^{(2\ell)} * \sigma\left(\tilde{L}^{(2\ell-1)} * \tilde{X}^{(\ell-1)}\right).
\end{aligned}$$

Notice that $\tilde{X}^{(\ell)}$ is the output of the pruned network, i.e.

$$N_0^{\left(S^{(1)},\ldots,S^{(2\ell)}\right)}(X) = \tilde{X}^{(\ell)} = \tilde{L}^{(2\ell)} * \sigma\left(\tilde{L}^{(2\ell-1)} * \tilde{X}^{(\ell-1)}\right). \qquad (4.15)$$

Observe that analogous equations to Eq. 4.12 and Eq. 4.13 hold for all pruned layers' output with probability $1-\epsilon$, namely

$$\left\|K^{(\ell)} * \tilde{X}^{(\ell-1)} - \tilde{L}^{(2\ell)} * \sigma\left(\tilde{L}^{(2\ell-1)} * \tilde{X}^{(\ell-1)}\right)\right\|_{max} \le \frac{\epsilon}{2\ell} \cdot \left\|\tilde{X}^{(\ell-1)}\right\|_{max} \qquad (4.16)$$

and

$$\left\|\sigma\left(K^{(i)} * \tilde{X}^{(i-1)}\right) - \sigma\left(\tilde{L}^{(2i)} * \sigma\left(\tilde{L}^{(2i-1)} * \tilde{X}^{(i-1)}\right)\right)\right\|_{\max} < \frac{\epsilon}{2\ell} \cdot \left\|\tilde{X}^{(i-1)}\right\|_{max}. \qquad (4.17)$$

Moreover, for each $1 \leq i \leq \ell - 1$

$$\begin{aligned}
&\left\|\tilde{X}^{(i)}\right\|_{max} \\
&= \left\|\tilde{X}^{(i)} - \sigma\left(K^{(i)} * \tilde{X}^{(i-1)}\right) + \sigma\left(K^{(i)} * \tilde{X}^{(i-1)}\right)\right\|_{max} \\
&\quad \text{by the triangle inequality} \\
&\leq \left\|\tilde{X}^{(i)} - \sigma\left(K^{(i)} * \tilde{X}^{(i-1)}\right)\right\|_{max} + \left\|\sigma\left(K^{(i)} * \tilde{X}^{(i-1)}\right)\right\|_{max} \\
&\quad \text{by Eq. 4.13} \\
&\leq \frac{\epsilon}{2\ell} \cdot \left\|\tilde{X}^{(i-1)}\right\|_{max} + \left\|\sigma\left(K^{(i)} * \tilde{X}^{(i-i)}\right)\right\|_{\max} \\
&\quad \text{by the Lipschitz property of } \sigma \\
&\leq \frac{\epsilon}{2\ell} \cdot \left\|\tilde{X}^{(i-1)}\right\|_{max} + \left\|K^{(i)} * \tilde{X}^{(i-i)}\right\|_{\max} \\
&\quad \text{by Proposition 33} \\
&\leq \frac{\epsilon}{2\ell} \cdot \left\|\tilde{X}^{(i-1)}\right\|_{max} + \left\|K^{(i)}\right\|_1 \left\|\tilde{X}^{(i-i)}\right\|_{\max} \\
&= \left\|\tilde{X}^{(i-1)}\right\|_{max} \left(1 + \frac{\epsilon}{2\ell}\right) \\
&\quad \text{unrolling the recurrence} \\
&\leq \left\|\tilde{X}^{(0)}\right\|_{max} \left(1 + \frac{\epsilon}{2\ell}\right)^i. \qquad (4.18)
\end{aligned}$$

Thus, combining Eq. 4.16 and Eq. 4.18 we get

$$\left\|K^{(\ell)} * \tilde{X}^{(\ell-1)} - \tilde{L}^{(2\ell)} * \sigma\left(\tilde{L}^{(2\ell-1)} * \tilde{X}^{(\ell-1)}\right)\right\|_{max} \leq \frac{\epsilon}{2\ell} \cdot \left(1 + \frac{\epsilon}{2\ell}\right)^{\ell-1}, \qquad (4.19)$$

and combining Eq. 4.17and Eq. 4.18 we get

$$\left\|\sigma\left(K^{(i)} * \tilde{X}^{(i-1)}\right) - \sigma\left(\tilde{L}^{(2i)} * \sigma\left(\tilde{L}^{(2i-1)} * \tilde{X}^{(i-1)}\right)\right)\right\|_{\max} < \frac{\epsilon}{2\ell} \cdot \left(1 + \frac{\epsilon}{2\ell}\right)^{i-1}. \qquad (4.20)$$

