3

Effective Theory of Deep Linear Networks at Initialization

> *. . . a system which has spherical symmetry . . . certainly cannot result in an organism such as a horse, which is not spherically symmetrical.*
>
> Alan Turing, on the limitations of toy models [24].

In this final warm-up chapter, we introduce and then solve a toy model of deep learning, the **deep linear network**.[1] As will be explained in §3.1, the deep linear network is simply an MLP with `linear` activation functions. In particular, such a network can only compute linear transformations of its inputs and certainly cannot result in a function such as a human, which is empirically known to be nonlinear. Nonetheless, the study of deep linear networks will serve as a useful blueprint for an *effective theory of deep learning* that we will develop more generally over the subsequent chapters. Specifically, the exercises in this chapter illustrate how layer-to-layer recursions control the statistics of deep neural networks in a very intuitive way, without getting bogged down by all the technical details.

To that end, in §3.2 we obtain and then exactly solve a layer-to-layer recursion for the two-point correlator of preactivations in deep linear networks. The result highlights that the statistics of the network sensitively depend on the setting of the *initialization hyperparameters*, with the sensitivity increasing exponentially with depth. This leads to the important concept of *criticality*, which we will explore in §5 in greater depth and sensitivity. In short, we learn that for networks to be well behaved, these hyperparameters need to be finely tuned.

Next, in §3.3 we obtain and then solve a layer-to-layer recursion for the four-point correlator, albeit for a single input to further simplify the algebra. This showcases the way in which the behavior of the network can depend on the *architecture hyperparameters*, particularly the width and depth of the network. In addition, we interpret the four-point

[1] For physicists, we give an analogy: the deep linear network is to deep learning as the simple harmonic oscillator is to quantum mechanics.

https://doi.org/10.1017/9781009023405.005 Published online by Cambridge University Press

connected correlator as a measurement of the *fluctuation* of the network function from draw to draw of the model parameters. Such fluctuations can interfere with the tuning of the initialization hyperparameters and need to be controlled so that networks behave reliably for typical draws. The scale of the fluctuations is set by the depth-to-width ratio of the network, highlighting this important *emergent scale* in the analysis of MLPs, and we'll see that the fluctuations can be kept under control by keeping the depth-to-width ratio of the network sufficiently small.

Finally, in §3.4 we obtain a recursion for an arbitrary M-point correlator for a deep linear network evaluated on a single input. Such recursions are all exactly solvable at any width n and depth L, meaning we can fully determine the statistics of these networks at initialization.[2] Given these nonperturbative solutions, we take the limit of large width, with fixed depth, and the limit of large depth, with fixed width, and show explicitly that these two limits do not commute. We also construct an interpolating solution with both large width and large depth – but fixed depth-to-width ratio L/n – and see how this *scale* serves as a perturbative parameter that controls all the interactions in the network and controls the validity of the perturbative analysis.

3.1 Deep Linear Networks

A deep linear network iteratively transforms an input $x_{i;\alpha}$ through a sequence of simple linear transformations

$$z_{i;\alpha}^{(\ell+1)} = b_i^{(\ell+1)} + \sum_{j=1}^{n_\ell} W_{ij}^{(\ell+1)} z_{j;\alpha}^{(\ell)}, \tag{3.1}$$

with $z_{i;\alpha}^{(0)} \equiv x_{i;\alpha}$ and $z_{i;\alpha}^{(\ell)} \equiv z_i^{(\ell)}(x_\alpha)$. Since the `linear` activation function is the identity function, $\sigma(z) = z$, there's no distinction here between preactivations and activations.

In this chapter, we'll simplify matters a bit by turning off all the biases, $b_i^{(\ell)} = 0$, so that the preactivations in layer ℓ are simply given by a repeated matrix multiplication of weight matrices as

$$z_{i;\alpha}^{(\ell)} = \sum_{j_0=1}^{n_0} \sum_{j_1=1}^{n_1} \cdots \sum_{j_{\ell-1}=1}^{n_{\ell-1}} W_{ij_{\ell-1}}^{(\ell)} W_{j_{\ell-1}j_{\ell-2}}^{(\ell-1)} \cdots W_{j_1 j_0}^{(1)} x_{j_0;\alpha} \equiv \sum_{j=1}^{n_0} \mathcal{W}_{ij}^{(\ell)} x_{j;\alpha}. \tag{3.2}$$

Here we have introduced an n_ℓ-by-n_0 matrix

$$\mathcal{W}_{ij}^{(\ell)} = \sum_{j_1=1}^{n_1} \cdots \sum_{j_{\ell-1}=1}^{n_{\ell-1}} W_{ij_{\ell-1}}^{(\ell)} W_{j_{\ell-1}j_{\ell-2}}^{(\ell-1)} \cdots W_{j_1 j}^{(1)}, \tag{3.3}$$

