Epilogue ε

Model Complexity from the Macroscopic Perspective

> *According to the hype of 1987, neural networks were meant to be intelligent models that discovered features and patterns in data. Gaussian processes in contrast are simply smoothing devices. How can Gaussian processes possibly replace neural networks? Were neural networks over-hyped, or have we underestimated the power of smoothing methods?*
>
> David MacKay [71].

Throughout this book, we've focused on deep-learning models that are very wide but also deep. Our reason for this focus is that such large neural networks with many many model parameters work extremely well in practice and thus form the foundation of the modern approach to artificial intelligence.

The success of these **overparameterized** models with far more parameters than training data has led many to simply conjecture that "more is better" when it comes to deep learning. In a more refined sense, there's mounting empirical evidence that a **scaling hypothesis** can accurately capture the behavior of deep neural networks, and its associated *scaling laws* overall point toward the optimality of the overparameterized regime.[1] The simplicity of these empirical laws recalls an earlier period in statistical physics, when a similar scaling hypothesis was conjectured to govern the behavior of certain complicated systems in statistical mechanics.[2]

[1] See, e.g., [72] for an empirical study of scaling laws in deep learning language models based on the transformer architecture. Empirically, it's observed that overparameterization is good; the optimal growth of the number of training samples $N_{\mathcal{A}}$ scales sublinearly with the growth in parameters P, though importantly they should still scale together with a power law: $N_{\mathcal{A}} \propto P^{\alpha}$ for $0 < \alpha < 1$.

[2] In physics, such scaling laws are an example of the phenomenon of *universality*, the fact that when a system has many elementary components, it can often be described by a very simple *effective theory* that's independent of the microscopic details of the underlying system [49]. The framework of the *renormalization group* then offers an explanation for how this universality arises by characterizing the flow from the microscopic to the macroscopic [38, 39]. This perhaps suggests that the analogous notion of *representation group flow* (cf. §4.6) may be able to explain the neural scaling laws of [72].

https://doi.org/10.1017/9781009023405.015 Published online by Cambridge University Press

However, the practical success of overparameterized models in deep learning appears to be in tension with orthodox machine learning and classic statistical theory. Heuristically, the *Occam's razor* principle of sparsity posits that we should favor the simplest hypothesis that explains our observations; in the context of machine learning, this is usually interpreted to mean that we should prefer models with fewer parameters when comparing models performing the same tasks. More quantitatively, we expect that models with fewer parameters will have smaller *generalization errors*, $\mathcal{E} \equiv \mathcal{L}_{\mathcal{B}} - \mathcal{L}_{\mathcal{A}}$, and will be less prone to overfit their training set $\mathcal{A}$. Vice versa, we should naively expect that overparameterized models *will* overfit their training data and generalize poorly. Thus, this orthodoxy is in direct conflict with the empirical success of overparameterized neural networks and is a big theoretical puzzle in understanding modern deep learning.[3]

In this brief epilogue, we're going to offer a resolution of this puzzle. The crux of the matter hinges on the notion of **model complexity**. On the one hand, our orthodox discussion of generalization above took a **microscopic perspective** – focusing on how a network works in terms of its explicit low-level components – and wanted to identify model complexity with model parameters. On the other hand, in this book we integrated out the model parameters and developed a **macroscopic perspective** – providing an effective theory description of the predictions of realistic fully-trained networks – for which this notion of model complexity is completely reversed.

Indeed, we motivated our effective theory approach in §0 on the basis that we would be able to find simplicity in the limit of a large number of model parameters, and from §1 to §∞ we've now seen how this hypothesis has been borne out again and again for realistic large-but-finite-width networks. Having finished all the technical calculations (outside of the appendices), we can see now that it's the depth-to-width aspect ratio,

$$r \equiv L/n, \tag{ε.1}$$

that controls the model complexity of overparameterized neural networks. To understand why, recall that this ratio emerged from our calculations as the expansion parameter or *cutoff* of our effective theory and determined how we could *truncate* the series expansion of the fully-trained distribution while still approximating the true behavior of networks with minimal error. This means that it's the number of data-dependent couplings of the truncated nearly-Gaussian distribution – and *not* the number of model parameters – that ultimately defines the model complexity in deep learning. From this macroscopic perspective, there's absolutely no conflict between the sparse intuition of Occam's razor in theory and the simplicity of the scaling hypothesis in practice.