We then see that with probability $1-\epsilon$ for all $S^{(i)} \in \{0,1\}^{\text{size}\left(L^{(i)}\right)}$ with $i \in [2\ell]$ (remember that $S^{(i)}$ is implicit in Eq. 4.10) and all $X \in [-1,1]^{D \times D \times c_0}$

$$\begin{aligned}
&\left\|f(X) - N_0^{\left(S^{(1)},\ldots,S^{(2\ell)}\right)}(X)\right\|_{max} \\
&= \left\|X^{(\ell)} - \tilde{X}^{(\ell)}\right\|_{max} \\
&= \left\|K^{(\ell)} * \sigma\left(\cdots \sigma\left(K^{(1)} * X\right)\right) - \tilde{L}^{(2\ell)} * \sigma\left(\cdots \sigma\left(\tilde{L}^{(1)} * X\right)\right)\right\|_{max}
\end{aligned}$$

by Eq. 4.9 and Eq. 4.15

$$= \left\| K^{(\ell)} * X^{(\ell-1)} - \tilde{L}^{(2\ell)} * \sigma\left(\tilde{L}^{(2\ell-1)} * \tilde{X}^{(\ell-1)}\right)\right\|_{max} \tag{4.21}$$

by the triangle inequality

$$\leq \left\| K^{(\ell)} * X^{(\ell-1)} - K^{(\ell)} * \tilde{X}^{(\ell-1)}\right\|_{max} + \left\| K^{(\ell)} * \tilde{X}^{(\ell-1)} - \tilde{L}^{(2\ell)} * \sigma\left(\tilde{L}^{(2\ell-1)} * \tilde{X}^{(\ell-1)}\right)\right\|_{max}$$

by the distributive property of convolution

$$= \left\| K^{(\ell)} * \left(X^{(\ell-1)} - \tilde{X}^{(\ell-1)}\right)\right\|_{max} + \left\| K^{(\ell)} * \tilde{X}^{(\ell-1)} - \tilde{L}^{(2\ell)} * \sigma\left(\tilde{L}^{(2\ell-1)} * \tilde{X}^{(\ell-1)}\right)\right\|_{max}$$

by Proposition 33

$$\leq \left\| K^{(\ell)}\right\|_1 \cdot \left\|\left(X^{(\ell-1)} - \tilde{X}^{(\ell-1)}\right)\right\|_{max} + \left\| K^{(\ell)} * \tilde{X}^{(\ell-1)} - \tilde{L}^{(2\ell)} * \sigma\left(\tilde{L}^{(2\ell-1)} * \tilde{X}^{(\ell-1)}\right)\right\|_{max}$$

since $\left\| K^{(\ell)}\right\|_1 \leq 1$

$$\leq \left\| X^{(\ell-1)} - \tilde{X}^{(\ell-1)}\right\|_{max} + \left\| K^{(\ell)} * \tilde{X}^{(\ell-1)} - \tilde{L}^{(2\ell)} * \sigma\left(\tilde{L}^{(2\ell-1)} * \tilde{X}^{(\ell-1)}\right)\right\|_{max} \tag{4.22}$$

by Eq. 4.19 (which holds with prob. $1-\epsilon$ across all layers)

$$\leq \left\| X^{(\ell-1)} - \tilde{X}^{(\ell-1)}\right\|_{max} + \frac{\epsilon}{2\ell} \cdot \left(1 + \frac{\epsilon}{2\ell}\right)^{\ell-1}. \tag{4.23}$$

Similarly, for $1 \leq i \leq \ell - 1$ we have (again, with probability $1-\epsilon$)

$$\left\| X^{(i)} - \tilde{X}^{(i)}\right\|_{max}$$

by Eq. 4.8 and Eq. 4.14

$$= \left\|\sigma\left(K^{(i)} * X^{(i-1)}\right) - \sigma\left(\tilde{L}^{(2i)} * \sigma\left(\tilde{L}^{(2i-1)} * \tilde{X}^{(i-1)}\right)\right)\right\|_{max}$$

by the same calculations from Eq. 4.21 to Eq. 4.22

$$\leq \left\| X^{(i-1)} - \tilde{X}^{(i-1)}\right\|_{max} + \left\| K^{(i)} * \tilde{X}^{(i-1)} - \tilde{L}^{(2i)} * \sigma\left(\tilde{L}^{(2i-1)} * \tilde{X}^{(i-1)}\right)\right\|_{max}$$

by Eq. 4.20 (which holds with prob. $1-\epsilon$ across all layers)

$$\leq \left\| X^{(i-1)} - \tilde{X}^{(i-1)}\right\|_{max} + \frac{\epsilon}{2\ell} \cdot \left(1 + \frac{\epsilon}{2\ell}\right)^{i-1}$$

unrolling the recurrence for $\left\|X^{(i-1)} - \tilde{X}^{(i-1)}\right\|_{max}$

$$\leq \sum_{j=1}^{i} \frac{\epsilon}{2\ell} \cdot \left(1 + \frac{\epsilon}{2\ell}\right)^{j-1}$$