[2] This notion of *solve* should not be confused with the solving of the training dynamics for a particular learning algorithm. In the context of deep linear networks, the dynamics of gradient descent were analyzed in [25]. In §10 and §∞, we will solve the training dynamics of gradient descent for MLPs with general activation functions in the context of our effective theory formalism.

https://doi.org/10.1017/9781009023405.005 Published online by Cambridge University Press

which highlights the fact that the preactivation at the ℓ-th layer is simply a linear transformation of the input. Additionally, let us set $C_W^{(\ell)} \equiv C_W$ so that the order-one part of the weight variance is layer independent. All together, this means that the initialization distribution over the weights is characterized by the following expectations:

$$\mathbb{E}\left[W_{ij}^{(\ell)}\right] = 0\,, \qquad \mathbb{E}\left[W_{i_1 j_1}^{(\ell)} W_{i_2 j_2}^{(\ell)}\right] = \delta_{i_1 i_2}\delta_{j_1 j_2}\frac{C_W}{n_{\ell-1}}. \tag{3.4}$$

Somewhat counterintuitively, deep linear networks generically represent a smaller set of functions than fully general linear transformations, a.k.a. one-layer networks of the same input-output dimensions.[3] As an extreme example, let's take a two-layer deep linear network in which the first hidden layer consists of a single neuron $n_1 = 1$ and consider the network output in the second layer $\ell = 2$. In this case, all the information in the input is compressed through a *bottleneck* into a single number in the first layer before being converted into an n_2-dimensional vector in the output layer. Surely, such a deep linear network represents a tinier subspace of linear transformations than the space given by all possible n_2-by-n_0 matrices, so long as $n_0, n_2 > 1$.

More importantly, we will show that the statistics of deep linear networks at initialization are also very different from those of one-layer networks. In particular, while the statistics of each $W_{ij}^{(\ell)}$ are given by a simple Gaussian distribution, the statistics of their product $\mathcal{W}_{ij}^{(\ell)}$ are non-Gaussian, depending in a complicated way on the depth ℓ and widths $n_1, \ldots, n_\ell$ of the network.

The goal of the rest of this chapter is to exactly work out this dependence. Concretely, we are going to compute the nontrivial distribution

$$p\Big(z^{(\ell)}\Big|\mathcal{D}\Big) \equiv p\Big(z^{(\ell)}\left(x_1\right), \ldots, z^{(\ell)}\left(x_{N_\mathcal{D}}\right)\Big) \tag{3.5}$$

of the preactivations $z_{i;\alpha}^{(\ell)} \equiv z_i^{(\ell)}(x_\alpha)$ implied by the iterated multiplication (3.2) when evaluated on the entire dataset $\mathcal{D}$. As mentioned in §1.2, a distribution is completely determined by the set of all its M-point correlators, and so our method for determining $p\Big(z^{(\ell)}\Big|\mathcal{D}\Big)$ will be to directly compute these correlators.

Before moving on to the next section, let's consider the simplest observable, the mean of the preactivation $z_{i;\alpha}^{(\ell)}$. Taking an expectation of the defining equation (3.2), it's easy to see that the mean preactivation must vanish at any layer:

[3] This is not necessarily a bad thing, since there are often both computational and representational advantages to focusing on a specialized class of functions. For instance, we saw that convolutional networks represent a much smaller set of functions than MLPs, and yet they are known to perform better on computer vision tasks due to their translational-invariance-respecting *inductive bias* as well as the fact that they require significantly less computation due to their sparse pattern of connections. Having said that, it's not obvious if deep linear networks have a useful inductive bias when compared to general linear transformations.

https://doi.org/10.1017/9781009023405.005 Published online by Cambridge University Press

$$\mathbb{E}\left[z_{i;\alpha}^{(\ell)}\right] = \sum_{j_0=1}^{n_0}\sum_{j_1=1}^{n_1}\cdots\sum_{j_{\ell-1}=1}^{n_{\ell-1}} \mathbb{E}\left[W_{ij_{\ell-1}}^{(\ell)} W_{j_{\ell-1}j_{\ell-2}}^{(\ell-1)}\cdots W_{j_1 j_0}^{(1)} x_{j_0;\alpha}\right] \tag{3.6}$$
$$= \sum_{j_0=1}^{n_0}\sum_{j_1=1}^{n_1}\cdots\sum_{j_{\ell-1}=1}^{n_{\ell-1}} \mathbb{E}\left[W_{ij_{\ell-1}}^{(\ell)}\right]\mathbb{E}\left[W_{j_{\ell-1}j_{\ell-2}}^{(\ell-1)}\right]\cdots\mathbb{E}\left[W_{j_1 j_0}^{(1)}\right] x_{j_0;\alpha} = 0\,,$$

since the weight matrices are mutually independent – and independent of the input – and have zero mean (3.4). By a similar argument, it's easy to see that any odd-point correlator of preactivations will vanish as well. Thus, going forward, we will only have to concern ourselves with the even-point correlators.

