Is My Neural Net Driven by the MDL Principle?

Eduardo Brandao, Stefan Duffner, Rémi Emonet, Amaury Habrard, François Jacquenet, Marc Sebban

Is my Neural Net driven by the MDL Principle? *

Eduardo Brandao[1 [0000−0002−7146−8255]], Stefan Duffner[2 [0000−0003−0374−3814]], Rémi Emonet[1 [0000−0002−1870−1329]], Amaury Habrard[1,3 [0000−0003−3038−9347]], François Jacquenet[1 [0000−0002−0653−0710]], and Marc Sebban[1 [0000−0001−6851−169X]]

[1] Université Jean Monnet Saint-Etienne, CNRS, Institut d Optique Graduate School, Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France
Eduardo.Brandao@univ-st-etienne.fr
[2] CNRS, INSA-Lyon, LIRIS, UMR5205, Université de Lyon, F-69621 Villeurbanne, France
Stefan.Duffner@liris.cnrs.fr
[3] Institut Universitaire de France (IUF)

Abstract. The Minimum Description Length principle (MDL) is a formalization of Occam's razor for model selection, which states that a good model is one that can losslessly compress the data while including the cost of describing the model itself. While MDL can naturally express the behavior of certain models such as autoencoders (that inherently compress data) most representation learning techniques do not rely on such models. Instead, they learn representations by training on general or, for self-supervised learning, pretext tasks. In this paper, we propose a new formulation of the MDL principle that relies on the concept of signal and noise, which are implicitly defined by the learning task at hand. Additionally, we introduce ways to empirically measure the complexity of the learned representations by analyzing the spectra of the point Jacobians. Under certain assumptions, we show that the singular values of the point Jacobians of Neural Networks driven by the MDL principle should follow either a power law or a lognormal distribution. Finally, we conduct experiments to evaluate the behavior of the proposed measure applied to deep neural networks on different datasets, with respect to several types of noise. We observe that the experimental spectral distribution is in agreement with the spectral distribution predicted by our MDL principle, which suggests that neural networks trained with gradient descent on noisy data implicitly abide the MDL principle.

Keywords: Neural Networks · MDL · Signal-Noise · Point Jacobians.

* This work has been funded by a public grant from the French National Research Agency (ANR) under the "France 2030" investment plan, which has the reference EUR MANUTECH SLEIGHT - ANR-17-EURE-0026. This work has also been partly funded and by a PhD grant from the French Ministry of Higher Education and Research.

1 Introduction

New data often traces out regularities found in past observations, an idea known as generalization: finding regularities that are consistent with available data which also apply to data that we are yet to encounter. In the context of supervised machine learning we measure it by *learning* the rules on observations by minimizing some loss function, and evaluating it on observed and unobserved data. The difference between risk in the training data and new observations is known as the *generalization gap*. When it is small, the model generalizes well.

In the context of empirical risk minimization the generalization gap can be estimated in terms of model complexity, which increases with its number of parameters. We thus expect to reduce the generalization gap through a form of *regularization*, either by explicitely reducing the number of parameters, controlling a norm [27,51], or e.g. using dropout [43,21] or batch normalization [24,29,41].

Surprisingly, neural networks (NN) trained by stochastic gradient descent (SGD) generalize well despite possessing a higher number of parameters than training data, even without explicit regularization[14]. An elegant explanation for this phenomenon is that SGD implicitly controls model complexity during learning [35,19], resulting in networks that are significantly simpler than their number of parameters suggests, as shown by several metrics to assess effective capacity, e.g. the model's number of degrees of freedom[12], which is related to generalization gap, or its intrinsic dimension[28]. It is thus puzzling that, in spite of their implicit simplicity, NN classifiers trained by SGD are able to perfectly fit pure random noise [53], even while explicitly using regularization. In pure random noise, there is no signal to learn a rule from, and to reduce the generalization gap we must reduce the *training* performance. Since common regularization methods are unable to achieve this, using them to control model expressiveness does not address generalization: we need to "rethink generalization".

To do so we offer the following insight. To learn, from noisy observations, regularities that apply to data that we are yet to encounter, we must do so in a noise insensitive way: we must learn from signal rather than from noise. If we do so, there is no generalization gap when learning from pure noise: since there is no signal, the model would simply not learn at all!

In this paper, we shall give a formulation of this insight in terms of a minimum description length principle (MDL), [37,38] a principle of model selection which can be seen as a formalization of Occam's Razor. MDL states the problem of learning from data in terms of finding regularities that we can use to compress it: *choose the model that provides the shortest description of data, comprising the model itself*[4]. This idea was formulated in different ways since it was first advanced in [37], to respond to technical difficulties in application[15]. In the original, two-part form, restricting the model class to finite sets, application of this principle turns into Kolmogorov's minimal sufficient statistic [48].

[4] This formulation is known as two-part MDL, which depending on the author can be seen as "traditional" (in opposition to "modern" MDL which uses a one-step encoding using universal encodings [15]) or "pure [48]".

MDL expresses the ability to generalize in terms of compressibility, which can be motivated using three main facts: (i) regularities in a random variable X can be used to losslessly compress it (ii) the minimum achievable code length is the entropy (iii) it is very unlikely that data that has no regularities can be compressed. Taken together, these imply a model's ability to compress data is likely due to finding a regularity, which will likely be found in new data as well. It is this intuitive appeal that motivates the use of MDL in spite of some conceptual difficulties, namely in selecting the encoding used to measure the length of the description of the model, which depends on the choice of encoding.

To address this difficulty, we propose an approach that uses both the signal and the noise in the data to implicitly define model complexity unambiguously: *Choose the model whose representation of the data can be used to compress the signal, but not the noise.*

Formalizing this statement requires a perspective of signal and noise that is particularly adjusted to classification problems, where the signal is task-defined [15], and everything else can be considered as noise. As we shall see, our MDL statement has a significant impact on the distribution of the singular values of the point Jacobian matrices of a NN. Networks that learn from noise (where their output can be used to compress the noise) tend to maximize singular values in arbitrary directions to capture the fake "signal" in local directions. As a result the spectrum is uniformly distributed. On the other hand, NN that learn from signal but not from noise (where their output can be used to compress the signal but not the noise) tend to capture local regularities in the signal by maximizing singular values in directions aligned with the data. These directions are, by definition of signal, not arbitrary. Since the network also tends to ignore everything that is not signal, by minimizing singular values in arbitrary directions, in the limit of infinite epochs, this results in a spectrum distributed according to a power law, with a large proportion of small singular values and a fat tail.

Our contributions Our main contributions in this paper are 3-fold: (i) we provide a formulation of the MDL principle that is generally applicable to learned representations (ii) we provide a capacity measure based upon this principle (iii) we show experimentally that neural networks are driven by the MDL principle.

Paper organization This paper is organized as follows: Sec. 2 contextualizes of our work, focusing on the sensitivity measure provided in [2]. We then provide a few information theoretic results in Sec. 3.1 to contextualize the our definition of signal and noise in Sec. 3.2. Section 4 is the core of our contribution: we define our MDL objective in Sec. 4.1, and provide the local approximation in Sec. 4.2 that allows us to predict the spectral distribution in Sec. 4.2. In Sec. 5 we present experimental results[5] which allow us to conclude in Sec. 6 that neural networks are driven by the MDL principle, and discuss future work.

[5] Repository: `https://anonymous.4open.science/r/ismymodeldrivenbymdl-96BA/`.

2 Related work

MDL has traditionally been used for model selection [40,18,15,3,34], but its intuitive appeal has led to applications in other areas such as pattern mining [11,23]. In supervised learning, MDL was used in NN as early as [22], in which the authors added Gaussian noise to the weights of the network to control their description length, and thus the amount of information required to communicate the NN. In classification, existing approaches are inspired in MDL for density estimation[15], and most can be reduced to the same approach based on the $0/1$ loss, which, while not making probabilistic assumptions about noise, was shown to behave suboptimally [16]. Existing modifications to address this[4,50] do not have, unlike our approach, a natural coding interpretation. Finding a formulation of MDL for classification that can be applied in general and realistic settings is thus an open problem, and this paper aims to contribute in this direction.

The relationship between noise, compressibility and generalization has been explored in [6], for example, to derive PAC-Bayes generalization bounds, or in the information bottleneck framework[46]. Closer to our approach [33] studies the stability of the output of NN with respect to the injection of Gaussian noise at the nodes, experiments showing that networks trained on random labels are more sensitive to random noise. In [2], the notion of stability of outputs is extended to layer-wise stability, improving network compressibility and generalization. The authors define layer sensitivity with respect to noise (essentially the expected stable rank with respect to the distribution of the noise), and show that stable layers tend to attenuate *Gaussian* noise. A compression scheme is provided for the layer weights that acts on layer outputs as Gaussian noise, which subsequent stable layers will thus tend to attenuate. This, since the output of the network is unchanged, shows that a network composed of stable layers is losslessly compressible. A generalization bound for the compressed network is then derived in terms of the empirical loss of the original network and the complexity of the compressed network. This work shows a clear connection between compressibility of the model and generalization, but the connection to MDL is less evident. We will show that enforcing our MDL principle leads to a measure that can be seen as an average of local sensitivities, which are similar to those defined in [2], but with crucial differences. In our approach, sensitivity is logarithmic, direction-dependent, and importantly combines sensitivity to signal and to noise.

3 MDL principle, signal, and noise

We begin this section by recalling a few fundamental results in information theory, which will be used to define signal and noise as used in this paper.

3.1 Information theory primer

MDL rests on three fundamental results: (i) regularities in a random variable X can be used to losslessly compress it using a non-singular code for X; (ii) the

minimum achievable codelength is the entropy; and (iii) it is extremely unlikely that data that has no regularities can be compressed. In this section, we provide proof sketches for (ii) and (iii) in 8.6(see e.g. [8] or [30] for detailed proofs) and motivate (i) in 3.1 with a toy example. A similar argument can be used to prove a finite-precision version of the Theorem 1 in [52], which provides a necessary condition for a 2-Layer ReLU network to be able to perfectly fit the training data. A straightforward application of this original result allows us to show 8.1, for example, that a two-layer network that can be losslessly compressed to less than about 125 kB cannot perfectly overfit cifar-10[26].

Preliminaries and notation A source code $C(X)$ (C when there is no risk of ambiguity) for a random variable X is a function from $\mathcal{X}$ the range of X to $\mathcal{D}^*$ the set of finite strings of a d-ary alphabet $\mathcal{D}$, associating $x \in \mathcal{X}$ to a codeword $C(x)$. The *length* of the codeword $l(x)$ is the number of elements in $C(x)$, and the expected code length is $L(X) := \mathbb{E}_X[l(x)]$. A code is said to be *non-singular* if every $x \in \mathcal{X}$ maps to an unique element of $\mathcal{D}^*$. An *extension* C^* of code C codes sequences $x_1x_2\cdots x_n$of elements of $\mathcal{X}$ as the concatenation of $C(x_1)C(x_2)\cdots C(x_n)$. A code is said to be *uniquely encoded* if its extension is non-singular. Since every element in $\mathcal{X}$ is unambiguously encoded with a unique string, non-singular codes allow us to losslessly compress data.

Optimal codelength and incompressible data The Kraft-Macmillan inequality 2, which provides a condition for the existence of a uniquely decodable code with given word lengths proves (ii):

Theorem 1 (Optimal code length). *The expected length for any uniquely decodable code C of a random variable X over an alphabet of size D is greater than or equal to $H_D(X)$ the entropy calculated in base D, with equality holding iff $D^{-l_i} = p_i$*

An optimal prefix code always exists (e.g. Huffman code), but for our purposes, the Shannon-Fano code, which sets codeword lengths $l(x) = \lceil -\log p(x) \rceil$[6] suffices. To give an informal argument for (iii), consider data X with no regularities (maximal entropy). By Thm. 3, the expected codelength of any prefix code of a discrete random variable X over an alphabet of size D is at least $H_D(X)$, with equality iff the $l_i = -\log_D p_i$. Since all n events have probability $\frac{1}{n}$, the expected code length per symbol is $L \geq -\sum_{i=1}^{n} p_i \log_D p_i = \log_D n$. The lower bound can be achieved by assigning each codeword to the leaves of a D-nary tree: the best code and worst code coincide, and so data cannot be compressed.