summing the geometric series

$$= \left(1 + \frac{\epsilon}{2\ell}\right)^{i} - 1. \tag{4.24}$$

Combining Eq. 4.23 and Eq. 4.24 we finally get that with probability $1 - \epsilon$

$$\begin{aligned}
&\left\| f\left(X\right) - N_0^{\left(S^{(1)},\ldots,S^{(2\ell)}\right)}\left(X\right)\right\|_{max} \\
&\leq \left(1 + \frac{\epsilon}{2\ell}\right)^{\ell-1} - 1 + \frac{\epsilon}{2\ell} \cdot \left(1 + \frac{\epsilon}{2\ell}\right)^{\ell-1} \\
&= \left(1 + \frac{\epsilon}{2\ell}\right)^{\ell} - 1 \\
&\leq e^{\frac{\epsilon}{2}} - 1 \\
&\qquad \text{since } \epsilon < 1 \\
&\leq \epsilon.
\end{aligned}$$

□

Chapter 5

10 Years of Research from Now

> Because the academic career puts a young person in a sort of compulsory situation to produce scientific papers in impressive quantity, a temptation to superficiality arises that only strong characters are able to resist. - Albert Einstein

In this work, after a brief overview of my first 10 years of research, I have presented some of my contributions to some theoretical aspects related to the role of sparsity in artificial neural networks. What is next? From reading Chapter 1, the reader has perhaps already formed a representation of my research path that is accurate enough to see how my work on sparsification in neural networks is just yet another attempt at pursuing the more ambitious goal of contributing sensibly to the understanding of the working of the human brain. In this respect, in this chapter, I sketch some research directions that I would like to pursue in the future. The theoretical results in Chapter 4 and some ongoing work I am pursuing, suggest that there are still interesting questions to be answered in the context of sparsity in artificial neural networks. In the following, I will first discuss the latter, and I will then elaborate more generally on future synergies that the field of neuroscience and theoretical computer science could benefit from.

In the present work, we have discussed a connection between the LTH and the Random Subset Sum Problem (RSSP). Rather surprisingly, the obtained theoretical results have also led me and my collaborators to insights into the area of neuromorphic computing which resulted in the deposit of a patent application [DCNV17]. We argue here that the connection between the RSSP and sparsity in neural networks is much deeper than its application to the SLTH. Generalizations of the RSSP could not only be leveraged to prove more general versions of the SLTH, but they also have important

implications for other training paradigms such as neural networks trained via genetic algorithms and spiking neural networks. Moreover, deep connections of an equivalent problem to the RSSP (the Random Number Partition Problem) have long been recognized in statistical physics. I am currently working on leveraging very recent advancements on the RSSP to obtain the first theoretical bounds on filter pruning for random CNNs (i.e. on ways to prune random CNNs that allow them to run more efficiently on the GPU). In parallel, I am also working on new results on the multidimensional variant of the RSSP which would improve recent general bounds on the integrality gap of random integer linear programs [BDHT]. Progress on the latter problem would also allow us to prove new results on some genetic algorithms for training neural networks. Moreover, the SLTH has practically been motivated by experimental works that train the network by only pruning it [RWK$^+$20]; the above generalization could motivate new versions of the previous paradigm. Not only the application of RSSP to the theory of neural networks appears a stimulating research direction, but it is also particularly original because of its connection to a classical problem in complexity theory, allowing the interaction of quite different areas of computer science. From a technical point of view, the above problems require improving probabilistic methods for the analysis of discrete random structures and processes (e.g. martingale techniques and related concentration inequalities and second-moment methods with dependent random variables) and developing new ones.

What insights could the above research provide for the understanding of sparsification in the brain? Some experimental works in the past have been at the origin of my interest in the phenomenon, such as [KWG$^+$01, WL03, TL12, TWK$^+$12] in which the authors show that the process of synapse elimination in the brain appears to be guided by a competitive process in which the less active synapses are eliminated. But more broadly, it might be inspiring to recall that the field of theoretical neuroscience has originally been in close contact with that of TCS, from McCulloch-Pitts neurons in 1943 to Von Neumann's influential book *The Computer and the Brain* in 1958. While the fields have drifted apart over the second half of the last century (with an approach predominantly guided by methodologies drawn from physics), we are nowadays witnessing a renewed synergy arising from an algorithmic lens on neuroscience [MPVL19]. In this respect, I have recently contributed to the Assembly Calculus (AC), an algorithmic framework that aims at bridging the gap between the behavior of individual neurons and the high-level cognitive functions of the brain by leveraging the Hebbian learning principle [dMC$^+$22]. The AC crucially leverages the concept of sparsity, with a group of neurons (assemblies) increasingly strengthening their interconnec-