3.2 Criticality

Since the mean is trivial, the next simplest candidate for an interesting observable is the two-point correlator $\mathbb{E}\left[z_{i_1;\alpha_1}^{(\ell)} z_{i_2;\alpha_2}^{(\ell)}\right]$, which quantifies the typical magnitudes of the preactivations. We'll first go through the math, and then we'll discuss the physics.

Math: Recursion for the Two-Point Correlator

Let's start slowly by first considering the two-point correlator in the first layer. Using the defining equation (3.2) to express the first-layer preactivations in terms of the inputs as

$$z_{i;\alpha}^{(1)} = \sum_{j}^{n_0} W_{ij}^{(1)} x_{j;\alpha}, \tag{3.7}$$

we can express the two-point correlator as

$$\mathbb{E}\left[z_{i_1;\alpha_1}^{(1)} z_{i_2;\alpha_2}^{(1)}\right] = \sum_{j_1,j_2=1}^{n_0} \mathbb{E}\left[W_{i_1 j_1}^{(1)} x_{j_1;\alpha_1} W_{i_2 j_2}^{(1)} x_{j_2;\alpha_2}\right] \tag{3.8}$$
$$= \sum_{j_1,j_2=1}^{n_0} \mathbb{E}\left[W_{i_1 j_1}^{(1)} W_{i_2 j_2}^{(1)}\right] x_{j_1;\alpha_1} x_{j_2;\alpha_2}$$
$$= \sum_{j_1,j_2=1}^{n_0} \frac{C_W}{n_0}\delta_{i_1 i_2}\delta_{j_1 j_2} x_{j_1;\alpha_1} x_{j_2;\alpha_2} = \delta_{i_1 i_2} C_W \frac{1}{n_0}\sum_{j=1}^{n_0} x_{j;\alpha_1} x_{j;\alpha_2}\,,$$

where to go from the second line to the third line, we Wick-contracted the two weights and inserted the variance (3.4). Additionally, let us introduce the notation

$$G_{\alpha_1\alpha_2}^{(0)} \equiv \frac{1}{n_0}\sum_{i=1}^{n_0} x_{i;\alpha_1} x_{i;\alpha_2} \tag{3.9}$$

for the inner product of the two inputs, normalized by the input dimension n_0. In terms of this object, we can rewrite the first-layer two-point correlator (3.8) as

$$\mathbb{E}\left[z_{i_1;\alpha_1}^{(1)} z_{i_2;\alpha_2}^{(1)}\right] = \delta_{i_1 i_2} C_W G_{\alpha_1\alpha_2}^{(0)}. \tag{3.10}$$

https://doi.org/10.1017/9781009023405.005 Published online by Cambridge University Press

Next, we could mindlessly repeat the same exercise to get the two-point correlator in any arbitrary layer, using the defining equation (3.2) to express $z^{(\ell)}_{i;\alpha}$ in terms of the input. Instead, in order to practice our recursive approach, let's evaluate the two-point correlator recursively. To do so, we inductively assume that the two-point correlator at the ℓ-th layer is known and then derive the two-point correlator at the $(\ell+1)$-th layer. Using the iteration equation (3.1) with the bias set to zero, we find

$$\begin{aligned}\mathbb{E}\left[z^{(\ell+1)}_{i_1;\alpha_1} z^{(\ell+1)}_{i_2;\alpha_2}\right] &= \sum_{j_1,j_2=1}^{n_\ell} \mathbb{E}\left[W^{(\ell+1)}_{i_1 j_1} W^{(\ell+1)}_{i_2 j_2} z^{(\ell)}_{j_1;\alpha_1} z^{(\ell)}_{j_2;\alpha_2}\right] \\ &= \sum_{j_1,j_2=1}^{n_\ell} \mathbb{E}\left[W^{(\ell+1)}_{i_1 j_1} W^{(\ell+1)}_{i_2 j_2}\right] \mathbb{E}\left[z^{(\ell)}_{j_1;\alpha_1} z^{(\ell)}_{j_2;\alpha_2}\right] \\ &= \delta_{i_1 i_2} C_W \frac{1}{n_\ell} \sum_{j=1}^{n_\ell} \mathbb{E}\left[z^{(\ell)}_{j;\alpha_1} z^{(\ell)}_{j;\alpha_2}\right] ,\end{aligned} \tag{3.11}$$

where to go from the first line to the second line, we used the fact that the weights $W^{(\ell+1)}$ of the $(\ell+1)$-th layer are statistically independent from the preactivations $z^{(\ell)}$ in the ℓ-th layer, and to go from the second line to the third line, we Wick-contracted the two weights and substituted in the variance (3.4). Notice that at *any* layer, the two-point correlator is proportional to the Kronecker delta $\delta_{i_1 i_2}$, vanishing unless the neural indices i_1 and i_2 are the same. With that in mind, let us decompose the two-point correlator as