Using regularities to compress To motivate (i), consider an object of mass m falling freely from a height h_0 on Earth (acceleration of gravity g), and a table recording heights $\{h_1, h_2, \ldots\}$ at times $\{t_1, t_2, \ldots\}$. which are known to obey

[6] The Shannon-Fano code is competitive, meaning that the probability that the expected length exceeds another code's by c bits does not exceed 2^{1-c} [8]

$h(t) = h_0 - \frac{1}{2}mgt^2$ since Galileo. This regularity can be used to losslessly compress the height-times table by replacing heights by $\Delta h_i = h_i - h(t_i)$, as we expect it to predict the first significant digits of the height with high confidence, and measurements are performed and stored with finite precision. We can thus store the *same* data (in expectation) using *less* digits, which amounts to lossless compression. The more regularities we are able to find in data, the more we can compress it. A better model, taking e.g. drag into account, increases confidence in the first significant digits of the predictions, reducing in expectation the number of significant digits of the deviations, allowing better compression.

Crucially, we did not take into account the size of the "data" that is the law itself. In the first case, we stored m and g, whereas in the second case we would need to store other quantities as well. There is a trade-off between the description lengths of the data and the model, as a better model takes longer to describe . In the limit, a very large model can decrease the description length of finite data simply by memorizing it. Notably, two-layer ReLU feed forward NN can do this with surprising ease[52] but, as predicted in the MDL framework, at the expense of an increase in complexity[5].

3.2 Signal and noise

This paper introduces an MDL principle that specifies the encoding scheme in which to measure the description length implicitly in terms of the signal and the noise in noisy data. To define signal and noise, we rely on [39] which defines noise as the part of the data that cannot be compressed with the models considered, the rest defining the information bearing signal. This idea is used in the paper in the context of Gaussian models arising in linear-quadratic regression problems to derive a decomposition of data that is similar to Kolmogorov's sufficient statistics [8]. In our case, we shall assume that the signal is implicitly provided by a given classification task, and define noise to be everything else.

Definition 1. *We define noise as "noise relative to a signal": given random variables X (signal) and Δ (noise) such that $X + \Delta$ is well-defined, we say that Δ is noise relative to X if for every $C_i \in \mathcal{C}$ non-singular code of X, we have $L(C_i(\Delta)) \geq H(\Delta) + \alpha$, with $\alpha > 0$.*

Note that if $C_j \in \mathcal{C}$ were optimal for Δ, then $L(C_i(\Delta)) = H(\Delta) \geq H(\Delta) + \alpha$, which with $\alpha > 0$ is a contradiction. The definition is thus equivalent to stating that there is no code of X in $\mathcal{C}$ (which may include the optimal code for X) that is optimal for Δ. Also note that the noise Δ is not particularly "disordered". Going back to 3.1, the physical laws that compress height vs. time data are unable to compress the effect of hitting the object with a baseball bat. Even if a model provides a simple description of some data, adding noise as defined in 1 destroys its ability to compress it. It is implicit in the MDL principle that not only do we learn the regularities in data, but also the "irregularities"!

4 Learning with the MDL principle

We now provide an MDL principle that eliminates the need for defining the model encoding, as in two-step MDL or a universal coding such as one-step MDL[17]. Instead, we utilize the signal and the noise in the training data to implicitly define the encoding. We then establish a lower bound of this maximization objective in terms of the minimal description lengths of signal and noise(cf. 3). We further simplify the problem by expressing it locally, which enables us to provide an interpretation in terms of sensitivities to the signal and noise. Finally, we combine these local problems to express a global MDL objective in terms of the spectra of the local Jacobians, and that the spectral distribution of models that maximize MDL is either power law or lognormal.

4.1 MDL objective

The MDL paradigm quantifies learning based on the ability to compress: if $f(X+\Delta)$ contains information about X it can compress it and conversely, if it does *not* contain information about the Δ, it cannot be used to compress it. This formulation implicitly defines the complexity of the model f in terms of unknown X and Δ present in training data. It is therefore applicable in a classification context, where these are defined with respect to a *task*. Formally:

Definition 2 (MDL principle). *Let $\tilde{X} = X + \Delta$ be noisy data, comprised of unknown signal X and a noise Δ parts in the sense of 1, and a model f_θ trained on $\tilde{X}$ according to some (e.g. classification) objective. Let $\mathcal{L}(X|f(\tilde{X}) = y)$ and $\mathcal{L}(\Delta|f(\tilde{X}) = y)$ be, respectively, the expected description length of X and Δ given knowledge $f_\theta(\tilde{X}) = y$. Then with $\gamma > 0$ a hyperparameter, f_θ follows the MDL principle if it maximizes*

$$\max_\theta \left\{ \int p_{f_\theta(\tilde{X})}(y)\mathcal{L}(\Delta|f_\theta(\tilde{X}) = y)dy - \gamma \int p_{f_\theta(\tilde{X})}(y)\mathcal{L}(X|f_\theta(\tilde{X}) = y)dy \right\} \quad (1)$$

The idea is to minimize the mean $\mathcal{L}(X|f(\tilde{X}) = y)$ and maximize $\mathcal{L}(\Delta|f(\tilde{X}) = y)$ seen as functions of y[7], with γ controlling the relative strength of these objectives.

A lower bound in terms of minimal description length Using Theorem 3 we can express the length of the description of noise knowing $f_\theta(\tilde{X}) = y$ as a multiple $\alpha(y) \geq 1$ of the length of the minimum length description for each y:

$$\begin{aligned} \int p_{f_\theta(\tilde{X})}(y)\mathcal{L}(\Delta|f_\theta(\tilde{X}) = y)dy &= \int p_{f_\theta(\tilde{X})}(y)\alpha(y)H(\Delta|f_\theta(\tilde{X}) = y)dy \\ &\geq \left(\inf_y \alpha(y)\right) \int p_{f_\theta(\tilde{X})}(y)H(\Delta|f_\theta(\tilde{X}) = y)dy \\ &= \left(\inf_y \alpha(y)\right) H(\Delta|f_\theta(\tilde{X})) \end{aligned}$$

[7] For classification, we work on an intermediate representation, which explains the use of integrals in calculating the expectation.

Proceeding similarly for the signal term we obtain

$$\int p_{f_\theta(\tilde{X})}(y)\mathcal{L}(X|f_\theta(\tilde{X})=y)dy \leq \left(\sup_y \beta(y)\right) H(X|f_\theta(\tilde{X}))$$

Denoting $\inf_y \alpha(y) := \alpha$ and $\sup_y \beta(y) := \beta$ the minimum and maximum expected description lengths of codes of noise and of signal, respectively, knowing $f_\theta(\tilde{X}) = y$, we combine the two desiderata and maximize a lower bound of 1:

$$\max_\theta \left\{ \alpha H(\Delta|f_\theta(\tilde{X})) - \gamma\beta H(X|f_\theta(\tilde{X})) \right\}$$

Since $H(\Delta|f_\theta(\tilde{X})) = H(\Delta, f_\theta(\tilde{X})) - H(f_\theta(\tilde{X}))$ and similarly for the second term,

$$\begin{aligned} H(\Delta|f_\theta(\tilde{X})) - \gamma\beta H(X|f_\theta(\tilde{X})) = \alpha H(f_\theta(\tilde{X})|\Delta) - \gamma\beta H(f_\theta(\tilde{X})|X) \\ + \alpha H(\Delta) - \gamma\beta H(X) + (\beta\gamma - \alpha)H(f_\theta(\tilde{(X)})) \end{aligned}$$

Ignoring terms independent of θ, since $\alpha > 0$, we obtain a lower bound of 1:

Proposition 1 (MDL objective lower bound). *Given noisy data $\tilde{X} = X + \Delta$ comprised of a signal X and a noise Δ parts, a model f_θ trained on $\tilde{X}$ according to MDL, $\lambda := \gamma\frac{\beta}{\alpha}$, the following is a lower bound of the the MDL objective:*

$$\max_\theta \left\{ H(f_\theta(\tilde{X})|\Delta) - \lambda H(f_\theta(\tilde{X})|X) + (\lambda - 1)H(f_\theta(\tilde{X})) \right\} \tag{2}$$

In this lower bound, λ has the role of γ modulated by the ratio between the worst case expected signal description length knowing the model output and the best case description length of the noise knowing the model output in units of entropy. EB : Note that to minimize the description length of the noisy data $H(f_\theta(\tilde{X}))$ we must have $\lambda - 1 < 0$ and hence objective 2 is MDL with a constraint on the conditional entropies. Since $\lambda < 1 \Rightarrow \alpha > \gamma\beta$ the implications depend on the model class $\{f_\theta\}$: if for the given model class Δ is more difficult to compress than X, then $\alpha > \beta$ and so $\gamma < 1$. This corresponds to, in 2, focusing relatively more on ignoring the noise. Conversely, if $\{f_\theta\}$ is such that X is mode difficult to compress, then $\gamma > 1$ and we focus relatively more on learning the signal.

4.2 Local formulation

We now simplify the problem in 2 by expressing it *locally* and then ultimately in terms of the spectrum of the point Jacobian matrix $\nabla f_\theta|_{x_k}$.

Local objective Let $f : A \subseteq \mathbb{R}^n \to B \subseteq \mathbb{R}^m$ be analytical, A compact and $x_1, \ldots, x_N \subseteq A$ and $\{V_k\}_{k=1\cdots N}$ a set of balls centered at x_k and with radius r_k such that $A \subseteq V_1 \cup \cdots \cup V_N$, chosen such that the Jacobian matrix of f is constant in each V_k in the sense of Prop. 5. Then to first order in $\delta x_k, \delta$:

$$\begin{aligned} f(\tilde{x}) &= f(\tilde{x}_k + \delta x_k + \delta_k) \\ &\approx f(\tilde{x}_k) + \nabla f|_{\tilde{x}_k}\delta x_k + \nabla f|_{\tilde{x}_k}\delta_k \\ &:= f(\tilde{x}_k) + J_k\delta x_k + J_k\delta_k \end{aligned}$$

with the approximation error controlled by the principal singular value of the Hessian (cf. app. 8.3 for a proof). Since the choice of V_k determines $f(\tilde{x}_k)$, assuming local independence of signal and noise EB : implies locally $H(f(X)|X) = 0, H(X|\Delta) = H(X)$; we can thus apply this approximation to 2 to obtain a local MDL objective:

Proposition 2 (Local MDL objective). *In the conditions and notation above, locally in V_k the MDL objective 2 can be expressed approximately as*

$$\max_{J_k} \lambda H(J_k \delta X_k) - H(J_k \Delta_k) \tag{3}$$

where $\delta X_k, \Delta_k$ denote the signal and the noise in V_k with respect to its center, and the approximation error is controlled by Prop. 5.

Interpretation in terms of sensitivity measure in [2] In [2] the authors define sensitivity of a mapping f with respect to noise Δ at x as $\mathbb{E}_{\delta \sim \Delta}\left[\frac{\|f(x+\delta)-f(x)\|^2}{\|f(x)\|^2}\right]$, which becomes $\frac{\|J_k(\delta)|^2}{\|f(x)\|^2}$ to first order in δ, in a region of constant Jacobian J_k, using the arguments in 4.2. In expectation, up to a scale, this is the variance of $J_k\Delta_k$ which is a measure of its complexity like the entropy above, (for a Gaussian distribution, up to a logarithm and a constant, the two coincide). $H(J_k\Delta_k)$ in prop. 2 thus corresponds to sensitivity with respect to noise and, by a similar argument, $H(J_k\delta X_k)$ to sensitivity with respect to signal. Our MDL objective thus selects the model that locally maximizes sensitivity with respect to signal and minimizes sensitivity with respect to noise. Although similar to [2], in our formulation sensitivity is logarithmic, direction-dependent (cf.4.2), and crucially combines sensitivity to signal and sensitivity to noise.