To see how this works in even greater detail, let us recall the three main problems we discussed at the beginning of the book in §0.2 and then review how the principle of

[3] See, e.g., the extensive discussion in [73] on the difficulty of trying to understand why large neural networks generalize so well according to traditional measures of model complexity.

https://doi.org/10.1017/9781009023405.015 Published online by Cambridge University Press

sparsity enabled us to solve them.[4] Taylor-expanding the trained network output around the network's initialization (0.5),

$$z(x_\delta;\theta^\star) = z(x_\delta;\theta) + \sum_{\mu=1}^{P} (\theta^\star - \theta)_\mu \frac{dz_\delta}{d\theta_\mu} + \frac{1}{2}\sum_{\mu,\nu=1}^{P} (\theta^\star - \theta)_\mu (\theta^\star - \theta)_\nu \frac{d^2 z_\delta}{d\theta_\mu d\theta_\nu} + \cdots , \tag{ε.2}$$

we illustrated **Problem 1**, (0.6), that we might have to compute an infinite number of terms,

$$z\,, \quad \frac{dz}{d\theta}\,, \quad \frac{d^2z}{d\theta^2}\,, \quad \frac{d^3z}{d\theta^3}\,, \quad \frac{d^4z}{d\theta^4}\,, \quad \cdots\,, \tag{ε.3}$$

Problem 2, (0.7), that we have to determine the map from the initialization distribution over the model parameters to the induced initialization distribution over the network output and its derivatives,

$$p(\theta) \to p\left(z, \frac{dz}{d\theta}, \frac{d^2z}{d\theta^2}, \ldots\right), \tag{ε.4}$$

and **Problem 3**,(0.8), that we have to solve the training dynamics, which can depend on *everything*,

$$\theta^\star \equiv [\theta^\star]\left(\theta,\, z,\, \frac{dz}{d\theta},\, \frac{d^2z}{d\theta^2},\, \ldots;\, \text{learning algorithm};\, \text{training data}\right). \tag{ε.5}$$

Let us now give our detailed solutions to this problem set – expanding on the schematic explanations that we gave in §0.2 – and then carefully examine them through the lens of *model complexity*. We'll begin first with the simple infinite-width limit and then discuss the nearly-simple-but-realistic $1/n$ truncation.

Sparsity at Infinite Width

In the infinite-width limit, we can now understand our solutions as follows:

- Addressing **Problem 1**, (0.11), all the higher derivative terms vanish, and we only need to keep track of two statistical variables:

$$z\,, \quad \frac{dz}{d\theta} \quad \Longrightarrow \quad z_\delta\,, \quad \widehat{H}_{\delta_1\delta_2}. \tag{ε.6}$$

[4] In this epilogue, we'll drop layer indices on our variables to ease the notation, as everything is evaluated at the output layer; we'll also sometimes drop the neural indices when they are unimportant.

https://doi.org/10.1017/9781009023405.015 Published online by Cambridge University Press

Note that this first derivative gives the *random features* of the linear model description, cf. (10.137), and the kernel associated with these features is just the NTK.

- Addressing **Problem 2**, we found that the network output and its first derivative are statistically independent, each governed by the simple distribution (0.12):

$$\lim_{n\to\infty} p\left(z, \frac{dz}{d\theta}, \frac{d^2z}{d\theta^2}, \ldots\right) = p(z_\delta)\,\delta\left(\sum_{\mu,\nu} \lambda_{\mu\nu} \frac{dz_{\delta_1}}{d\theta_\mu}\frac{dz_{\delta_2}}{d\theta_\nu} - \Theta_{\delta_1\delta_2}\right), \tag{ε.7}$$

  where on the right-hand side, $p(z_\delta)$ is a zero-mean Gaussian distribution with its variance given by the kernel $K_{\delta_1\delta_2}$ (4.106), and the second factor is a Dirac delta function distribution that fixes the contraction of first derivatives that make up the NTK $\widehat{H}_{\delta_1\delta_2}$ to be deterministically given by the frozen NTK $\Theta_{\delta_1\delta_2}$ (9.4).