$$\mathbb{E}\left[z^{(\ell)}_{i_1;\alpha_1} z^{(\ell)}_{i_2;\alpha_2}\right] \equiv \delta_{i_1 i_2} G^{(\ell)}_{\alpha_1 \alpha_2} \tag{3.12}$$

and introduce a generalization of the above notation (3.9) for an arbitrary layer ℓ. Multiplying this equation by $\delta_{i_1 i_2}$, summing over $i_1, i_2 = 1, \ldots, n_\ell$, and dividing it by n_ℓ, the quantity $G^{(\ell)}_{\alpha_1 \alpha_2}$ can also be expressed as

$$G^{(\ell)}_{\alpha_1 \alpha_2} = \frac{1}{n_\ell} \sum_{j=1}^{n_\ell} \mathbb{E}\left[z^{(\ell)}_{j;\alpha_1} z^{(\ell)}_{j;\alpha_2}\right] \tag{3.13}$$

and can thus be thought of as the average inner-product of preactivations in the ℓ-th layer, divided by the number of neurons in the layer n_ℓ. This inner product depends on sample indices *only* and lets us interpret $G^{(\ell)}_{\alpha_1 \alpha_2} \equiv G^{(\ell)}(x_{\alpha_1}, x_{\alpha_2})$ as the covariance of the two inputs, x_{α_1} and x_{α_2}, after passing through an ℓ-layer deep linear network.

With all this notation introduced and fully interpreted, it's easy to see that the above recursion (3.11) can be compactly summarized by

$$G^{(\ell+1)}_{\alpha_1 \alpha_2} = C_W G^{(\ell)}_{\alpha_1 \alpha_2}, \tag{3.14}$$

which describes how the covariance $G^{(\ell)}_{\alpha_1 \alpha_2}$ evolves from layer to layer. Apparently, to transform the covariance from layer ℓ to layer $\ell+1$, we simply multiply by the constant C_W. The initial condition $G^{(0)}_{\alpha_1 \alpha_2}$ is given by the inner product of the two inputs (3.9), and the solution is an exponential

$$G^{(\ell)}_{\alpha_1 \alpha_2} = \left(C_W\right)^\ell G^{(0)}_{\alpha_1 \alpha_2}, \tag{3.15}$$

https://doi.org/10.1017/9781009023405.005 Published online by Cambridge University Press

as is typical for a repeated application of matrix multiplication. Note that the factor of the width n_ℓ in the variance of the weights (3.4) nicely dropped out, indicating that this was in fact the proper way to scale the variance.

Physics: Criticality

Already at this point our analysis illustrates an interesting and very general phenomenon. Considering the solution (3.15), generically one of two things happens. If $C_W > 1$, the covariance blows up exponentially, quickly being driven to a fixed point $G^\star_{\alpha_1\alpha_2} = \infty$ for all pairs of inputs and leading to a divergent network output. If $C_W < 1$, the covariance exponentially decays to a fixed point $G^\star_{\alpha_1\alpha_2} = 0$ for all pairs of inputs, quickly curtailing any data dependence in the network output. Any time an observable approaches a value exponentially quickly, we'll refer to the limiting value as a **trivial fixed point**. The value $G^\star_{\alpha_1\alpha_2} = \infty$ associated with $C_W > 1$ and the value $G^\star_{\alpha_1\alpha_2} = 0$ associated with $C_W < 1$ are prime examples of trivial fixed points.

Exploring this further, first note that the diagonal part of the covariance at the output layer L estimates the typical magnitude of the output for a given input $x_{i;\alpha}$:

$$G^{(L)}_{\alpha\alpha} = \mathbb{E}\left[\frac{1}{n_L}\sum_{j=1}^{n_L}\left(z^{(L)}_{j;\alpha}\right)^2\right]. \tag{3.16}$$

With this observable in mind, the aforementioned exponential behavior should immediately set off alarm bells, signaling either some sort of numerical instability ($C_W > 1$) or loss of information ($C_W < 1$). In addition, note that the target values for the different components of the network output are typically $O(1)$ numbers, neither exponentially large nor small. Such exponential behavior of the network should thus make it extremely difficult to learn to approximate the desired function. In this way, this *exploding and vanishing covariance problem* is a baby version of the infamous exploding and vanishing gradient problem – a well-known obstacle to gradient-based training of deep networks – which we shall make more precise in §9.