Finally, since $\lambda < 1$, if $H(J_k\delta X_k) > H(J_k\Delta_k)$ then 3 is upper bounded by zero, where $\lambda = \frac{H(J_k\Delta_k)}{H(J_k\delta X_k)}$. Maximizing 3 thus corresponds to getting closer to a model that *locally* produces the same balance between sensitivity to signal and to noise, determined by the *global* parameter λ. This problem cannot always be solved. Consider f a one layer ReLU network of width N; the *local* $\{J_k\}$ are given by deleting a certain number of rows in the pre-ReLU Jacobian, which is the weight matrix of f. Since f can have at most 2^N different $\{J_k\}$, the conjunction of local problems can only be solved if the number of V_k where the balance between sensitivities needs to be adjusted *differently* is smaller that 2^N. The case of deeper networks is similar, each new ReLU layer of width M_i multiplying the number of possible Jacobians by 2^{M_i}.

Local objective: spectral formulation To provide a spectral version of 2, we express J_k in terms of its singular value decomposition (SVD), and the signal and noise in terms of local PCA representations. We work in V_k but omit the label k for simplicity. Jacobian, signal, and noise refer to the *local* versions.

Proposition 3. *(Local objective spectral formulation) In the conditions of prop. 2, the following is its lower bound:*

$$\max_{\sigma} \left\{ \lambda \left(\max_{i} \left\{ \log \sigma_i + H(\delta X^i_{pca}) \right\} \right) - \sum_j \left(H(\Delta^j_{pca}) + \log \sigma_j \right) \right\} \tag{4}$$

Proof. Let $J = U\Sigma V^\top$ be the singular value decomposition of $J \in \mathbb{R}^{n\times m}$. The signal δX can be expressed as the transform to local coordinates of δX_{pca}, the signal in local PCA coordinates $\delta X = W^\top_{signal}\delta X_{pca}$, and similarly for noise: $\Delta = W^\top_{noise}\Delta_{pca}$, where W_{signal}, W_{noise} are, respectively, the PCA coordinate transformation for signal and for noise. EB : Noting that U has determinant one everywhere we thus have

$$\lambda H(J\delta X) - H(J\Delta) = \lambda H(\Sigma V W^\top_{signal}\delta X_{pca}) - H(\Sigma V W^\top_{noise}\Delta_{pca})$$

The $VW^\top$ are contractions measuring the alignment between the singular vectors of the Jacobian and the principal components of the signal (for W_{signal}) and noise (for W_{noise}). We thus maximize the RHS of this expression by:

- aligning J with δX and then maximizing the logarithm of the singular values in the non-zero dimensions: if δX is locally low-dimensional, the singular values that get maximized are few.
- aligning J with Δ and then minimizing the logarithm of the singular values in the non-zero dimensions: since Δ tends to be relatively high-dimensional, all singular values of J tend to be minimized.[8]

The overall effect is to maximize a few neighborhood-dependent singular values of J, and minimize all the rest – consistently with the experimental observations 1. Since δX and J are unknown, so are the "selected" directions. The full entropy of the local signal is at least as great as that of its components. Replacing it with the entropy of the singular direction i for which the entropy of the transformed signal is maximal, we obtain a lower bound of the local objective.

4.3 Combining local objectives to obtain a spectral distribution

We combine local objectives by maximizing their sum over all local patches V_k. This is essentially assuming cross-patch independence. For it to hold, (i) the network should be able to produce sufficiently many local Jacobians as explained in 4.2 and (ii) $V_i \cap V_j$ should be small for all i, j. Assumption (i) holds in practice since we work in the overparameterized regime and (ii) holds for ReLU networks. Both assumptions are thus expected to hold as a first approximation, although [20] suggests more complex behavior and will be considered in future work.

Recalling that we do not know which singular value gets "selected" and assuming that the signal is locally low-dimensional (which is known as "the manifold hypothesis"[7,10]), which we take for simplicity to mean that $\max_{i_k} H(\delta X^{i_k}_k) \approx$

[8] A similar argument can be found in [2] in the discussion of noise sensitivity.

$H(\delta X_k)$ we obtain, summing over the M patches of rank-N_k Jacobian

$$\sum_{k=1}^{M} \left\{ \lambda \left(\max_{i_k} \left\{ \log \sigma_{i_k} + H(\delta X_k^{i_k}) \right\} \right) - \sum_{j=1}^{N_k} \left(H(\Delta_k^j) + \log \sigma_j \right) \right\}$$

Simplifying and maximizing over the singular values of all the J_k leads to

$$\max_{\sigma} \left\{ \lambda M \mathbb{E}\left[\log \sigma\right] + \lambda H(X) - H(\Delta) - \bar{N} M \mathbb{E}\left[\log \sigma\right] \right\}$$

where expectations of both log singular values and Jacobian rank are over the patches, the latter denoted $\bar{N}$ for readability. As the sum of lower bounds of non-positive quantities is non-positive, its maximum value is zero, where

$$\mathbb{E}\left[\log \sigma\right] = \frac{H(\Delta) - \lambda H(X)}{M(\lambda - \bar{N})} \tag{5}$$

Expectation as a model-dataset measure of complexity For this expectation to be positive the entropy of the noise must be sufficiently smaller than the entropy of the signal, since $\lambda - \bar{N} < 0$ because $0 < \lambda < 1$. If 1 holds, $\mathbb{E}\left[\log \sigma\right]$ thus decreases with the number of patches of constant Jacobian and the mean Jacobian rank. It is thus a measure of model complexity which increases with the weighted difference between the entropy of noise and the entropy of signal, *it depends on the signal and noise*. All things being equal, for the same $\mathbb{E}\left[\log \sigma\right]$ models trained with more noise will have smaller M and $\tilde{N}$. Adding noise is a form of regularization. If on the other hand, entropy of noise is greater than the entropy of signal, the reverse effect is produced. On very noisy data (relative to signal!), models trained with more noise need to become more complex.

4.4 The MDL spectral distributions

We now show that the predicted distribution that is compatible with 6 is a power law or, for NN trained with SGD, a lognormal distribution. The true spectral distribution contains information on, e.g. architecture and training process whereas in the maxent formalism [25] we use, the prediction is maximally non-committal: it contains no information on the MDL-trained network beyond its adherence to the MDL principle and the signal-to-noise entropies of the training data.

Incorporating knowledge of the expectation of the log spectrum and SGD The distribution that incorporates knowledge of the expectation of the spectrum6 *and nothing else* is the maximum entropy distribution for which the constraint on the spectrum 6 holds[25]. Specifically, the power law distribution $p(\sigma) = \frac{\alpha-1}{\alpha} \left(\frac{\sigma}{b}\right)^{-\alpha}$, where $\alpha = 1 + \frac{1}{\mathbb{E}[\log \sigma] - \log b}$ and b is a cutoff parameter. Power laws model scale-free phenomena[9], but can emerge when aggregating data over many

[9] Since $p(k\sigma) = a\,(k\sigma)^{\alpha} = ak^{\alpha}\sigma^{\alpha}$. Since the constant is a normalization factor, we must have $p(k\sigma) = p(\sigma)$.

scales [54,13], as we did in 4.3 to obtain eq. 6. For a ReLU NN trained by SGD, there is also a constraint on the *variance* of $\log \sigma$: the spectrum depends continuously on the network weights (cf. sec. 4.2), which are SGD-updated using a *finite* number of steps. The corresponding maxent distribution is the lognormal, which is the Gaussian distribution with given mean and variance in log-scale.

5 Experimental results

Our experiments show that spectral distribution matches theoretical predictions in 4.4, suggesting that NN are driven by the MDL principle. We study the effect of noise in the point Jacobian spectral distribution of three groups of models of increasing complexity, ReLU MLPs, Alexnet, and Inception trained on MNIST [9] and cifar-10 [26], using the experimental setup in [52]. See app. 8.5 for details. The section is organized as follows: (i) we present two different types of noise and discuss expected consequences with respect to spectral distribution, and (ii) we present and discuss the experimental results.

5.1 Experimental Noise

We study two forms of "natural" noise: *label noise*, used in [52] and *dataset noise*, which consists in adding a lossy compressed version of a similar dataset.

Label noise We focus on instance-independent symmetric label noise[42], which randomly assigns labels to training and test examples unconditionally on example and training label with probability p. Label noise can be modelled realistically using human annotators[49], but the former choice is closer to the MDL sense 1. In this setting, the entropy of the introduced noise can be estimated as $p \cdot H(X_0)$, since incorrectly labelled examples become noise with respect to the classification task. This allows us to express the numerator of 6 for the noised dataset in terms of the entropy and noise of the original dataset as $H(\Delta_p) - \lambda H(X_p) = H(X_0) - \lambda H(X_0) + p(1+\lambda)H(X) > H(X_0) - \lambda H(X_0)$. All things being equal, for NN following MDL, the log Jacobian point spectrum increases with the probability of label noise p.

Dataset noise We add to the original dataset D_0 a *similar* dataset D_{sim} lossy compressed at rate r. Symbolically $D_r = D_0 + rD_{sim}$. We choose D_{sim} commonly used in place of D_0 in ML practice: cifar-100 for cifar-10, and Fashion-MNIST for MNIST. To compress $\tilde{D}$ we reconstruct it using only a certain number of PCA components. This causes less bias in setting r, compared to compressing with e.g. jpeg [36] or an autoencoder, in which the architecture introduces an element of arbitrarily, but we lose the ability to set the compression rate at will. Since for the noised dataset $X_r + \Delta_r$ the numerator in 6 can be written as $H(\Delta_r) - \lambda H(X_r) = H(\Delta_0) - \lambda H(X_0) + r(H(X_{sim}) + H(\Delta_{sim})$. All things being equal, for NN that follow MDL, the average log Jacobian point spectrum decreases with r. Interestingly, assuming the entropies of the similar dataset are approximately

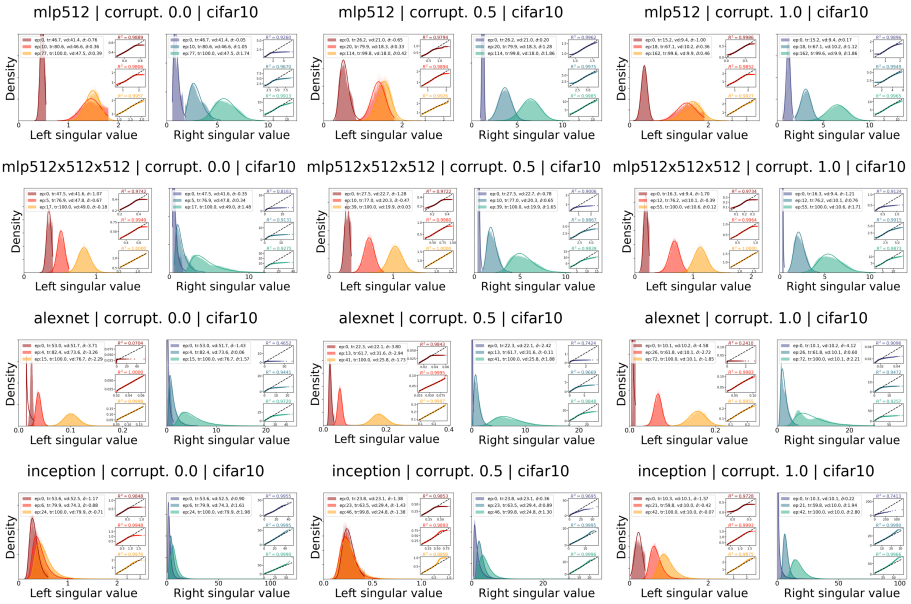

Fig. 1: Point Jacobian spectral distribution for *model* | ***label noise*** | *cifar-10*, from first epoch to overfit. "Left" and "right" distributions (cf. 8.7) are represented separately for each triplet for clarity. The best fit lognormal plot is superimposed on each histogram, with the corresponding probability plot on the right, with the line of best fit (R^2 displayed on top). Legend elements, in order: epoch, training and validation accuracy, and the mean log spectrum.

the same as that of the original dataset, we obtain $H(\Delta_r) - \lambda H(X_r) = (1 + r)H(\Delta_0) - (\lambda - r)H(X_0)$, which corresponds to the same maximization objective with a rescaled $\lambda_r = \frac{\lambda - r}{1+r} < \lambda_0$ corresponding to less sensitivity to signal.