tions while weakening those with the rest of the network. The analysis of such processes is at the heart of many applications in TCS, including many that I have investigated in the past. For example, in my recent joint results on the analysis of Levy flights in theoretical biology, I discuss the connections with the Small World phenomenon [CdGN21]. Small worldness is a concept that has been introduced in the context of complex networks, and it is believed to be a crucial feature of the brain [FZB16]. I am also investigating random graph models that could serve as *null models* to test hypotheses on network data extracted from neurological recordings (previous work in this respect is [FCC+21], which we mentioned in Section 1.3); the hyperbolic random graph model [KPK+10], for example, has attracted a lot of attention in the complex networks community because of its ability to reproduce salient properties observed in real networks, such as small worldness. I am also trying to provide a more solid theoretical explanation for the efficacy of the sparse hashing algorithm, called FlyHash, that has been observed in the olfactory system of drosophilas [DSN17], which should highlight the appealing algorithmic properties of FlyHash. Many other results as well found direct links between neuroscience and areas of TCS in which lies my expertise (e.g. distributed computing [AAB+11]), and provide a strong signal towards the profitability of further exploring the synergy between TCS and theoretical neuroscience.

Co-authored references

[BBN18] Luca Becchetti, Vincenzo Bonifaci, and Emanuele Natale. Pooling or Sampling: Collective Dynamics for Electrical Flow Estimation. In *Proceedings of the 17th Conference on Autonomous Agents and MultiAgent Systems*, Stockholm, April 2018. International Foundation for Autonomous Agents and Multiagent Systems.

[BCE+17] Petra Berenbrink, Andrea Clementi, Robert Elsässer, Peter Kling, Frederik Mallmann-Trenn, and Emanuele Natale. Ignore or Comply?: On Breaking Symmetry in Consensus. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, PODC '17, pages 335–344, New York, NY, USA, 2017. ACM.

[BCM+18] Luca Becchetti, Andrea Clementi, Pasin Manurangsi, Emanuele Natale, Francesco Pasquale, Prasad Raghavendra, and Luca Trevisan. Average Whenever You Meet: Opportunistic Protocols for Community Detection. In Yossi Azar, Hannah Bast, and Grzegorz Herman, editors, *26th Annual European Symposium on Algorithms (ESA 2018)*, volume 112 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 7:1–7:13, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[BCN+15] Luca Becchetti, Andrea Clementi, Emanuele Natale, Francesco Pasquale, and Riccardo Silvestri. Plurality Consensus in the Gossip Model. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, SODA '15, pages 371–390, San Diego, California, 2015. SIAM.

[BCN+16] L. Becchetti, A. Clementi, Emanuele Natale, F. Pasquale, and L. Trevisan. Stabilizing Consensus with Many Opinions. In

Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SODA '16, pages 620–635, Arlington, Virginia, 2016. SIAM.

[BCN+17a] L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and G. Posta. Self-stabilizing repeated balls-into-bins. *Distributed Computing*, pages 1–10, December 2017.

[BCN+17b] Luca Becchetti, Andrea Clementi, Emanuele Natale, Francesco Pasquale, Riccardo Silvestri, and Luca Trevisan. Simple dynamics for plurality consensus. *Distributed Computing*, 30(4):293–306, August 2017.

[BCN+19] Luca Becchetti, Andrea Clementi, Emanuele Natale, Francesco Pasquale, and Luca Trevisan. Finding a Bounded-Degree Expander Inside a Dense One. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms*, Proceedings, pages 1320–1336. Society for Industrial and Applied Mathematics, December 2019.

[BCN20a] Luca Becchetti, Andrea Clementi, and Emanuele Natale. Consensus Dynamics: An Overview. *ACM SIGACT News*, 51(1):58–104, March 2020.

[BCN+20b] Luca Becchetti, Andrea E. Clementi, Emanuele Natale, Francesco Pasquale, and Luca Trevisan. Find Your Place: Simple Distributed Algorithms for Community Detection. *SIAM Journal on Computing*, 49(4):821–864, January 2020.

[BKN18] Lucas Boczkowski, Amos Korman, and Emanuele Natale. Minimizing message size in stochastic communication patterns: Fast self-stabilizing protocols with 3 bits. *Distributed Computing*, pages 1–19, March 2018.

[BN16] Michele Borassi and Emanuele Natale. KADABRA is an ADaptive Algorithm for Betweenness via Random Approximation. In Piotr Sankowski and Christos Zaroliagis, editors, *Proceedings of the 24th Annual European Symposium on Algorithms (ESA'16)*, volume 57 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:18, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[BN19] Michele Borassi and Emanuele Natale. KADABRA is an ADaptive Algorithm for Betweenness via Random Approximation. *ACM Journal of Experimental Algorithmics*, 24(1), February 2019.

[BNFK18] Lucas Boczkowski, Emanuele Natale, Ofer Feinerman, and Amos Korman. Limits on reliable information flows through stochastic populations. *PLOS Computational Biology*, 14(6):e1006195, June 2018.