However, we were actually a little too quick in our analysis before: what happens if we tune the weight variance C_W so that it's precisely equal to 1? This is clearly a special point in the hyperparameter space of initialization, separating the exponentially growing solution from the exponentially decaying solution. Going back to the recursion (3.14), we see that if $C_W = 1$ then the covariance is fixed, $G^{(\ell)}_{\alpha_1\alpha_2} = G^{(0)}_{\alpha_1\alpha_2} \equiv G^\star_{\alpha_1\alpha_2}$, manifestly preserving the full covariance of the input data even after passing through many layers of the deep linear network. This is a bona fide **nontrivial fixed point**, as it doesn't exponentially trivialize the structure of input data. Thus, at least at this heuristic level of analysis, choosing $C_W = 1$ appears to be essential for preserving the structure of the input data in a numerically stable manner. More generally, flowing to a nontrivial fixed point seems to be a necessary condition for deep networks to do anything useful.

https://doi.org/10.1017/9781009023405.005 Published online by Cambridge University Press

When we fine-tune the initialization hyperparameters of a network so that the covariance avoids exponential behavior, we'll call them **critical initialization hyperparameters**.[4] For deep linear networks, the critical initialization hyperparameter $C_W = 1$ separates two regimes, one with an exponentially growing covariance for $C_W > 1$ and the other with an exponentially decaying covariance for $C_W < 1$. When the weight variance is tuned to criticality, $C_W = 1$, the network has a perfect self-similarity of the covariance, preserving it exactly through the evolution from layer to layer.

In §5, we will extend our analysis of **criticality** to MLPs that use any particular activation function. And, as shall be seen further on in §10 and §∞, tuning a network to criticality is *critical* for any *deep* network to be well behaved and perform useful tasks – at least without otherwise employing ad-hoc tricks to ensure that signals can propagate stably.

3.3 Fluctuations

Recall from §1 that if a distribution is Gaussian and has a zero mean, then the covariance completely specifies the distribution. If the preactivation distribution $p\left(z^{(\ell)}\middle|\mathcal{D}\right)$ were Gaussian, this would mean that the critical tuning of the one initialization hyperparameter $C_W = 1$ would be sufficient to ensure that any observable is well behaved. However, if the distribution $p\left(z^{(\ell)}\middle|\mathcal{D}\right)$ is not Gaussian, then it's not clear a priori whether observables depending on higher-point connected correlators will be well behaved with the same tuning. In principle, such observables could require other tunings of C_W that are incompatible with the critical setting $C_W = 1$ for the covariance $G^{(\ell)}_{\alpha_1\alpha_2}$. To settle this question, let's look at the next simplest observable, the connected four-point correlator. As before, we'll go through the math first and discuss the physics second.

In this section and the next, to simplify the algebra we'll focus on correlators of preactivations that are evaluated only on a single input $x_\alpha = x$. This is sufficient to qualitatively highlight the importance of the higher-point correlators while letting us avoid the interference of some annoying technical manipulations. Accordingly, in these sections we will drop the sample indices on preactivations and denote the covariance as

$$G^{(\ell)}_2 \equiv G^{(\ell)}_{\alpha\alpha} = G^{(\ell)}(x, x). \tag{3.17}$$

In the next chapter, we'll consider the fully general case.

[4] This word choice is motivated by the analogy to critical phenomena in statistical physics. For instance, consider the prototypical example: a magnet made of iron. At high temperature, the magnetic moments – or spins – of the iron atoms point in random directions, leading to a paramagnetic phase without any coherent magnetic field. By contrast, at low temperature, the spins instead try to collectively orient in the same direction, leading to a ferromagnetic phase with coherent magnetic field – think of the ∩-shaped cartoon magnet that children play with. A critical temperature separates these two phases of magnetism, and the magnet set to the critical temperature will exhibit very special behavior that is neither paramagnetism nor ferromagnetism but known as self-similarity.

https://doi.org/10.1017/9781009023405.005 Published online by Cambridge University Press

Math: Recursion for the Four-Point Correlator

As we did for the two-point correlator in the previous section, we'll begin by working out the four-point correlator in the first layer and then derive and solve a recursion for the correlator in the deeper layers. First for the first layer, using the defining equation (3.7) with the sample index omitted, we have for the *full* four-point correlator