5.2 Discussion

As Figs. 1 and 2 show, NN trained using SGD are driven by the MDL principle: (i) their spectra is remarkably well-fit by a lognormal distribution, as predicted in 4.4, and experimental spectra become globally more lognormal with training epoch (cf. fit overlay on the histograms, and inset probability plots); also, as predicted in the discussion following 6 (ii) for each model $\mathbb{E}[\log \sigma]$ tends to increase with noise (iii) and with model complexity, which also influences the quality of lognormal fit[10], Inception being the overall best and MLP the overall worst. Remarkably, these observations hold for both label noise and dataset noise. In the early stages of the training process, though, representation-building takes

[10] The number of training epochs being relatively small, we did not find a power-law behavior.

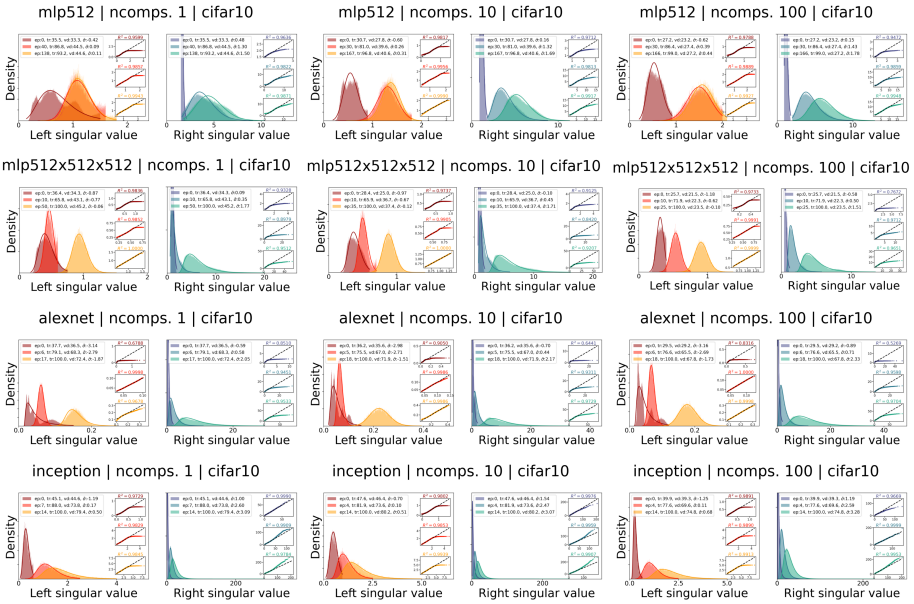

Fig. 2: Point Jacobian spectral distribution for *model* | ***nbr. pca comp.*** | *cifar-10*, from first epoch to overfit where possible. "Left" and "right" distributions (cf. 8.7) are represented separately for each triplet for clarity. The best fit lognormal plot is superimposed on each histogram, with the corresponding probability plot on the right, with the line of best fit (R^2 displayed on top). Legend elements, in order: *epoch*, *training and validation accuracy*, and the *mean log spectrum*.

precedence. This can be inferred by observing that experimental distributions are typically bimodal (see Sec. 8.7, for figures and discussion), and noting that at the last linear layer of a classification-induced representation, one of the directions should leave the output relatively more unchanged than the others: the direction assigned to the class of the training point (see 14 for a visual explanation). Representation building occurs early, as can be seen in Figs. 8.7 or in Figs. 1 and 2, dominating MDL in early epochs. To handle this asymmetry, we divide the spectrum in each of its two modalities (cf. 8.5). The statements above apply to each of the two parts of the spectrum, corresponding to the two representations. The observations above hold for MNIST as well, exception being where the initial spectrum is multi-modal (suggesting a great degeneracy of the directions in which the classification prediction does not change — i.e. MNIST is very simple). In this case our splitting method is ineffective, as we would need to split the spectral distribution into each of the several modalities.

6 Conclusion and future work

In this work, we propose an MDL principle that implicitly defines model complexity in terms of signal and noise: *choose the model whose representation of the data can be used to compress the signal, but not the noise.* We show that models driven by this principle locally maximize sensitivity to the signal and minimize the sensitivity to noise, and predict that the point Jacobian spectrum of NN trained by gradient descent follow either a power law or a lognormal distribution. We provide experimental evidence supporting this prediction, hinting that neural networks trained by gradient descent are driven by the MDL principle.

As for future work we plan, aiming at a generalization bound, to extend the connection established in 4.2, by making the MDL objective layer wise as in [2]. Another possible extension is to use our findings to explain the power law behavior of the spectra of the layer weight matrices and connection to generalization gap found in [31,32], by noting that each point Jacobian of ReLU networks is a sub-matrix of the product of the network weight matrices, which can beexpressed in terms of the singular values of the point Jacobian submatrix via an interlacing inequality[45].

7 Ethical statement

This paper presents a contribution that is essentially fundamental, theoretical and methodological. We do not see any immediate ethical or societal issues. Our experimental evaluation considers classic benchmarks of the literature and our analysis focuses on particular mathematical properties of point Jacobians spectra of trained neural networks. Our work follows ethical guidelines in modern machine learning research in general and in representation learning in particular. The application of the methodology presented in this paper should consider ethical implications that can arise from the datasets used of the applications targeted.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), `https://www.tensorflow.org/`, software available from tensorflow.org
2. Arora, S., Ge, R., Neyshabur, B., Zhang, Y.: Stronger generalization bounds for deep nets via a compression approach. CoRR (2018), `http://arxiv.org/abs/1802.05296v4`
3. Barron, A., Rissanen, J., Yu, B.: The minimum description length principle in coding and modeling. IEEE transactions on information theory **44**(6), 2743–2760 (1998)

4. Barron, A.R.: Complexity regularization with application to artificial neural networks. Nonparametric functional estimation and related topics pp. 561–576 (1991)
5. Blier, L., Ollivier, Y.: The description length of deep learning models. Advances in Neural Information Processing Systems **31** (2018)
6. Blum, A., Langford, J.: Pac-mdl bounds. In: Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings. pp. 344–357. Springer (2003)
7. Cayton, L.: Algorithms for manifold learning. Univ. of California at San Diego Tech. Rep **12**(1-17), 1 (2005)
8. Cover, T.M., Thomas, J.A.: Elements of information theory. John Wiley & Sons (2012)
9. Deng, L.: The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine **29**(6), 141–142 (2012)
10. Fefferman, C., Mitter, S., Narayanan, H.: Testing the manifold hypothesis. Journal of the American Mathematical Society **29**(4), 983–1049 (2016)
11. Galbrun, E.: The minimum description length principle for pattern mining: a survey. Data mining and knowledge discovery **36**(5), 1679–1727 (2022)
12. Gao, T., Jojic, V.: Degrees of freedom in deep neural networks. In: Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence. pp. 232–241. UAI'16, AUAI Press, Arlington, Virginia, USA (2016)
13. Gheorghiu, S., Coppens, M.O.: Heterogeneity explains features of "anomalous" thermodynamics and statistics. Proceedings of the National Academy of Sciences **101**(45), 15852–15856 (2004)
14. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), `http://www.deeplearningbook.org`
15. Grünwald, P.: Minimum description length tutorial. Advances in minimum description length: Theory and applications **5**, 1–80 (2005)
16. Grünwald, P., Langford, J.: Suboptimal behavior of bayes and mdl in classification under misspecification. Machine Learning **66**, 119–149 (2007)
17. Grünwald, P., Roos, T.: Minimum description length revisited. International Journal of Mathematics for Industry **11**(01), 1930001 (2019). https://doi.org/10.1142/S2661335219300018, `https://doi.org/10.1142/S2661335219300018`
18. Hansen, M.H., Yu, B.: Minimum description length model selection criteria for generalized linear models. Lecture Notes-Monograph Series pp. 145–163 (2003)
19. Hardt, M., Recht, B., Singer, Y.: Train faster, generalize better: Stability of stochastic gradient descent. In: International conference on machine learning. pp. 1225–1234. PMLR (2016)
20. He, H., Su, W.J.: The local elasticity of neural networks. arXiv preprint arXiv:1910.06943 (2019)
21. Helmbold, D.P., Long, P.M.: On the inductive bias of dropout. The Journal of Machine Learning Research **16**(1), 3403–3454 (2015)
22. Hinton, G.E., Van Camp, D.: Keeping the neural networks simple by minimizing the description length of the weights. In: Proceedings of the sixth annual conference on Computational learning theory. pp. 5–13 (1993)
23. Hu, B., Rakthanmanon, T., Hao, Y., Evans, S., Lonardi, S., Keogh, E.: Using the minimum description length to discover the intrinsic cardinality and dimensionality of time series. Data Mining and Knowledge Discovery **29**, 358–399 (2015)

24. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. PMLR (2015)
25. Jaynes, E.: Where do we stand on maximum entropy? The Maximum Entropy Formalism pp. 15–118 (1978)
26. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
27. Krogh, A., Hertz, J.A.: A simple weight decay can improve generalization. In: Advances in neural information processing systems. pp. 950–957 (1992)
28. Li, C., Farkhoor, H., Liu, R., Yosinski, J.: Measuring the intrinsic dimension of objective landscapes. In: International Conference on Learning Representations
29. Luo, P., Wang, X., Shao, W., Peng, Z.: Towards understanding regularization in batch normalization. arXiv preprint arXiv:1809.00846 (2018)
30. MacKay, D.J., Mac Kay, D.J.: Information theory, inference and learning algorithms. Cambridge university press (2003)
31. Martin, C.H., Mahoney, M.W.: Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. arXiv preprint arXiv:1810.01075 (2018)
32. Martin, C.H., Mahoney, M.W.: Heavy-tailed universality predicts trends in test accuracies for very large pre-trained deep neural networks. In: Proceedings of the 2020 SIAM International Conference on Data Mining. pp. 505–513. SIAM (2020)
33. Morcos, A.S., Barrett, D.G.T., Rabinowitz, N.C., Botvinick, M.: On the importance of single directions for generalization. CoRR (2018), `http://arxiv.org/abs/1803.06959v4`
34. Myung, J.I., Navarro, D.J., Pitt, M.A.: Model selection by normalized maximum likelihood. Journal of Mathematical Psychology **50**(2), 167–179 (2006)
35. Neyshabur, B., Tomioka, R., Srebro, N.: In search of the real inductive bias: on the role of implicit regularization in deep learning. arXiv preprint arXiv:1412.6614 (2014)
36. Pennebaker, W.B., Mitchell, J.L.: JPEG: Still image data compression standard. Springer Science & Business Media (1992)
37. Rissanen, J.: Modeling by shortest data description. Automatica **14**(5), 465–471 (1978)
38. Rissanen, J.: A universal prior for integers and estimation by minimum description length. The Annals of statistics **11**(2), 416–431 (1983)
39. Rissanen, J.: MDL denoising. IEEE Transactions on Information Theory **46**(7), 2537–2543 (2000)
40. Rissanen, J.: Strong optimality of the normalized ml models as universal codes and information in data. IEEE Transactions on Information Theory **47**(5), 1712–1717 (2001)
41. Santurkar, S., Tsipras, D., Ilyas, A., Madry, A.: How does batch normalization help optimization? arXiv preprint arXiv:1805.11604 (2018)
42. Song, H., Kim, M., Park, D., Shin, Y., Lee, J.G.: Learning from noisy labels with deep neural networks: a survey. IEEE Transactions on Neural Networks and Learning Systems (2022)
43. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)
44. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 31 (2017)

45. Thompson, R.C.: Principal submatrices ix: Interlacing inequalities for singular values of submatrices. Linear Algebra and its Applications **5**(1), 1–12 (1972)

46. Tishby, N., Zaslavsky, N.: Deep learning and the information bottleneck principle. In: 2015 ieee information theory workshop (itw). pp. 1–5. IEEE (2015)

47. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al.: Scipy 1.0: fundamental algorithms for scientific computing in python. Nature methods **17**(3), 261–272 (2020)

48. Vitányi, P.M., Li, M.: Minimum description length induction, bayesianism, and kolmogorov complexity. IEEE Transactions on information theory **46**(2), 446–464 (2000)

49. Wei, J., Zhu, Z., Cheng, H., Liu, T., Niu, G., Liu, Y.: Learning with noisy labels revisited: A study using real-world human annotations. In: International Conference on Learning Representations (2022), `https://openreview.net/forum?id=TBWA6PLJZQm`

50. Yamanishi, K.: A decision-theoretic extension of stochastic complexity and its applications to learning. IEEE Transactions on Information Theory **44**(4), 1424–1439 (1998)

51. Yoshida, Y., Miyato, T.: Spectral norm regularization for improving the generalizability of deep learning. arXiv preprint arXiv:1705.10941 (2017)

52. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization. In: International Conference on Learning Representations (2017), `https://openreview.net/forum?id=Sy8gdB9xx`

53. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. Communications of the ACM **64**(3), 107–115 (2021)

54. Zhao, K., Musolesi, M., Hui, P., Rao, W., Tarkoma, S.: Explaining the power-law distribution of human mobility through transportationmodality decomposition. Scientific reports **5**(1), 1–7 (2015)

8 Appendixes

8.1 Finding a finite precision network that overfits

Memorizing data with Neural Networks Two-layer ReLU feed forward neural networks can do this for given, arbitrary sample size n, data in arbitrary dimension d with surprising ease: as stated in [52], Theorem 1, within the set of such networks $\mathcal{N}$ with at least $2n + d$ parameters, there is at least one $N \in \mathcal{N}$ that will be able to compress y perfectly, by expressing it in terms of x. This is *not* at odds with the MDL principle (i) since MDL states that compressibility of data without a rule is *unlikely*, rather than impossible (ii) the description lenght of the network is not taken into account (see[33]).