[BNV19] Hossein Baktash, Emanuele Natale, and Laurent Viennot. A Comparative Study of Neural Network Compression. Research Report, INRIA Sophia Antipolis - I3S, October 2019.

[CdGN21] Andrea Clementi, Francesco d'Amore, George Giakkoupis, and Emanuele Natale. Search via Parallel Lévy Walks on Z2. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, PODC'21, pages 81–91, New York, NY, USA, July 2021. Association for Computing Machinery.

[CDIG+15] Andrea Clementi, Miriam Di Ianni, Giorgio Gambosi, Emanuele Natale, and Riccardo Silvestri. Distributed community detection in dynamic graphs. *Theoretical Computer Science*, 2015.

[CGG+18] Andrea Clementi, Mohsen Ghaffari, Luciano Gualà, Emanuele Natale, Francesco Pasquale, and Giacomo Scornavacca. A Tight Analysis of the Parallel Undecided-State Dynamics with Two Colors. In Igor Potapov, Paul Spirakis, and James Worrell, editors, *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*, volume 117 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 28:1–28:15, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[CGN+20] Andrea Clementi, Luciano Gualà, Emanuele Natale, Francesco Pasquale, Giacomo Scornavacca, and Luca Trevisan. Consensus vs Broadcast, with and without Noise. In *ITCS 2020 - 11th Annual Innovations in Theoretical Computer Science*, 11th Innovations in Theoretical Computer Science Conference, pages 42–43, Seattle, United States, January 2020.

[CGPS17] Andrea E. F. Clementi, Luciano Gualà, Francesco Pasquale, and Giacomo Scornavacca. On the Parallel Undecided-State Dynamics with Two Colors. *arXiv:1707.05135 [cs]*, July 2017.

[CIG+13] Andrea Clementi, Miriam Di Ianni, Giorgio Gambosi, Emanuele Natale, and Riccardo Silvestri. Distributed Community Detection in Dynamic Graphs. In *Structural Information and Communication Complexity*, Lecture Notes in Computer Science, pages 1–12. Springer, Cham, July 2013.

[CLN+23] Pierluigi Crescenzi, Hicham Lesfari, Emanuele Natale, Aurora Rossi, and Paulo Serafim. Un framework open-source écrit en Julia pour la modélisation d'évaluation globale intégrée. In *ROADEF 2023 - 24ème édition du congrès annuel de la société française de recherche opérationnelle et d'aide à la décision*, Rennes, France, February 2023. Société Française de Recherche Opérationnelle et d'Aide à la Décision.

[CNNS18] Emilio Cruciani, Emanuele Natale, André Nusser, and Giacomo Scornavacca. Phase Transition of the 2-Choices Dynamics on Core-Periphery Networks. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pages 777–785, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

[CNRS23] Pierluigi Crescenzi, Emanuele Natale, Aurora Rossi, and Paulo Bruno Serafim. WorldDynamics.jl: A Julia Package for Developing and Simulating Integrated Assessment Models. working paper or preprint, June 2023.

[CNS19] Emilio Cruciani, Emanuele Natale, and Giacomo Scornavacca. Distributed Community Detection via Metastability of the 2-Choices Dynamics. In *AAAI 2019 - Thirty-Third AAAI Conference Association for the Advancement of Artificial Intelligence*, Honolulu, Hawaii, United States, January 2019.

[CNZ21] Andrea Clementi, Emanuele Natale, and Isabella Ziccardi. Parallel Load Balancing on constrained client-server topologies. *Theoretical Computer Science*, 895:16–33, December 2021.

[dCdG+22] Arthur da Cunha, Francesco d'Amore, Frédéric Giroire, Hicham Lesfari, Emanuele Natale, and Laurent Viennot. Revisiting the

Random Subset Sum problem. Research Report, Inria Sophia Antipolis - Méditerranée, Université Côte d'Azur ; Inria Paris, April 2022.

[dCN22] Francesco d'Amore, Andrea Clementi, and Emanuele Natale. Phase transition of a nonlinear opinion dynamics with noisy interactions. *Swarm Intelligence*, 16(4):261–304, December 2022.

[DCNV17] A.C.W. Da Cunha, E. Natale, and L. Viennot. Résistance équivalente modulable à partir de résistances imprécises (programmable equivalent resistances from imprecise resistors), Institut national de recherche en sciences et technologies du numérique, 2022. France Patent deposit n°FR2210217.

[dCNV22] Arthur da Cunha, Emanuele Natale, and Laurent Viennot. Proving the Strong Lottery Ticket Hypothesis for Convolutional Neural Networks. In *ICLR 2022 - 10th International Conference on Learning Representations*, Virtual, France, April 2022.

[dCNV23] Arthur da Cunha, Emanuele Natale, and Laurent Viennot. Neural Network Information Leakage through Hidden Learning. In *International Conference on Optimization and Learning (OLA2023)*, May 2023.