$$\begin{aligned}
&\mathbb{E}\left[z_{i_1}^{(1)} z_{i_2}^{(1)} z_{i_3}^{(1)} z_{i_4}^{(1)}\right] \\
&= \sum_{j_1,j_2,j_3,j_4=1}^{n_0} \mathbb{E}\left[W_{i_1 j_1}^{(1)} W_{i_2 j_2}^{(1)} W_{i_3 j_3}^{(1)} W_{i_4 j_4}^{(1)}\right] x_{j_1} x_{j_2} x_{j_3} x_{j_4} \\
&= \frac{C_W^2}{n_0^2} \sum_{j_1,j_2,j_3,j_4=1}^{n_0} \left(\delta_{i_1 i_2}\delta_{j_1 j_2}\delta_{i_3 i_4}\delta_{j_3 j_4} + \delta_{i_1 i_3}\delta_{j_1 j_3}\delta_{i_2 i_4}\delta_{j_2 j_4} + \delta_{i_1 i_4}\delta_{j_1 j_4}\delta_{i_2 i_3}\delta_{j_2 j_3}\right) x_{j_1} x_{j_2} x_{j_3} x_{j_4} \\
&= C_W^2 \left(\delta_{i_1 i_2}\delta_{i_3 i_4} + \delta_{i_1 i_3}\delta_{i_2 i_4} + \delta_{i_1 i_4}\delta_{i_2 i_3}\right) \left(G_2^{(0)}\right)^2 ,
\end{aligned} \tag{3.18}$$

where to go from line two to line three, we made three distinct pairings for the two Wick contractions of the four weights and then used the weight variance (3.4) to evaluate each contraction. To get to the final line, we evaluated the sums over the j indices and then substituted using our definition of the inner product (3.9), which for a single input simply reads

$$G_2^{(0)} = \frac{1}{n_0} \sum_{j=1}^{n_0} x_j x_j . \tag{3.19}$$

Comparing this result (3.18) with the two-point correlator in the first layer (3.10), we note that this answer is precisely what we'd expect for the full four-point correlator if the preactivation distribution were exactly Gaussian. Thus, deep linear networks appear to be simply Gaussian after a single layer, at least at the four-point correlator level of analysis.[5]

This Gaussianity does *not* hold in deeper layers. To see that, let's derive and solve a recursion for the four-point correlator. Beginning with the iteration equation (3.1) with zero bias, we find

$$\begin{aligned}
&\mathbb{E}\left[z_{i_1}^{(\ell+1)} z_{i_2}^{(\ell+1)} z_{i_3}^{(\ell+1)} z_{i_4}^{(\ell+1)}\right] \\
&= \sum_{j_1,j_2,j_3,j_4=1}^{n_\ell} \mathbb{E}\left[W_{i_1 j_1}^{(\ell+1)} W_{i_2 j_2}^{(\ell+1)} W_{i_3 j_3}^{(\ell+1)} W_{i_4 j_4}^{(\ell+1)}\right] \mathbb{E}\left[z_{j_1}^{(\ell)} z_{j_2}^{(\ell)} z_{j_3}^{(\ell)} z_{j_4}^{(\ell)}\right]
\end{aligned} \tag{3.20}$$

[5] In the next chapter, we'll show very generally that the preactivation distribution is always Gaussian in the first layer.

https://doi.org/10.1017/9781009023405.005 Published online by Cambridge University Press

$$
\begin{aligned}
&= \frac{C_W^2}{n_\ell^2} \sum_{j_1,j_2,j_3,j_4=1}^{n_\ell} \left(\delta_{i_1 i_2}\delta_{j_1 j_2}\delta_{i_3 i_4}\delta_{j_3 j_4} + \delta_{i_1 i_3}\delta_{j_1 j_3}\delta_{i_2 i_4}\delta_{j_2 j_4} + \delta_{i_1 i_4}\delta_{j_1 j_4}\delta_{i_2 i_3}\delta_{j_2 j_3}\right) \\
&\quad \times \mathbb{E}\left[z_{j_1}^{(\ell)} z_{j_2}^{(\ell)} z_{j_3}^{(\ell)} z_{j_4}^{(\ell)}\right] \\
&= C_W^2 \left(\delta_{i_1 i_2}\delta_{i_3 i_4} + \delta_{i_1 i_3}\delta_{i_2 i_4} + \delta_{i_1 i_4}\delta_{i_2 i_3}\right) \frac{1}{n_\ell^2} \sum_{j,k=1}^{n_\ell} \mathbb{E}\left[z_j^{(\ell)} z_j^{(\ell)} z_k^{(\ell)} z_k^{(\ell)}\right] ,
\end{aligned}
$$

where on the second line we used the independence of the $(\ell+1)$-th-layer weights from the ℓ-th-layer preactivations, on the third line we again made three distinct pairings for the two pairs of Wick contractions of the four weights, and on the last line we made judicious use of Kronecker deltas to collapse the sums.