The proof of theorem 1 in [52], implicitly assumes infinite precision in the weights of the network, which would take up infinite, and thus unavailable, space. For completeness, we briefly describe the gist of the proof in [52], before adapting it to the finite precision case.

The proof rests on a Lemma that constructs a matrix A that is lower-triangular and has non-zero and distinct real diagonal elements: the first differences of an increasing sequence. A is hence non-singular, since the diagonal elements of a triangular matrix are its singular values. The authors then proceed to stating the overfitting problem, for a 2-layer ReLU network, as the solution of linear system in a matrix B. This matrix can be made of type A via a judicious choice of network parameters a and b. Precisely, one needs to chose a, b such that for every sample x_j we have $a^\top x_j < a^\top x_{j+1}$. This can always be done for distinct x_i, by the Archimedian property of the reals. It remains to select b_j such that $a^\top x_j < b_j < a^\top x_{j+1}$, which can be done because $\mathbb{R}$ is complete. The remaining parameters of the network w are precisely the solutions of the system in B.

Memorizing data with finite-precision Neural Networks In finite precision, this cannot be done in general. The number of significant figures of $a^\top x_j$ is at most that of x_j. The problem of finding constants a and b such that we can place a b between every two $a^\top x_j$ can be done surely by picking b with one more significant figure than x_j. The number of significant figures in w, on the other hand, depends on that of y as well: it is the minimum between the number of significant figures of x plus one, and the number of significant figures of y.

We thus have the following proposition:

Proposition 4. *In order to be able to overfit data x, y with s_x, s_y significant figures, it suffices a neural network with n parameters with one more significant digit than x, d parameters with the same number of significant digits as x, and n parameters with the same precision as y for an expected number of bits of* $(n(s_x + 1) + ds_x + ns_y) \log_2 10$.

Memorizing cifar-10 Typically, the number of significant figures in ML pipelines is fixed, and it is the same for data and for weights (a 32 bit float). In the unlikely

event that data is too closely packed (some data only differs by one in the last significant digit), then there is no guaranteed overfit.

Incidentally, this precisely gives a lower bound to the parametric complexity of the model "2-layer ReLU networks": they will be able to perfectly fit data with precision s_x, s_y if the number of parameters satisfies the constraints above.

Note that the number of significant digits in 8 bit images is 3, and the number of significant digits in 10 classification choices is 1. If the model can be compressed to less than $\left(6 \times 10^4 \times (3+1) + 32^2 \times 3 + 6 \times 10^4 \times 1\right) \log_2 10$, which amounts to about 125 kB, it cannot thus be expected to overfit.

8.2 Deriving the local approximation, alternative derivation

Within each local patch the Jacobian J has a singular vector decomposition $J = U\Sigma V^\top$. Assume without loss of generality that $J \in \mathbb{R}^{n\times m}$. Then $U \in \mathbb{R}^{m\times m}$ is orthogonal, $\Sigma \in \mathbb{R}^{m\times n}$ is a diagonal matrix and $V \in \mathbb{R}^{n\times n}$ is orthogonal as well. The entropy $H(V^\top \delta X) = H(\delta X)$ since V^T has determinant one everywhere. In the same way, $H(J\delta X) = H(U^\top J\delta X) = H(\Sigma V^\top \delta X)$. In the case of an embedding, the matrix Σ has a number of zero components along its diagonal, which will, upon multiplying by $V^\top \delta X$ produce a vector that has a number of zero components independently of the other components. Hence, the entropy of the zero part and the nonzero part is the same as the entropy of the nonzero part. So the entropy above is just the entropy of the non-zero components after acting on data with $V^\top$. Explicitly, $H(\sigma_1 v_1^\top \delta X, \ldots, \sigma_k v_k^\top \delta X, 0, \ldots, 0) = H(\sigma_1 v_1^\top \delta X, \ldots, \sigma_k v_k^\top \delta X)$. Plugging into our objective we obtain

$$\begin{aligned}\max_J \lambda H(J\delta X) - H(J\Delta) &= \max_\sigma \lambda H(\sigma_1 v_1^\top \delta X, \ldots, \sigma_k v_k^\top \delta X) - H(\sigma_1 v_1^\top \Delta, \ldots, \sigma_k v_k^\top \Delta) \\ &\geq \max_\sigma \left\{ \max_i \lambda H(\sigma_i v_i^\top \delta X) - H(\sigma_1 v_1^\top \Delta, \ldots, \sigma_k v_k^\top \Delta) \right\}\end{aligned}$$

Noting that

$$\begin{aligned}H(\sigma_1 v_1^\top \Delta, \ldots, \sigma_k v_k^\top \Delta) &\leq \sum_i H(\sigma_i v_i^\top \Delta) \\ &= \sum_i H(\Delta_i) + \log \sigma_i\end{aligned}$$

where we called $v_i^\top \Delta := \Delta_i$. Doing the same for δX and plugging in our objective above, we obtain

$$\begin{aligned}\max_J \lambda H(J\delta X) - H(J\Delta) &\geq \max_\sigma \left\{ \max_i \lambda H(\sigma_i v_i^\top \delta X) - H(\sigma_1 v_1^\top \Delta, \ldots, \sigma_k v_k^\top \Delta) \right\} \\ &= \max_\sigma \left\{ \lambda \max_i \{\log \sigma_i + H(\delta X_i)\} - H(\sigma_1 v_1^\top \Delta, \ldots, \sigma_k v_k^\top \Delta) \right\} \\ &\geq \max_\sigma \left\{ \lambda \max_i \{\log \sigma_i + H(\delta X_i)\} - \left(\sum_i H(\Delta_i) + \log \sigma_i \right) \right\}\end{aligned}$$

Note now that we can do three things to maximize this local lower bound: imagine that we start training and there is one component of the data that happens to have an image with larger entropy; assuming that the singular values at start are the same, then this is because we are more aligned with the data. And so we will promote this alignment by increasing both the singular value along that direction and rotating it in order to improve the alignment.

The last terms are interesting as well: in order to reduce them and thus increase the lower bound, we can do two things: reduce the mean entropy of the projections of noise onto the singular directions and reduce the mean logarithm of the singular values of the Jacobian. Although the latter can be done without restriction, the former depends on the local shape of noise. Without further assumptions, the only sure way to reduce the mean entropy is to remove dimensions altogether.

But note that since the logarithm can be very negative, it's even better to *keep* the dimensions and just focus on decreasing the singular values to as close to zero as possible.

8.3 Controlling the approximation error of the local objective

We now show that a given $x_1, \ldots, x_N$ and an error budget E, a set of radii can be chosen such that the maximum linear approximation error does not exceed it, and these radii are inversely proportional to the largest principal singular value of the point Hessian matrices. Conversely, for a compact domain, a set of radii can be chosen such that every point is inside one of the neighborhoods of the $x_1, \ldots, x_N$ that minimizes the total approximation error.

Intuitively, since the Hessian matrix at a point controls the curvature, the curvature along the maximum curvature direction controls how far we are able to go away from the point while not changing the Jacobian too much.

Proposition 5. *Let $f : A \subseteq \mathbb{R}^n \to B \subseteq \mathbb{R}^m$ be analytical, with A compact and $x_1, \ldots, x_N \subseteq A$. Then given $E^k > 0$, a set of balls $\{V_k\}_{k=1\cdots N}$ centered at x_k and with radius r_k can be chosen such that the approximation error is upper bounded by*

$$\forall_{k=1\cdots N},\ \sup_{\substack{w\in V_k \\ i=1\cdots m}} \frac{1}{2}\sigma_1(\nabla^2 f^i|_w)\cdot\|r_k\|^2 = E^k$$

where $\sigma_1(\nabla^2 f^i|_w)$ is the first singular value of the Hessian matrix of the component f^i calculated at $w \in V_k$.

Proof. For each component f^i of f, Taylor's theorem states that the approximation error of $f^i(x_k+r_k)-f^i(x_k) \approx \left(\nabla f^i|_{x_k}\right) r_k$, along a radius r_k, in Lagrangean form, is $\frac{1}{2}{r_k}^\top \left(\nabla^2 f^i|_w\right) r_k$, where w is a point between $x_k, x_k + r_k$. The approx-

imation error is thus

$$\begin{aligned}\frac{1}{2}{r_k}^\top \left(\nabla^2 f^i|_w\right) r_k &= \frac{1}{2}\frac{{r_k}^\top \left(\nabla^2 f^i|_w\right) r_k}{{r_k}^\top r_k} \cdot \|r_k\|^2 \\ &\leq \frac{1}{2}\sigma_1(\nabla^2 f^i|_w) \cdot \|r_k\|^2 \\ &\sup_{\substack{w\in V_k \\ i=1\cdots m}} \frac{1}{2}\sigma_1(\nabla^2 f^i|_w) \cdot \|r_k\|^2\end{aligned}$$

, where we used the definition of the first singular value in terms of the Rayleigh quotient to establish this result on the sup norm. A result that holds for other norms follows from convexity of the norms and the bound on each of the components of the vector of the Hessian matrices.

To see the converse, consider that given a compact set and a point, there is always a ball that contains it. Hence so would a union of such balls. Since each of the radii sets an upper bound for the local approximation error, with $\sup_{\substack{w\in V_k \\ i=1\cdots m}} \frac{1}{2}\sigma_1(\nabla^2 f^i|_w) := \sigma_1^k$, we can write the total error as

$$\min_{r_k} \sum_{k=1}^{N} \sigma_1^k \cdot r_k^2$$

with the constraint that $A \subseteq V_1 \cup \cdots \cup V_N$.

8.4 Combining local objectives

In order to maximize the combined local objective

$$\max_{\sigma} \left\{\lambda M \mathbb{E}\left[\log \sigma\right] + \lambda H(X) - H(\Delta) - \bar{N} M \mathbb{E}\left[\log \sigma\right]\right\}$$

and aiming at an expression for the spectral distribution, we follow the strategy highlighted in 4.2, which we repeat below for completeness, for the combined objective

- aligning J with δX and then maximizing the logarithm of the singular values in the non-zero dimensions: if δX is locally low-dimensional, the singular values that get maximized are few.
- aligning J with Δ and then minimizing the logarithm of the singular values in the non-zero dimensions: since Δ tends to be relatively high-dimensional, all singular values of J tend to be minimized.