[dMC+22] Francesco d'Amore, Daniel Mitropolsky, Pierluigi Crescenzi, Emanuele Natale, and Christos Papadimitriou. Planning with Biological Neurons and Synapses. In *Proceedings of the AAAI Conference on Artificial Intelligence 2022*, volume Vol. 36 No. 1: AAAI-22 Technical Tracks 1, Vancouver, Canada, February 2022. Inria & Université Cote d'Azur, CNRS, I3S, Sophia Antipolis, France ; Gran Sasso Science Institute (L'Aquila, Italie) ; Department of Computer Science, Columbia University, New York. Type: Research Report.

[FCC+21] Matteo Frigo, Emilio Cruciani, David Coudert, Rachid Deriche, Emanuele Natale, and Samuel Deslauriers-Gauthier. Network alignment and similarity reveal atlas-based topological differences in structural connectomes. *Network Neuroscience*, 5(3):711–733, September 2021.

[FN19] Pierre Fraigniaud and Emanuele Natale. Noisy rumor spreading and plurality consensus. *Distributed Computing*, 32(4):257–276, August 2019.

[Nat17] Emanuele Natale. *On the Computational Power of Simple Dynamics*. Theses, Sapienza University of Rome, February 2017.

[RDN23] Aurora ROSSI, Samuel DESLAURIERS-GAUTHIER, and Emanuele NATALE. Temporal Brain Networks, March 2023.

[RN19] Iliad Ramezani and Emanuele Natale. On the Necessary Memory to Compute the Plurality in Multi-Agent Systems. In *CIAC'19 - 11th International Conference on Algorithms and Complexity*, Rome, Italy, May 2019.

Other references

[AAB+11] Yehuda Afek, Noga Alon, Omer Barad, Eran Hornstein, Naama Barkai, and Ziv Bar-Joseph. A biological solution to a fundamental distributed computing problem. *Science (New York, N.Y.)*, 331(6014):183–185, January 2011.

[AS16] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. Wiley, Hoboken, New Jersey, 4th edition edition, January 2016.

[BAB+19] Nathan Baker, Frank Alexander, Timo Bremer, Aric Hagberg, Yannis Kevrekidis, Habib Najm, Manish Parashar, Abani Patra, James Sethian, Stefan Wild, Karen Willcox, and Steven Lee. Workshop Report on Basic Research Needs for Scientific Machine Learning: Core Technologies for Artificial Intelligence. Technical report, USDOE Office of Science (SC), Washington, D.C. (United States), February 2019.

[BAN17] Bilash Kanti Bala, Fatimah Mohamed Arshad, and Kusairi Mohd Noh. *System Dynamics*. Springer Texts in Business and Economics. Springer Singapore, Singapore, 2017.

[BCMP04] C. Borgs, J. T. Chayes, S. Mertens, and B. Pittel. Phase diagram for the constrained integer partitioning problem. *Random Structures & Algorithms*, 24(3):315–380, 2004.

[BCP01] Christian Borgs, Jennifer T. Chayes, and Boris G. Pittel. Phase transition and finite-size scaling for the integer partitioning problem. *Random Struct. Algorithms*, 19(3-4):247–288, 2001.

[BDHT] Sander Borst, Daniel Dadush, Sophie Huiberts, and Samarth Tiwari. On the integrality gap of binary integer programs with Gaussian data.

[BGOFG20] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *Proceedings of machine learning and systems*, 2:129–146, 2020.

[BGPS06] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE transactions on information theory*, 52(6):2508–2530, 2006.

[BGS21] Paul Bastide, George Giakkoupis, and Hayk Saribekyan. Self-stabilizing clock synchronization with 1-bit messages. In *Proceedings of the Thirty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '21, pages 2154–2173, USA, March 2021. Society for Industrial and Applied Mathematics.

[BLMG22] Rebekka Burkholz, Nilanjana Laha, Rajarshi Mukherjee, and Alkis Gotovos. On the existence of universal lottery tickets. In *International Conference on Learning Representations*, 2022.

[Bon20] Vincenzo Bonifaci. On the Convergence Time of a Natural Dynamics for Linear Programming. *Algorithmica*, 82(2):300–315, February 2020.

[Bur22] Rebekka Burkholz. Most Activation Functions Can Win the Lottery Without Excessive Depth. In *Thirty-Sixth Conference on Neural Information Processing Systems*, December 2022.

[CFSV04] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(03):265–298, May 2004.

[CL06] Fan Chung and Linyuan Lu. Concentration inequalities and martingale inequalities: A survey. *Internet Mathematics*, 3(1):79–127, 2006.

[CNS19] Emilio Cruciani, Emanuele Natale, and Giacomo Scornavacca. Distributed Community Detection via Metastability of the 2-Choices Dynamics. In *AAAI 2019 - Thirty-Third AAAI Conference Association for the Advancement of Artificial Intelligence*, Honolulu, Hawaii, United States, January 2019.