- Addressing **Problem 3**, we obtained a solution for the trained model parameters $\theta^\star$ in a *closed form* (0.13):

$$\lim_{n\to\infty} \theta^\star_\mu = \theta_\mu(t=0) - \sum_{\nu,\tilde{\alpha}_1,\tilde{\alpha}_2,i} \lambda_{\mu\nu} \frac{dz_{i;\tilde{\alpha}_1}}{d\theta_\nu} \widetilde{\Theta}^{\tilde{\alpha}_1\tilde{\alpha}_2} \left(z_{i;\tilde{\alpha}_2} - y_{i;\tilde{\alpha}_2}\right), \tag{ε.8}$$

  with the associated fully-trained network outputs $z_\delta(T) = z(x_\delta;\theta^\star)$ given in (10.28). We further showed in §10.2.2 that this fully-trained solution is independent of the algorithm used to train the network.

Combining these insights, we found that the fully-trained distribution,

$$\lim_{n\to\infty} p\Big(z(T)\Big) \equiv p\Big(z(T)\Big|y_{\tilde{\alpha}}, K_{\delta_1\delta_2}, \Theta_{\delta_1\delta_2}\Big), \tag{ε.9}$$

is a *Gaussian distribution*; the reason for writing it as a conditional distribution in this way is that the mean, (10.40), is only a function of the vector of training set labels, $y_{\tilde{\alpha}}$, and the frozen NTK matrix, $\Theta^{(L)}_{\delta_1\delta_2}$, while the variance, (10.41), is only a function of the kernel matrix, $K^{(L)}_{\delta_1\delta_2}$, and the frozen NTK matrix, $\Theta^{(L)}_{\delta_1\delta_2}$. In other words, the distribution of predictions on a test sample, $x_{\dot{\beta}}$, will depend on the test sample together with all of the training data, $\mathcal{D} = \{\dot{\beta}\} \cup \mathcal{A}$, and the shape of that data dependence is governed by the specific functional forms of the data-dependent couplings, i.e., the output-layer kernel $K(x_{\delta_1}, x_{\delta_2})$ and output-layer frozen NTK $\Theta(x_{\delta_1}, x_{\delta_2})$. Thus, the infinite-width solution (ε.9) allows for a very sparse description, depending only on a few objects in a simple way.

Near-Sparsity at Finite Width

Similarly, at large-but-finite width, we can now understand our solutions as follows:

- Addressing **Problem 1**, (0.16), all derivatives $d^k f/d\theta^k$ for $k \geq 4$ are $O(1/n^2)$, and so we only need to keep track of the statistical variables up to the third derivative:

$$z\,, \quad \frac{dz}{d\theta}\,, \quad \frac{d^2z}{d\theta^2}\,, \quad \frac{d^3z}{d\theta^3} \quad \Longrightarrow \quad z_\delta\,, \widehat{H}_{\delta_1\delta_2}\,, \widehat{\mathrm{d}H}_{\delta_0\delta_1\delta_2}\,, \widehat{\mathrm{dd_I}H}_{\delta_0\delta_1\delta_2\delta_3}\,, \widehat{\mathrm{dd_{II}}H}_{\delta_1\delta_2\delta_3\delta_4}. \tag{ε.10}$$

https://doi.org/10.1017/9781009023405.015 Published online by Cambridge University Press

Here, we note that the NTK, dNTK, and ddNTKs capture all the terms up to the third derivative in our Taylor expansion (ε.2) for a gradient-based learning update, cf. (∞.4), (∞.6), and (∞.9).