Aligning the Jacobian and δX implies that the entropy $H(\delta X^{i_k})$ is maximal. Using the manifold hypothesis, we assume that the maximal entropy component accounts for most of the entropy, that is $\max_{i_k} H(\delta X_k^{i_k}) \approx H(\delta X_k)$. As explained in 4.2, the maximal entropy component does not necessarily correspond to the

maximal singular value: during training, singular spaces corresponding to higher-order singular values will also be "selected". Taking this "selection" as a one-sample estimate of the mean justifies replacing the maximization over i_k in the expression above with $\mathbb{E}[\log \sigma_k] + H(\delta X_k)$. Under the assumption of cross-patch independence, we have $\sum_{i=1}^{M} H(\delta X_k) = H(X)$ and similarly for Δ. The expression below follows from linearity of expectation:

$$\max_{\sigma} \left\{ \lambda M \mathbb{E}\left[\log \sigma\right] + \lambda H(X) - H(\Delta) - \bar{N} M \mathbb{E}\left[\log \sigma\right] \right\}$$

Since each of the terms in the sum above is negative, this expression is non-positive. At the maximum, we thus have

$$\mathbb{E}\left[\log \sigma\right] = \frac{H(\Delta) - \lambda H(X)}{M(\lambda - \bar{N})} \tag{6}$$

8.5 Experimental setup

The experimental setup follows [52] closely. We investigate two image classification datasets, namely the MNIST dataset [**?**] and the CIFAR10 dataset [26]. Both datasets are composed of 50,000 training and 10,000 validation images, distributed across 10 different classes. In CIFAR10, each image in the dataset has dimensions of 32x32, with 3 color channels. To scale the pixel values into the range of [0, 1], we normalize them by dividing each value by 255. Additionally, we center crop the images to obtain a size of 28x28, and normalize them by subtracting the mean and dividing the adjusted standard deviation independently for each image, adapting the per_image_whitening function in Tensorflow [1], as presented in[52]. The same procedure adapted to one channel, except center cropping which is unnecessary since MNIST images are 28x28, is performed on MNIST.

On both datasets, we use two common deep architectures, which were adapted to smaller image sizes/single-channel images: a simplified Inception model [44] and Alexnet [26]. As in [52], the simplified Inception model uses a combination of 1x1 and 3x3 convolution pathways, while the simplified Alexnet is constructed using two (convolution 5x5 $\rightarrow$ max-pool 3x3 $\rightarrow$ local-response-normalization) modules followed by two fully connected layers with 384 and 192 hidden units, respectively. We utilize a 10-unit linear layer for prediction, and to calculate the point Jacobians. All architectures employ the standard rectified linear activation functions (ReLU).

We also study two fully connected multi-layer perceptrons (MLPs): one having a hidden layer with 512 units, the other having three hidden layers of the same size.

For all experiments, we train the models using SGD with a momentum of 0.9, using an initial learning rate of 0.01. We apply a decay factor of 0.95 per epoch to adjust the learning rate, and train the models without weight decay, dropout, or any other explicit regularization techniques.

In all experiments, we calculate the Jacobian at the linear layer, using automatic differentiation with Pytorch's torch.autograd.functional.jacobian method. The point Jacobian spectrum is calculated at all training and test examples using Pytorch's torch.linalg.svdvals, which is a port of Numpy's.

The experimental spectral distributions were split at the first deepest trough. The lognormal fit of each modality, the probabilty plots and line of best fit were calculated using scipy [47].

8.6 A few fundamental results in Information theory

We repeat a few results in the main paper for readability, while extending some of the arguments.

Preliminaries and notation source code $C(X)$ (simply "code" from now, and often denoted C when there is no risk of ambiguity) for a random variable X is a function from $\mathcal{X}$ the range of X to $\mathcal{D}^*$ the set of finite strings of a d-ary alphabet $\mathcal{D}$, associating $x \in \mathcal{X}$ to a codeword $C(x)$. The *length* of the codeword $l(x)$ is the number of elements in $C(x)$, and the expected code length is $L(X) := \mathbb{E}_X[l(x)]$. A code is said to be non-singular if every $x \in \mathcal{X}$ maps to an unique element of $\mathcal{D}^*$. An extension $C^*$of code C codes sequences $x_1x_2 \cdots x_n$of elements of $\mathcal{X}$ as the concatenation of $C(x_1)C(x_2) \cdots C(x_n)$. A code is said to be uniquely encoded if its extension is non-singular. Since it every element in $\mathcal{X}$ is unambiguously encoded with a unique string, non-singular codes allow us to losslessly compress data.

Optimal codelength and irregular data Results (ii) and (iii) crucially rest on the Kraft-Macmillan inequality:

Theorem 2 (Kraft-Macmillan inequality). *For any uniquely decodable code C over an alphabet of size D, the codeword lengths $l_1, l_2, \ldots, l_m$ must satisfy the inequality*

$$\sum_i D^{-l_i} \leq 1 \tag{7}$$

Conversely, given a set of codeword lengths that satisfy 7, there exists a uniquely decodable code with these word lengths.

The idea of the proof for prefix codes (known as the Kraft inequality) is that prefix codes from a D - adic alphabet can be seen as the childless nodes on a rooted tree. No prefix code can then be among the descendants of another. Hence, the sum of the descendants of prefix codes cannot exceed the number of leaves of the tree: $\sum_i D^{l_{max}-l_i} \leq D^{l_{max}}$. The converse is established simply by noting that lengths that satisfy 7 can be placed on rooted tree. If they couldn't there would be more words of length l_i than descendants of non-used codes; but since the total mass at every level is constant, this cannot happen.

Theorem 3 (Optimal code length). *The expected code length for any uniquely decodable code C of a random variable X over an alphabet of size D is greater than or equal to $H_D()$ the entropy calculated in base D, with equality holding iff $D^{-l_i} = p_i$*

To establish this, consider the difference between the entropy and the expected length. The result then follows from theorem 2 and non-negativity of relative entropy, which is a consequence of the concavity of logarithm (Jensen inequality). An optimal prefix code always exists (e.g. Huffman code), but for our purposes, it suffices that the Shannon-Fano code, which sets codeword lengths $l(x) = \lceil -\log p(x) \rceil$ is competitive in the sense that the probability that the expected length exceeds another code's by c bits does not exceed 2^{1-c}.

Finally, we give an informal argument to justify the statement that it is extremely unlikely that data with no regularities (with maximal entropy) can be compressed. By the Kraft-Macmillan inequality 2, for every prefix code of a random variable X over an alphabet of size D, the expected codeword length is no greater than the entropy, with equality iff the $l_i = -\log_D p_i$. Assuming X is discrete, all n events have the same probability $\frac{1}{n}$. Hence, the expected code length (per symbol) is $L \geq -\sum_{i=1}^{n} p_i \log_D p_i = \log_D n$. The lower bound is what we can achieve simply by assigning each codeword to the leaves of a D-nary tree: the best code coincides with the worst possible code, and so data cannot be compressed.

8.7 Additional figures

Point Jacobian spectrum, full spectra For the figures detailing the full spectrum, i.e. figs. 5 through 13, note that train and test distributions are the same, since the underlying distributions are the same for this relatively simple dataset. We also note that the overall shape of each distribution does not change significantly with the addition of noise. As discussed in the main text, we note the clear bimodality in all spectral distributions, comparatively higher "lognormality" of Inception, and that the addition of noise increases the mean spectrum.

Finally, note that at the beginning of training, MLP and Alexnet's predictions are very local, since there is a great number of relatively small singular values (high peak), and more so with the addition of noise. This effect is also observed in Inception, but to much less extent. Using fig. 14 as illustration, MLP and Alexnet are more *conservative* than Inception, keeping close to the image of the training examples for small perturbations. This is similar to the strategies of generalization that we commonly use (e.g. maxent). Inception, on the other hand, which generalizes better, while not being conservative at all, which suggests that it does so by focusing on the signal.

Point Jacobian spectrum MNIST Figures 3 and 4 parallel those in the main text for the MNIST dataset.

Dataset noise illustration Figure 14 provides a graphical illustration of the representation building leading to bimodality. The example is for a classifier $f : \mathbb{R}^3 \to \mathbb{R}^2$ and an autoencder g for visualization, but the overall idea extends to higher dimensions.

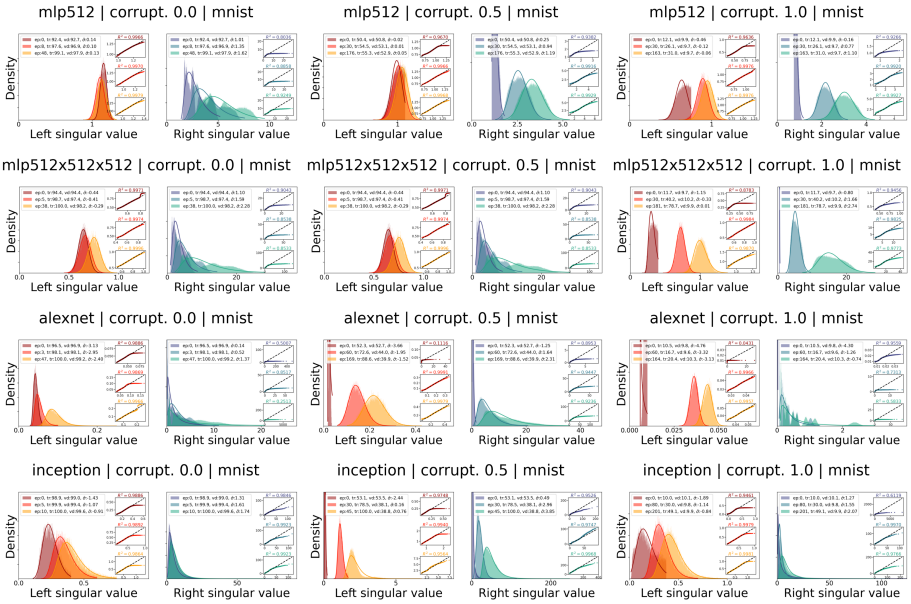

Fig. 3: Point Jacobian spectral distribution for *model* | ***label noise*** | *MNIST*, from first epoch to overfit. "Left" and "right" distributions (cf. 8.7) are represented separately for each triplet for clarity. The best fit lognormal plot is superimposed on each histogram, with the corresponding probability plot on the right, with the line of best fit (R^2 displayed on top). Legend elements, in order: epoch, training and validation accuracy, and the mean log spectrum.

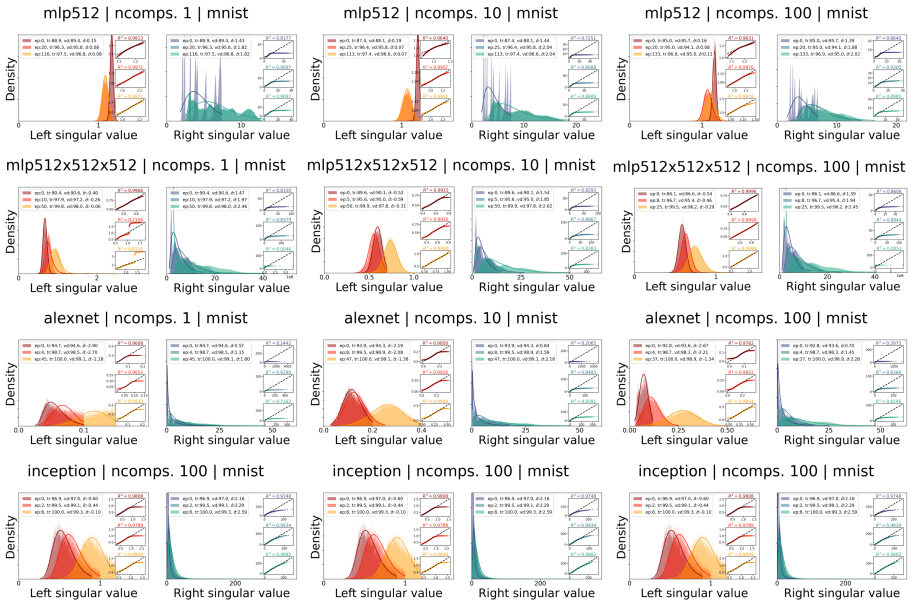

Fig. 4: Point Jacobian spectral distribution for *model* | ***nbr. pca comp.*** | *MNIST*, from first epoch to overfit. "Left" and "right" distributions (cf. 8.7) are represented separately for each triplet for clarity. The best fit lognormal plot is superimposed on each histogram, with the corresponding probability plot on the right, with the line of best fit (R^2 displayed on top). Legend elements, in order: epoch, training and validation accuracy, and the mean log spectrum.