Now, we see from this recursion that at *any* layer the full four-point correlator is proportional to the factor $(\delta_{i_1 i_2}\delta_{i_3 i_4} + \delta_{i_1 i_3}\delta_{i_2 i_4} + \delta_{i_1 i_4}\delta_{i_2 i_3})$, a fixed tensor structure that specifies the neural-index dependence of the correlator. Thus by decomposing the full four-point correlator as

$$
\mathbb{E}\left[z_{i_1}^{(\ell)} z_{i_2}^{(\ell)} z_{i_3}^{(\ell)} z_{i_4}^{(\ell)}\right] \equiv \left(\delta_{i_1 i_2}\delta_{i_3 i_4} + \delta_{i_1 i_3}\delta_{i_2 i_4} + \delta_{i_1 i_4}\delta_{i_2 i_3}\right) G_4^{(\ell)}, \tag{3.21}
$$

we can put all of the layer dependence into this simpler object $G_4^{(\ell)}$ and not worry about neural indices in our recursion. In terms of this decomposition, the result (3.18) for the correlator in the first layer becomes

$$
G_4^{(1)} = C_W^2 \left(G_2^{(0)}\right)^2, \tag{3.22}
$$

and the final factor in the above recursion (3.20) can be rewritten as

$$
\frac{1}{n_\ell^2} \sum_{j,k=1}^{n_\ell} \mathbb{E}\left[z_j^{(\ell)} z_j^{(\ell)} z_k^{(\ell)} z_k^{(\ell)}\right] = \frac{1}{n_\ell^2} \sum_{j,k=1}^{n_\ell} \left(\delta_{jj}\delta_{kk} + \delta_{jk}\delta_{jk} + \delta_{jk}\delta_{kj}\right) G_4^{(\ell)} = \left(1 + \frac{2}{n_\ell}\right) G_4^{(\ell)}. \tag{3.23}
$$

Using this, the entire recursion above (3.20) can be rewritten simply as a recursion for $G_4^{(\ell)}$ as

$$
G_4^{(\ell+1)} = C_W^2 \left(1 + \frac{2}{n_\ell}\right) G_4^{(\ell)}. \tag{3.24}
$$

This recursion, with the initial condition set by (3.22), has a simple solution:

$$
\begin{aligned}
G_4^{(\ell)} &= C_W^{2\ell} \left[\prod_{\ell'=1}^{\ell-1} \left(1 + \frac{2}{n_{\ell'}}\right)\right] \left(G_2^{(0)}\right)^2 \\
&= \left[\prod_{\ell'=1}^{\ell-1} \left(1 + \frac{2}{n_{\ell'}}\right)\right] \left(G_2^{(\ell)}\right)^2 ,
\end{aligned} \tag{3.25}
$$

where in the final line we substituted in the solution (3.15) for the covariance. Now let's extract some physics from this compact formula.

https://doi.org/10.1017/9781009023405.005 Published online by Cambridge University Press

Physics: Large-n Expansion, Non-Gaussianities, Interactions, and Fluctuations

To start, we note that the four-point correlator (3.25) drastically simplifies in the limit of an infinite number of neurons per hidden layer ($n_\ell \to \infty$). In such a limit, the solution (3.25) degenerates to

$$G_4^{(\ell)} = \left(G_2^{(\ell)}\right)^2 , \tag{3.26}$$

and the full four-point correlator (3.21) becomes

$$\mathbb{E}\left[z_{i_1}^{(\ell)} z_{i_2}^{(\ell)} z_{i_3}^{(\ell)} z_{i_4}^{(\ell)}\right] = \left(\delta_{i_1 i_2}\delta_{i_3 i_4} + \delta_{i_1 i_3}\delta_{i_2 i_4} + \delta_{i_1 i_4}\delta_{i_2 i_3}\right)\left(G_2^{(\ell)}\right)^2 . \tag{3.27}$$

This is exactly what we'd find if the preactivation distribution were Gaussian: the four-point correlator is determined entirely by the two-point correlator, with the tensor structure determined by Wick's theorem. In fact, as we will show in the next chapter, for any MLP with any particular choice of a nonlinear activation function, the preactivation distribution is governed by Gaussian statistics in this infinite-width limit, implying no interactions between the neurons in such a limit. However, despite the rather large computational resources that *big tech* can throw at machine-learning problems, realistic MLPs simply do not have an infinite number of neurons per layer. To understand such realistic MLPs, we'll have to back off this infinite-width limit.

To illustrate this most clearly, let's set all the hidden layer widths to be equal $n_1 = n_2 = \cdots = n_{L-1} \equiv n$. Then, evaluating (3.25), the deviation from the infinite-width limit at the level of four-point correlator statistics is encoded by the difference

$$\begin{aligned} G_4^{(\ell)} - \left(G_2^{(\ell)}\right)^2 &= \left[\left(1+\frac{2}{n}\right)^{\ell-1} - 1\right]\left(G_2^{(\ell)}\right)^2 \\ &= \frac{2(\ell-1)}{n}\left(G_2^{(\ell)}\right)^2 + O\left(\frac{1}{n^2}\right) , \end{aligned} \tag{3.28}$$

where in the last line we expanded in $1/n$ and kept the leading correction to the infinite-width limit.[6] In particular, at criticality where $G_2^{(\ell)}$ is constant, this leading correction (3.28) scales inversely proportionally with the width and proportionally with the depth. Thus, the deviation from infinite width is proportional to the depth-to-width ratio of the network, our first encounter with this important **emergent scale**. There are multiple ways to think about this finite-width correction.