- Addressing **Problem 2**, (0.17), we evaluated the distribution of all these statistical variables and found that its joint distribution is nearly-Gaussian:

$$p\left(z, \frac{dz}{d\theta}, \frac{d^2z}{d\theta^2}, \ldots\right) = p\left(z, \widehat{H}, \widehat{\mathrm{d}H}, \widehat{\mathrm{dd_I}H}, \widehat{\mathrm{dd_{II}}H}\right) + O\left(\frac{1}{n^2}\right). \tag{ε.11}$$

- Addressing **Problem 3**, (0.18), we were able to use perturbation theory to solve the nonlinear training dynamics and evaluate the predictions of fully-trained networks at finite width:

$$z(x_\delta;\theta^\star) = [z(x_\delta;\theta^\star)]\left(z, \widehat{H}, \widehat{\mathrm{d}H}, \widehat{\mathrm{dd_I}H}, \widehat{\mathrm{dd_{II}}H}; \text{algorithm projectors}\right). \tag{ε.12}$$

Here, the details of the *algorithm dependence* of the prediction is manifest, captured entirely by a handful of algorithm projectors, cf. (∞.142)–(∞.147).

Combining these insights, we found that the fully-trained distribution,

$$p\Big(z(T)\Big) \equiv p\Big(z(T)\Big|y, G, H, V, A, B, D, F, P, Q, R, S, T, U\Big) + O\left(\frac{1}{n^2}\right), \tag{ε.13}$$

is a *nearly-Gaussian distribution*; its statistics are entirely described by the conditional variables listed here, though we've suppressed the sample indices in order to fit them all on one line.[5] In addition to the metric G and the NTK mean H, in this conditioning we're accounting for the finite-width data-dependent couplings arising from our decompositions of the four-point connected correlator of preactivations $\mathbb{E}\left[zzzz\right]_{\text{connected}}$, (4.77), the NTK–preactivation cross correlator $\mathbb{E}\left[\widehat{\Delta H}zz\right]$, (8.67), the NTK variance $\mathbb{E}\left[\widehat{\Delta H}^2\right]$, (8.82), the dNTK–preactivation cross correlator $\mathbb{E}\left[\widehat{\mathrm{d}H}z\right]$, (11.45), and the means of the ddNTKs $\mathbb{E}\left[\widehat{\mathrm{dd_I}H}\right]$ and $\mathbb{E}\left[\widehat{\mathrm{dd_{II}}H}\right]$, (∞.11) and (∞.12). Importantly, all of these finite-width tensors at $O(1/n)$ are functions of exactly four input samples each, e.g., for the four-point vertex we have $V_{(\delta_1\delta_2)(\delta_3\delta_4)} \equiv V(x_{\delta_1}, x_{\delta_2}, x_{\delta_3}, x_{\delta_4})$, and the specific functional forms of these data-dependent couplings determine the overall data dependence of the distribution. Thus, though slightly more complicated than the infinite-width description (ε.9), the solution truncated to $O(1/n)$, (ε.13), is a nearly-sparse description, depending only on two-hands-full of objects in a nearly-simple way.

[5] Note that we've also suppressed the algorithm projectors in this conditioning, presuming that we are considering a fixed learning algorithm: once an algorithm is fixed, the projectors can only be fixed functions of any training set tensors that contain $O(1)$ terms – i.e., the metric submatrix and the NTK mean submatrix, both evaluated on the training set only, $\widetilde{G}_{\tilde{\alpha}_1\tilde{\alpha}_2}$ and $\widetilde{H}_{\tilde{\alpha}_1\tilde{\alpha}_2}$ – and of the global learning rate, η, see, e.g., (∞.142)–(∞.147) for gradient descent. Thus, the algorithm dependence is taken care of entirely by the tensors we're conditioning on already.

https://doi.org/10.1017/9781009023405.015 Published online by Cambridge University Press

Model Complexity of Fully-Trained Neural Networks

These effective theory results, (ε.9) and (ε.13), should make it clear that for overparameterized neural networks, it is no longer appropriate to identify the number of model parameters with the model complexity. Consider a fixed combined training and test dataset of size $N_{\mathcal{D}}$:

- For the Gaussian distribution, (ε.9), that describes an ensemble of fully-trained infinite-width networks, we only need

$$n_{\text{out}} N_{\mathcal{A}} + \left[\frac{N_{\mathcal{D}}(N_{\mathcal{D}}+1)}{2}\right] + \left[\frac{N_{\mathcal{D}}(N_{\mathcal{D}}+1)}{2}\right] = O\left(N_{\mathcal{D}}^2\right) \tag{ε.14}$$

  numbers in order to completely specify the distribution, with each term corresponding to the numbers needed to enumerate $y_{i;\tilde{\alpha}}$, $K_{\delta_1\delta_2}$, and $\Theta_{\delta_1\delta_2}$, respectively.