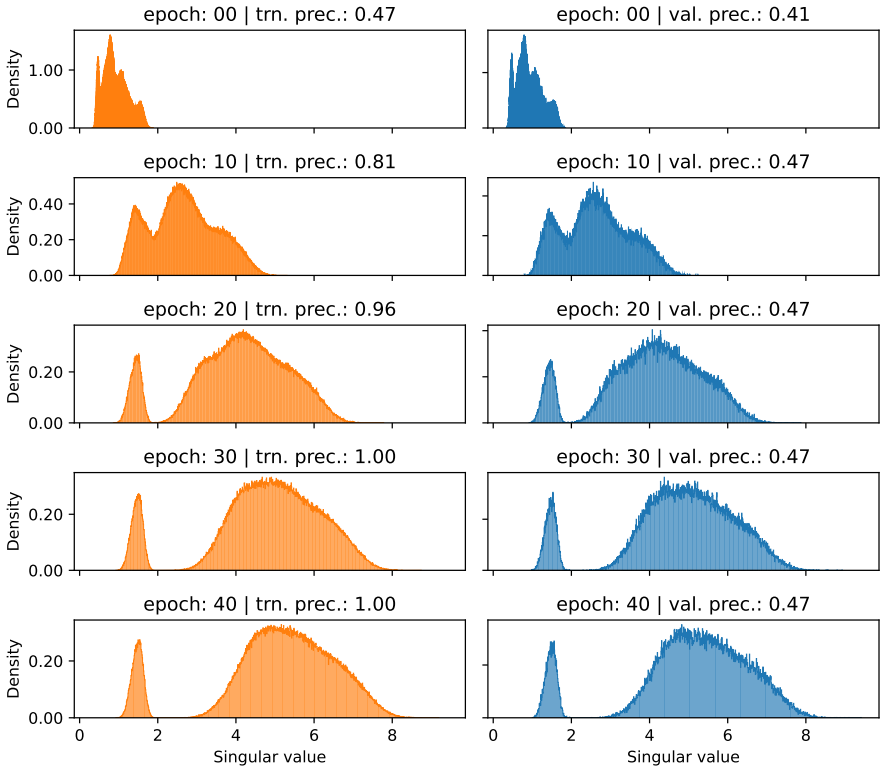

Fig. 5: Full point train (left) and validation (right) Jacobian spectrum for MLP trained on cifar-10, from first epoch to overfit, with no label noise. Horizontal scale is the same across all plot in the same column. Epochs and performance are indicated on top of each graph.

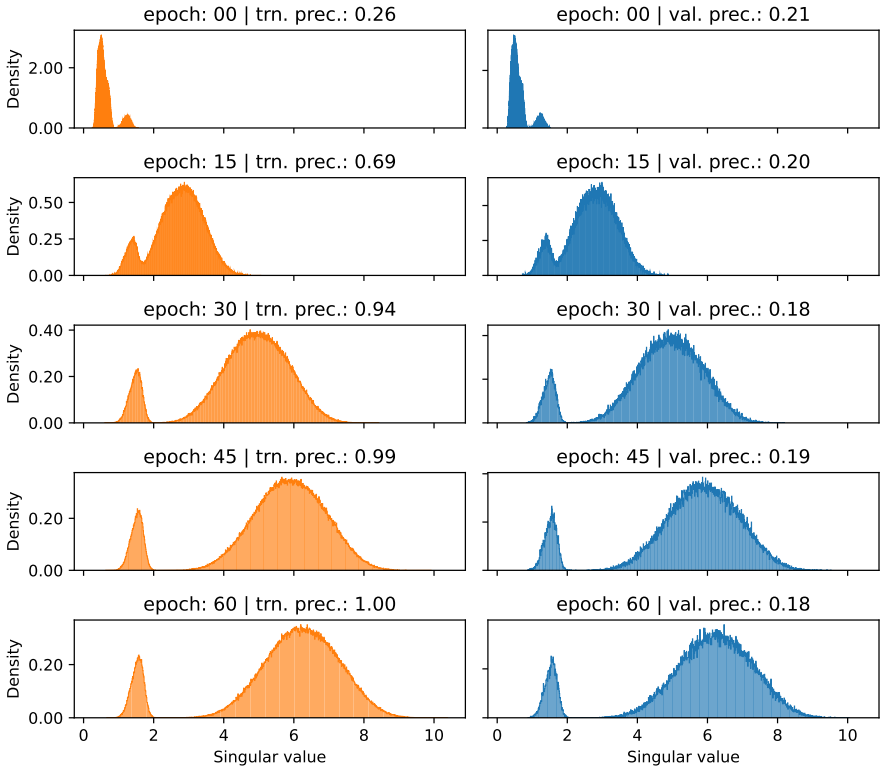

Fig. 6: Full point train (left) and validation (right) Jacobian spectrum for MLP trained on cifar-10, from first epoch to overfit, with label noise $p = 0.5$. Horizontal scale is the same across all plot in the same column. Epochs and performance are indicated on top of each graph.

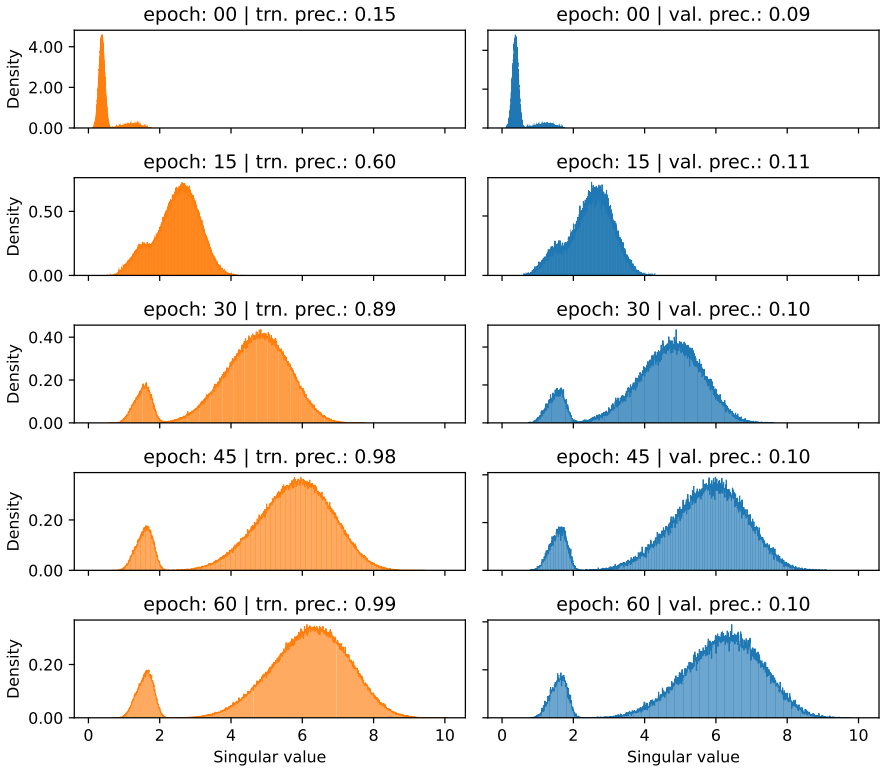

Fig. 7: Full point train (left) and validation (right) Jacobian spectrum for MLP trained on cifar-10, from first epoch to overfit, with label noise $p = 1.0$. Horizontal scale is the same across all plot in the same column. Epochs and performance are indicated on top of each graph.

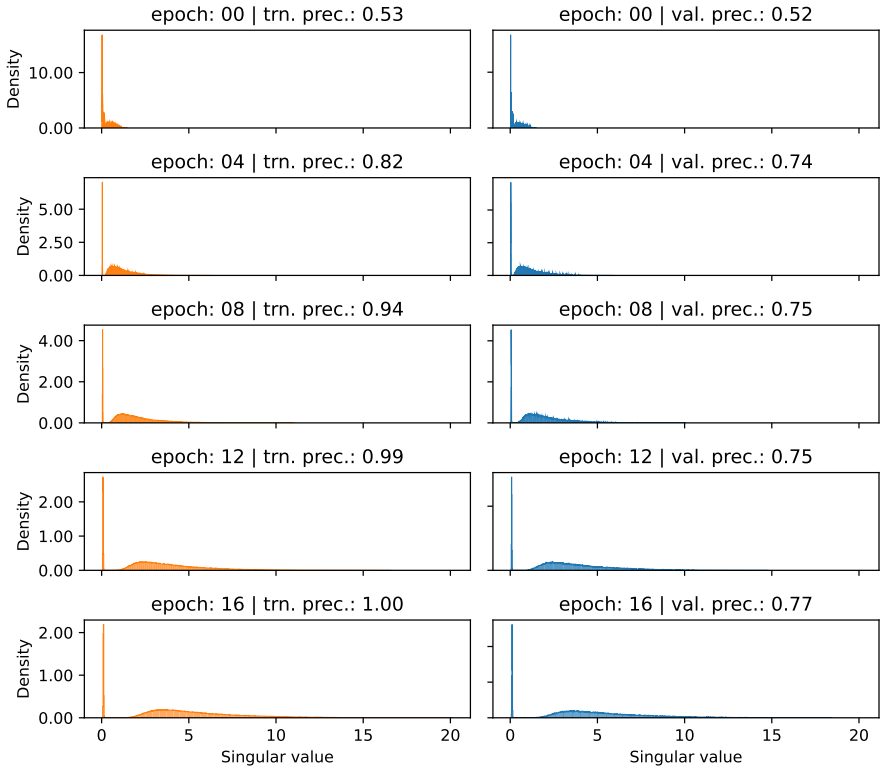

Fig. 8: Full point train (left) and validation (right) Jacobian spectrum for Alexnet trained on cifar-10, from first epoch to overfit, with label noise $p = 0.0$. Horizontal scale is the same across all plot in the same column. Epochs and performance are indicated on top of each graph.

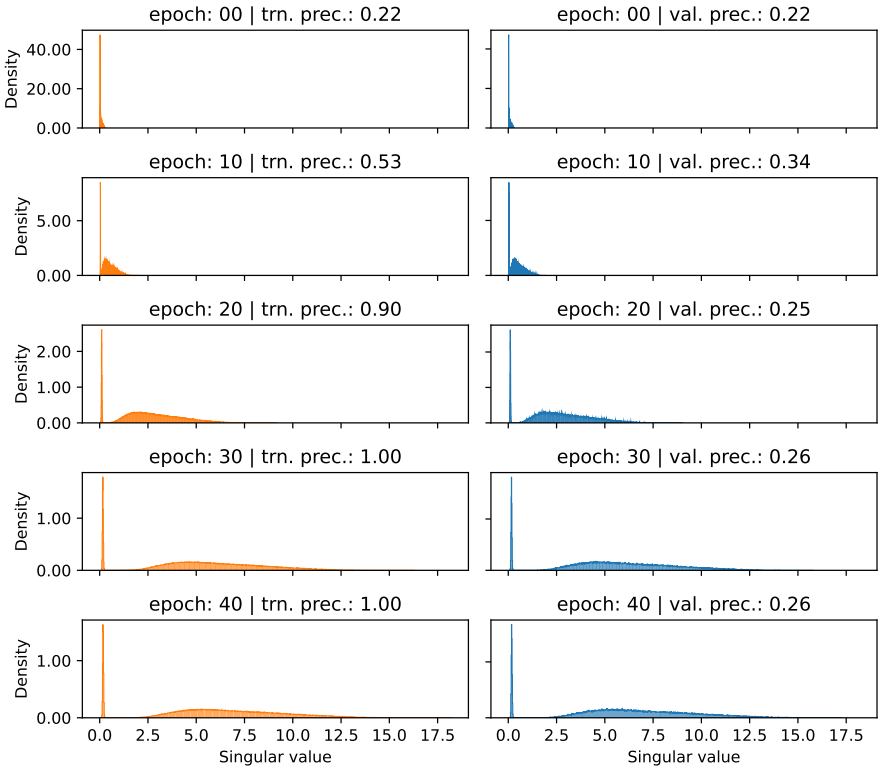

Fig. 9: Full point train (left) and validation (right) Jacobian spectrum for Alexnet trained on cifar-10, from first epoch to overfit, with label noise $p = 0.5$. Horizontal scale is the same across all plot in the same column. Epochs and performance are indicated on top of each graph.