[CRRS17] Colin Cooper, Tomasz Radzik, Nicolás Rivera, and Takeharu Shiraga. Fast Plurality Consensus in Regular Expanders. In

Andréa W. Richa, editor, *31st International Symposium on Distributed Computing (DISC 2017)*, volume 91 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 13:1–13:16, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Cru19] Emilio Cruciani. *Simple Randomized Distributed Algorithms for Graph Clustering*. PhD thesis, Gran Sasso Science Institute, 2019.

[Daw06] Richard Dawkins. *Unweaving the Rainbow: Science, Delusion and the Appetite for Wonder*. Penguin, London, 1er édition edition, 2006.

[DF89] Martin E. Dyer and Alan M. Frieze. Probabilistic analysis of the multidimensional knapsack problem. *Math. Oper. Res.*, 14(1):162–176, 1989.

[DGH+87] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. Epidemic algorithms for replicated database maintenance. In *Proceedings of the 6th ACM Symposium on Principles of Distributed Computing (PODC)*, 1987.

[DK] Benjamin Doerr and Anatolii Kostrygin. Randomized Rumor Spreading Revisited.

[DP09] Devdatt P. Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009.

[DSN17] Sanjoy Dasgupta, Charles F. Stevens, and Saket Navlakha. A neural algorithm for a fundamental computing problem. *Science*, 358(6364):793–796, November 2017.

[FB21] Jonas Fischer and Rebekka Burkholz. Towards strong pruning for lottery tickets with non-zero biases. *CoRR*, abs/2110.11150, 2021.

[FC19] Jonathan Frankle and Michael Carbin. 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, 2019.

[FGG96] Richard Phillips Feynman, David L Goodstein, and Judith R Goodstein. *Feynman's Lost Lecture: The Motion of Planets Around the Sun*. Norton, 1996.

[FK17] Ofer Feinerman and Amos Korman. The ANTS problem. *Distributed Computing*, 30(3):149–168, June 2017.

[For73] Jay Wright Forrester. *World Dynamics*. Cambridge, Mass. : Wright-Allen Press, 1973.

[FZB16] Alex Fornito, Andrew Zalesky, and Edward T. Bullmore. *Fundamentals of Brain Network Analysis*. Elsevier/Academic Press, Amsterdam ; Boston, 2016.

[GJ79] Michael R Garey and David S Johnson. *Computers and Intractability*, volume 174. freeman San Francisco, 1979.

[GK20] Brieuc Guinard and Amos Korman. The Search Efficiency of Intermittent L\'evy walks Optimally Scales with Target Size. *arXiv:2003.13041 [cs, q-bio]*, April 2020.

[JB03] E. T Jaynes and G. Larry Bretthorst. *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge, UK; New York, NY, 2003.

[Kar72] Richard M. Karp. Reducibility among Combinatorial Problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Springer US, 1972.

[KPK+10] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguñá. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, September 2010.

[KV22] Amos Korman and Robin Vacus. Early Adapting to Trends: Self-Stabilizing Information Spread using Passive Communication. In *Proceedings of the 2022 ACM Symposium on Principles of Distributed Computing*, PODC'22, pages 235–245, New York, NY, USA, July 2022. Association for Computing Machinery.

[KWG+01] Cynthia R. Keller-Peck, Mark K. Walsh, Wen-Biao Gan, Guoping Feng, Joshua R. Sanes, and Jeff W. Lichtman. Asynchronous Synapse Elimination in Neonatal Motor Units. *Neuron*, 31(3):381–394, August 2001.

[Lap18] Pierre Simon Laplace. *Théorie Analytique Des Probabilités*. Wentworth Press, 2018.

[Len17] Johannes Lengler. Drift Analysis. *arXiv:1712.00964 [cs, math]*, December 2017.

[Lev37] Paul Levy. *Théorie de l'addition Des Variables Aléatoires*. Gauthier-Villars, 1937.

[Lue82] George S. Lueker. On the Average Difference between the Solutions to Linear and Integer Knapsack Problems. In Ralph L. Disney and Teunis J. Ott, editors, *Applied Probability-Computer Science: The Interface Volume 1*, Progress in Computer Science, pages 489–504. Birkhäuser, Boston, MA, 1982.

[Lue98] George S. Lueker. Exponentially small bounds on the expected optimum of the partition and subset sum problems. *Random Structures and Algorithms*, 12:51–62, 1998.

[Lyn96] Nancy A Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.

[MCP21] Daniel Mitropolsky, Michael J. Collins, and Christos H. Papadimitriou. A Biologically Plausible Parser. In *Transactions of the Association for Computational Linguistics*, August 2021.

[Mer01] Stephan Mertens. A physicist's approach to number partitioning. *Theor. Comput. Sci.*, 265(1-2):79–108, 2001.