- For the nearly-Gaussian distribution, (ε.13), that describes an ensemble of fully-trained finite-width networks with small-but-nonzero aspect ratios, $0 < r \ll 1$, we now need

$$O\left(N_{\mathcal{D}}^4\right) \tag{ε.15}$$

  numbers in order to completely specify the distribution, with the counting dominated by the finite-width tensors, each having exactly four sample indices.

Thus, while each infinite-width network has an *infinite* number of microscopic model parameters, its macroscopic data-dependent couplings are only *quadratic* in samples. Meanwhile, our finite-width networks have fewer model parameters than our infinite-width network – i.e., finite < infinite – but their macroscopic effective description is more complicated! What we have found here is a manifestation of the **microscopic-macroscopic duality**: under this duality, complexity in *parameter space* is transformed into simplicity in *sample space*, and density in *model parameters* is exchanged for sparsity in *data-dependent couplings*. In the overparameterized regime, this duality indicates that we really should identify the model complexity with the data-dependent couplings rather than the model parameters.

To further elaborate on this general point, we could imagine carrying out our finite-width $1/n$ expansion, (0.15), to higher orders as

$$p\Big(z(T)\Big) = p^{\{0\}}\Big(z(T)\Big) + \frac{p^{\{1\}}\Big(z(T)\Big)}{n} + \frac{p^{\{2\}}\Big(z(T)\Big)}{n^2} + O\left(\frac{1}{n^3}\right). \tag{ε.16}$$

To begin, now the preactivation distribution at initialization,

$$p\Big(z\Big|\mathcal{D}\Big) = \frac{1}{Z} e^{-S(z)} + O\left(\frac{1}{n^3}\right), \tag{ε.17}$$

https://doi.org/10.1017/9781009023405.015 Published online by Cambridge University Press

will be effectively described in terms of a *sextic action*,

$$\begin{aligned} S(z) \equiv & \frac{1}{2}\sum_{i=1}^{n_L}\sum_{\delta_1,\delta_2\in\mathcal{D}} g^{\delta_1\delta_2} z_{i;\delta_1} z_{i;\delta_2} \\ & - \frac{1}{8}\sum_{i_1,i_2=1}^{n_L}\sum_{\delta_1,\ldots,\delta_4\in\mathcal{D}} v^{(\delta_1\delta_2)(\delta_3\delta_4)} z_{i_1;\delta_1} z_{i_1;\delta_2}\, z_{i_2;\delta_3} z_{i_2;\delta_4} \\ & + \frac{1}{24}\sum_{i_1,i_2,i_3=1}^{n_L}\sum_{\delta_1,\ldots,\delta_6\in\mathcal{D}} u^{(\delta_1\delta_2)(\delta_3\delta_4)(\delta_5\delta_6)} z_{i_1;\delta_1} z_{i_1;\delta_2}\, z_{i_2;\delta_3} z_{i_2;\delta_4}\, z_{i_3;\delta_5} z_{i_3;\delta_6}. \end{aligned} \tag{ε.18}$$

In this action, the *sextic coupling* scales as

$$u^{(\delta_1\delta_2)(\delta_3\delta_4)(\delta_5\delta_6)} = O\left(\frac{1}{n^2}\right) \tag{ε.19}$$

and leads to a nontrivial connected six-point correlator characterized by a *six-point vertex*: $U_{(\delta_1\delta_2)(\delta_3\delta_4)(\delta_5\delta_6)}$.[6] At criticality, this connected correlator scales as

$$\frac{1}{n^2} U_{(\delta_1\delta_2)(\delta_3\delta_4)(\delta_5\delta_6)} \propto O\left(\frac{L^2}{n^2}\right), \tag{ε.20}$$