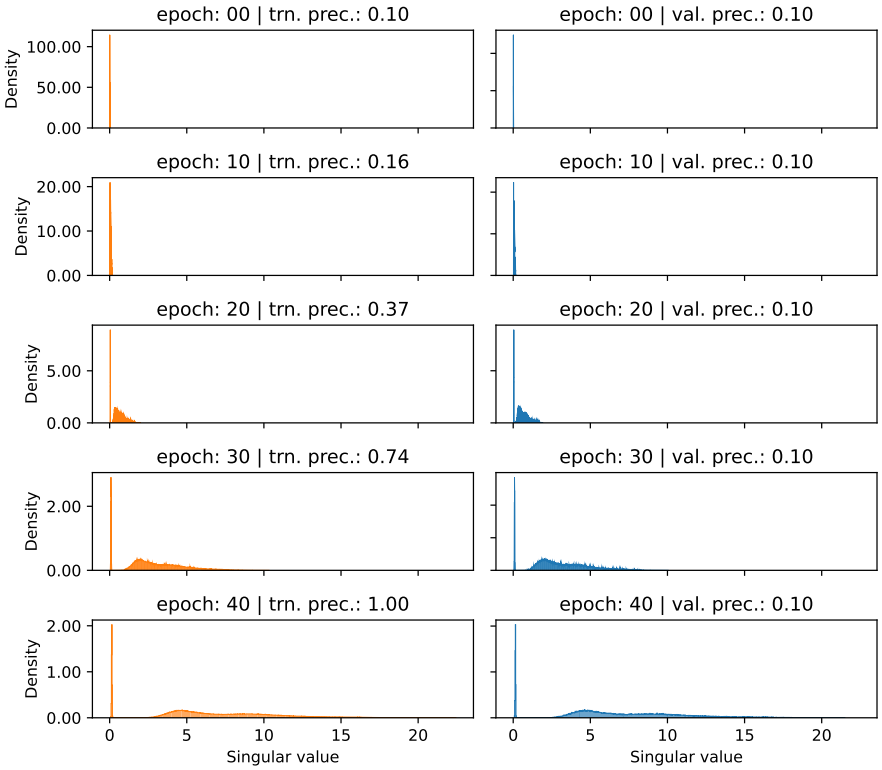

Fig. 10: Full point train (left) and validation (right) Jacobian spectrum for Alexnet trained on cifar-10, from first epoch to overfit, with label noise $p = 1.0$. Horizontal scale is the same across all plot in the same column. Epochs and performance are indicated on top of each graph.

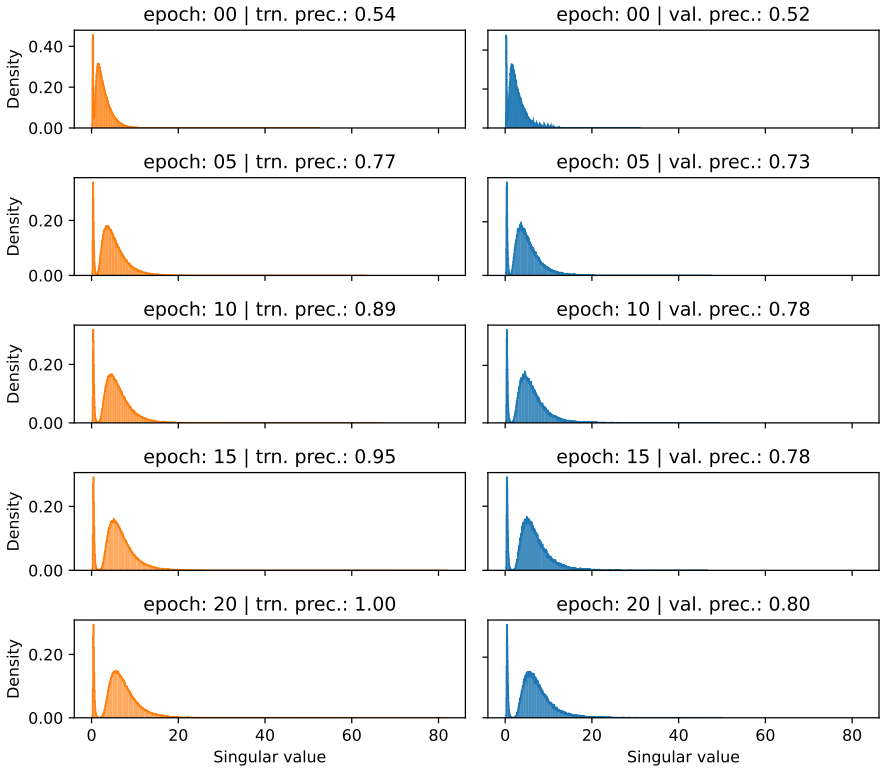

Fig. 11: Full point train (left) and validation (right) Jacobian spectrum for Inception trained on cifar-10, from first epoch to overfit, with label noise $p = 0.0$. Horizontal scale is the same across all plot in the same column. Epochs and performance are indicated on top of each graph.

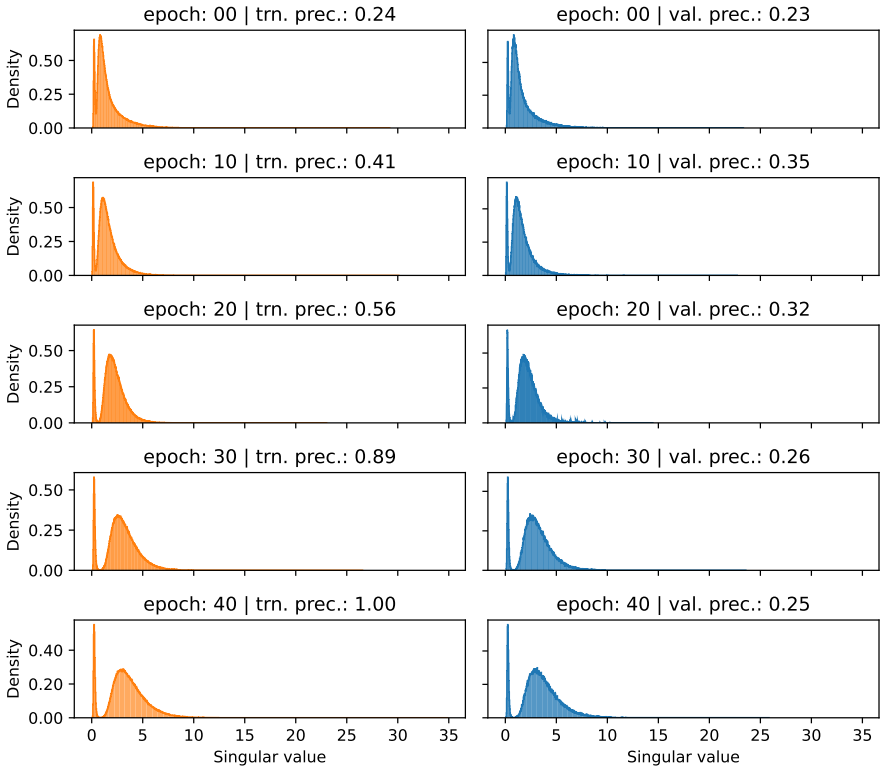

Fig. 12: Full point train (left) and validation (right) Jacobian spectrum for Inception trained on cifar-10, from first epoch to overfit, with label noise $p = 0.5$. Horizontal scale is the same across all plot in the same column. Epochs and performance are indicated on top of each graph.

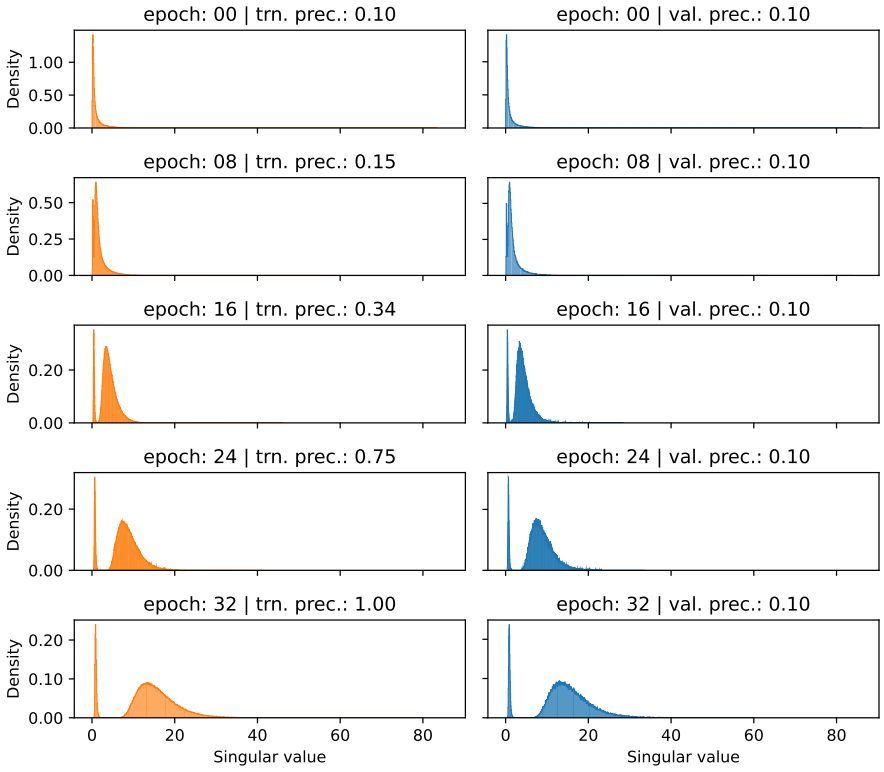

Fig. 13: Full point train (left) and validation (right) Jacobian spectrum for Inception trained on cifar-10, from first epoch to overfit, with label noise $p = 1.0$. Horizontal scale is the same across all plot in the same column. Epochs and performance are indicated on top of each graph.

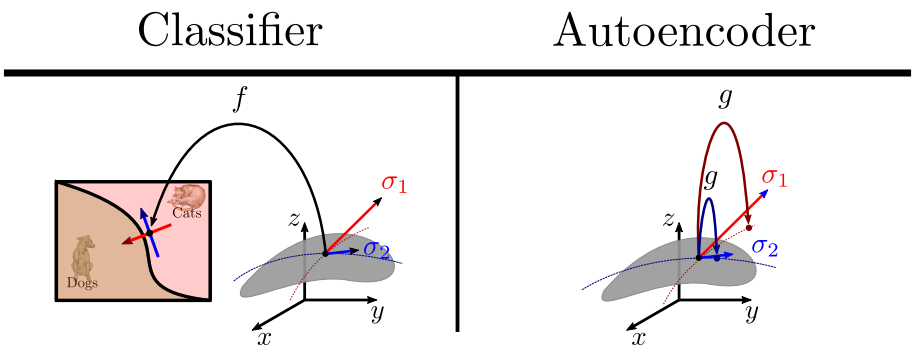

Fig. 14: Principal directions of the point Jacobian for a classifier f and an autoencoder g for three-pixel pictures of cats and dogs on the neighborhood of a given point. By definition, the norm of the change of the image through f for perturbations in first singular direction σ_1 is maximal among all directions, and similarly for the second direction in the orthogonal space to the first. Note that since the destination space is in $\mathbb{R}^2$, there are only two singular directions in the original $\mathbb{R}^3$. For each point P the two directions are the directions of respectively maximum and minimum change with respect to "cat-dog". As for the autoencoder, reconstruction is much more sensitive to perturbations along σ_1 than σ_2: changes along the latter are reconstructed as being the same image, which means that the model considers them as being noise.