[MM09] Marc Mezard and Andrea Montanari. *Information, Physics, and Computation*. Oxford Graduate Texts. Oxford University Press, Oxford ; New York, 2009.

[MMR17] Pat Morin, Wolfgang Mulzer, and Tommy Reddad. Encoding Arguments. *ACM Computing Surveys*, 50(3):1–36, July 2017.

[MPVL19] Wolfgang Maass, Christos H. Papadimitriou, Santosh Vempala, and Robert Legenstein. Brain Computation: A Computer Science Perspective. In Bernhard Steffen and Gerhard Woeginger,

editors, *Computing and Software Science: State of the Art and Perspectives*, pages 184–199. Springer International Publishing, Cham, 2019.

[MRL+18] Frances C. Moore, James Rising, Niklas Lollo, Cecilia Springer, Valeri Vasquez, Alex Dolginow, Chris Hope, and David Anthoff. Mimi-PAGE, an open-source implementation of the PAGE09 integrated assessment model. *Scientific Data*, 5(1):180187, September 2018.

[MT17] Elchanan Mossel and Omer Tamuz. Opinion exchange dynamics. *Probability Surveys*, 14:155–204, 2017.

[MU17] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press, Cambridge, United Kingdom ; New York, NY, USA, 2 edition, July 2017. 10

[MYSSS] Eran Malach, Gilad Yehudai, Shai Shalev-Schwartz, and Ohad Shamir. Proving the lottery ticket hypothesis: Pruning is all you need. In *International Conference on Machine Learning*, pages 6682–6691. PMLR.

[Nor18] William Nordhaus. Evolution of modeling of the economics of global warming: Changes in the DICE model, 1992–2017. *Climatic Change*, 148(4):623–640, June 2018.

[OHR] Laurent Orseau, Marcus Hutter, and Omar Rivasplata. Logarithmic Pruning is All You Need. In *Advances in Neural Information Processing Systems*, volume 33, pages 2925–2934. Curran Associates, Inc.

[PRN+20] Ankit Pensia, Shashank Rajput, Alliot Nagle, Harit Vishwakarma, and Dimitris S. Papailiopoulos. Optimal lottery tickets via subset sum: Logarithmic over-parameterization is sufficient. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[PVM+20] Christos H. Papadimitriou, Santosh S. Vempala, Daniel Mitropolsky, Michael Collins, and Wolfgang Maass. Brain computation by assemblies of neurons. *Proceedings of the National Academy of Sciences*, June 2020.

[Rey18] Andy M. Reynolds. Current status and future directions of Lévy walk research. *Biology Open*, 7(1):bio030106, January 2018.

[RWK+20] Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. 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*, pages 11890–11899. Computer Vision Foundation / IEEE, 2020.

[Sha07] Devavrat Shah. Gossip Algorithms. *Foundations and Trends® in Networking*, 3(1):1–125, 2007.

[SK86] Michael F. Shlesinger and Joseph Klafter. Lévy Walks Versus Lévy Flights. In H. Eugene Stanley and Nicole Ostrowsky, editors, *On Growth and Form: Fractal and Non-Fractal Patterns in Physics*, NATO ASI Series, pages 279–283. Springer Netherlands, Dordrecht, 1986.

[ST01] John Slaney and Sylvie Thiébaux. Blocks World revisited. *Artificial Intelligence*, 125(1):119–153, January 2001.

[TL12] Stephen G. Turney and Jeff W. Lichtman. Reversing the Outcome of Synapse Elimination at Developing Neuromuscular Junctions In Vivo: Evidence for Synaptic Competition and Its Mechanism. *PLOS Biol*, 10(6):e1001352, June 2012.

[TWK+12] Juan C. Tapia, John D. Wylie, Narayanan Kasthuri, Kenneth J. Hayworth, Richard Schalek, Daniel R. Berger, Cristina Guatimosim, H. Sebastian Seung, and Jeff W. Lichtman. Pervasive Synaptic Branch Removal in the Mammalian Neuromuscular System at Birth. *Neuron*, 74(5):816–829, June 2012.

[VBH+99] G. M. Viswanathan, Sergey V. Buldyrev, Shlomo Havlin, M. G. E. da Luz, E. P. Raposo, and H. Eugene Stanley. Optimizing the success of random searches. *Nature*, 401(6756):911–914, October 1999.

[Win71] Terry Winograd. Procedures as a Representation for Data in a Computer Program for Understanding Natural Language. January 1971.

[WL03] Mark K. Walsh and Jeff W. Lichtman. In Vivo Time-Lapse Imaging of Synaptic Takeover Associated with Naturally Occurring Synapse Elimination. *Neuron*, 37(1):67–73, January 2003.

[ZLLY19] Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing Lottery Tickets: Zeros, Signs, and the Supermask. *NIPS*, page 11, 2019.