consistent with the expectations of our effective theory cutoff. Thus, we expect that the refined description, (ε.16), is accurate to order L^2/n^2 but at the cost of a significant increase in the model complexity: the counting will be dominated by the $1/n^2$ finite-width tensors – each of which has six sample indices – and so we now require

$$O\left(N_{\mathcal{D}}^6\right) \tag{ε.21}$$

numbers in order to specify all the data-dependent couplings of the distribution. In general, to achieve an accuracy of order L^k/n^k, we expect that a macroscopic description

$$p\Big(z(T)\Big) = \sum_{m=0}^{k} \frac{p^{\{m\}}\Big(z(T)\Big)}{n^m} + O\left(\frac{L^{k+1}}{n^{k+1}}\right) \tag{ε.22}$$

[6] We computed this vertex in the second layer in footnote 9 of §4.2, and we've further taught you everything that you need to know in order to extend the computation of such higher-point correlators to deeper layers and then analyze their scaling at criticality. As a checkpoint for your algebra, the single-input recursion for the six-point vertex specialized to the `ReLU` activation function is

$$U^{(\ell+1)} = \frac{1}{8}\left[C_W^{(\ell+1)}\right]^3 \left[U^{(\ell)} + 30V^{(\ell)}K^{(\ell)} + 44\left(K^{(\ell)}\right)^3\right], \tag{ε.23}$$

which at criticality has a solution

$$\frac{U^{(\ell)}}{n^2\left(K^{(\ell)}\right)^3} = 75\frac{\ell^2}{n^2} + \cdots. \tag{ε.24}$$

https://doi.org/10.1017/9781009023405.015 Published online by Cambridge University Press

will have its model complexity dominated by data-dependent couplings with $2k$-sample indices, requiring

$$O\left(N_{\mathcal{D}}^{2k}\right) \tag{ε.25}$$

numbers. In this way, the **$1/n$ expansion** gives a sequence of effective theories with increasing accuracy at the cost of increasing complexity.[7]

Importantly, for any particular network architecture that we want to describe, as the depth-to-width ratio $r \equiv L/n$ increases, we'll in principle need to include more and more of these higher-order terms, making our macroscopic effective theory more and more complex:

- In the strict limit $r \to 0$, the interactions between neurons turn off, and the sparse $O(N_{\mathcal{D}}^2)$ *Gaussian* description of the infinite-width limit, (ε.9), will be accurate. Such networks are not really deep, as $L/n = 0$, and they do not learn representations (§10.4.3).

- In the regime $0 < r \ll 1$, there are small nontrivial interactions between neurons, and the nearly-sparse $O(N_{\mathcal{D}}^4)$ *nearly-Gaussian* description of the finite-width effective theory truncated at order $1/n$, (ε.13), will be accurate. Such networks are wide while at the same time having nontrivial depth, $L/n \neq 0$, and they do learn representations (§11.4.3).

- For larger r, the neurons are strongly-coupled, and a more generic $O(N_{\mathcal{D}}^{2k})$ *non-Gaussian* description, (ε.22), would in principle be necessary. However, in this

[7] Note here that this counting is for the union of the training set and the test set, $\mathcal{D} = \mathcal{A} \cup \mathcal{B}$, rather than just for the training set $\mathcal{A}$; in other words, the stochastic predictions made on a test sample, $x_{\hat{\beta}}$, will necessarily depend on that test sample. In particular, every time you want to make a prediction on a new sample that you haven't predicted before, $N_{\mathcal{D}}$ increases in size, and so does your description of the joint distribution – (ε.9) and (ε.13) – over the entire dataset $\mathcal{D}$. Statistical models that have this property are sometimes called **nonparametric models**, since the "parameters" of the full distribution – i.e., the *data-dependent couplings* – depend on data points that you may not have even seen yet.

From the macroscopic perspective, the description of any *single* prediction scales with the size of the training set as $O\left((N_{\mathcal{A}}+1)^p\right) \approx O(N_{\mathcal{A}}^p)$; in principle you can just plug $x_{\hat{\beta}}$ into a prediction formula such as (∞.154). From the microscopic perspective, if we train a model and find a solution $\theta^\star \equiv \theta^\star(\mathcal{A})$ for the model parameters given a training set $\mathcal{A}$, then in practice we can then forget about that training set and simply make predictions as $z(x_{\hat{\beta}}; \theta^\star)$ – paying only the computation complexity cost of using the model to make a prediction. Thus, in deep learning we can think of this nonparametric growth of macroscopic model complexity with the size of the test set as similar in nature to the microscopic $O(P)$ complexity of the forward pass needed to make a new prediction.

Potentially it may have been useful to tell you – at the very least in order to understand this epilogue's epigraph – that a nonparametric model based on a Gaussian distribution is called a **Gaussian process**, and accordingly people sometimes say that neural networks in the infinite-width limit are Gaussian processes. Similarly, if you wanted to talk about the distribution over finite-width networks in the context of nonparametric statistics, you might call it a **nearly-Gaussian process**. These processes are distributions over functions $z(x)$, where, e.g., z is the trained model output function. However, the reason why we haven't brought this up before is that we find this terminology unnecessary: for any fixed dataset $\mathcal{D}$, we don't have to worry about distributions over *functions* and instead can just think about a joint distribution over the finite *sets* of outputs evaluated on the dataset.

https://doi.org/10.1017/9781009023405.015 Published online by Cambridge University Press

> case the the macroscopic perspective leads to an *ineffective* description that is not tractable, and relatedly, we do not expect such networks to be practically useful for machine learning tasks (§3.4).[8]

In this way, our effective theory cutoff scale r governs the model complexity of the statistics needed to faithfully describe the behavior of different neural networks. The simplicity of the *macroscopic perspective* only emerges for small values of the cutoff r.

With that in mind, the practical success of deep learning in the overparameterized regime and the empirical accuracy of a simple scaling hypothesis is really telling us that useful neural networks should be *sparse* – hence the preference for larger and larger models – but not too sparse – so that they are also *deep*. Thus, from the macroscopic perspective, a **nearly-sparse** model complexity is perhaps the most important inductive bias of deep learning.

♦♦♦♦

For an *information-theoretic* estimate of the depth-to-width ratio $r^\star$ for which the wide attraction of simplicity and the deep need of complexity are balanced to the end of *near-sparsity*, please feel free to flip the page and make your way through Appendix A.

[8] In §3.4, we were able to access this regime in the special case of *deep linear networks*. There, we saw that higher-point connected correlators can grow uncontrollably even when the network is tuned to criticality, and there's no reason to expect that this would be any more favorable for nonlinear activation functions. Moreover, as we discussed in §5.4, the growth of these higher-order correlators in networks for all choices of activation functions will lead to large fluctuations from instantiation to instantiation, meaning that the ensemble description (ε.22) can no longer be trusted for any *particular* network. Altogether, this suggests that networks of an aspect ratio r that require large sample-space complexity, $O\left(N_{\mathcal{D}}^{2k}\right)$ with large k, will generically exhibit strongly-coupled chaos; we do not expect such networks to be effectively describable or practically trainable.

Even if we could find a workable network with such a large complexity, would we ever need it? As a *gedanken model*, let's consider an ineffective description with almost no truncation, say $k \sim N_{\mathcal{D}}$, which comes with an exponential number of data-dependent couplings, $O\left(N_{\mathcal{D}}^{N_{\mathcal{D}}}\right)$. Such an exponentially complex description would only really be appropriate when we have *unstructured data*: e.g., if we have a binary labeling $f(x) = \{0, 1\}$ for which each label is chosen *randomly*, then the number of possible functions for $N_{\mathcal{D}}$ uncorrelated data points is $2^{N_{\mathcal{D}}}$; each time we want to incorporate a new input-output pair into our dataset, we'd have to *double* the complexity of our description. Such unstructured data is at odds with one of the main purposes of machine learning: recognizing patterns in the data – i.e., correlations – which allow for the learning of representations and the finding of sparse descriptions. In fact, as there's no efficient way to learn such unstructured datasets – see, e.g., the *no-free-lunch theorem* of [74, 75] – in practice we cannot possibly require these ineffective descriptions for any realistic machine learning scenarios.

https://doi.org/10.1017/9781009023405.015 Published online by Cambridge University Press