First, the connected four-point correlator (1.54) is given by

$$\mathbb{E}\left[z_{i_1}^{(\ell)} z_{i_2}^{(\ell)} z_{i_3}^{(\ell)} z_{i_4}^{(\ell)}\right]\Big|_{\text{connected}} = \left(\delta_{i_1 i_2}\delta_{i_3 i_4} + \delta_{i_1 i_3}\delta_{i_2 i_4} + \delta_{i_1 i_4}\delta_{i_2 i_3}\right)\left[G_4^{(\ell)} - \left(G_2^{(\ell)}\right)^2\right], \tag{3.29}$$

[6] This approximation is valid so long as the depth of the network doesn't grow too large. Stay tuned for the analysis in the next section where we will discuss how this limit breaks down.

https://doi.org/10.1017/9781009023405.005 Published online by Cambridge University Press

which directly connects the difference (3.28) to our measure of non-Gaussianity for the distribution. We see that the non-Gaussianity grows as the network deepens, and the preactivation statistics in layer ℓ are *nearly-Gaussian* so long as the emergent scale, the depth-to-width-ratio, remains perturbatively small. From the action perspective, this means that the quartic coupling changes – or **runs** – as the layer at which we consider the preactivation distribution changes, with the coupling growing in proportion with layer ℓ.

Second, in §1.3 we gave another interpretation for a nonzero connected four-point correlator as measuring interactions – i.e., the breakdown of statistical independence – between the different components of the random vector. To be very specific, let us look at a particular entry of the connected four-point correlator tensor with $i_1 = i_2 = j$ and $i_3 = i_4 = k$ for $j \neq k$. This entry can be expressed as

$$\mathbb{E}\left[\left(z_j^{(\ell)} z_j^{(\ell)} - G_2^{(\ell)}\right)\left(z_k^{(\ell)} z_k^{(\ell)} - G_2^{(\ell)}\right)\right] = G_4^{(\ell)} - \left(G_2^{(\ell)}\right)^2 , \quad \text{for } j \neq k. \tag{3.30}$$

This shows that the deviation of $z_j^{(\ell)} z_j^{(\ell)}$ from its mean value $\mathbb{E}\left[z_j^{(\ell)} z_j^{(\ell)}\right] = G_2^{(\ell)}$ on a particular neuron j is correlated with the same deviation from the mean on a different neuron k. We can thus interpret the finite-width difference (3.28) as controlling intralayer interactions between distinct neurons, with the strength of the interactions growing with depth.

Third, we can see that some observables that are deterministic in the infinite-width limit start to fluctuate at finite width. To this end, let us consider the simple observable

$$\mathcal{O}^{(\ell)} \equiv \mathcal{O}\left(z^{(\ell)}\right) \equiv \frac{1}{n}\sum_{j=1}^{n} z_j^{(\ell)} z_j^{(\ell)} , \quad \text{for } \ell < L, \tag{3.31}$$

which captures the average magnitude of the preactivations over all the different neurons in a hidden layer ℓ for a given instantiation of the network weights. Its mean over different realizations of the weights is given by the expectation

$$\mathbb{E}\left[\mathcal{O}^{(\ell)}\right] = \frac{1}{n}\sum_{j=1}^{n} \mathbb{E}\left[z_j^{(\ell)} z_j^{(\ell)}\right] = G_2^{(\ell)}, \tag{3.32}$$

and the magnitude of this observable's fluctuation from instantiation to instantiation is measured by its variance

$$\begin{aligned}
\mathbb{E}\left[\left(\mathcal{O}^{(\ell)} - \mathbb{E}\left[\mathcal{O}^{(\ell)}\right]\right)^2\right] &= \frac{1}{n^2}\sum_{j,k=1}^{n} \mathbb{E}\left[z_j^{(\ell)} z_j^{(\ell)} z_k^{(\ell)} z_k^{(\ell)}\right] - \left(G_2^{(\ell)}\right)^2 \\
&= \frac{1}{n^2}\sum_{j,k=1}^{n} \left(\delta_{jj}\delta_{kk} + \delta_{jk}\delta_{jk} + \delta_{jk}\delta_{kj}\right) G_4^{(\ell)} - \left(G_2^{(\ell)}\right)^2 \\
&= \frac{2}{n} G_4^{(\ell)} + \left[G_4^{(\ell)} - \left(G_2^{(\ell)}\right)^2\right] \\
&= \frac{2\ell}{n}\left(G_2^{(\ell)}\right)^2 + O\left(\frac{1}{n^2}\right) ,
\end{aligned} \tag{3.33}$$

https://doi.org/10.1017/9781009023405.005 Published online by Cambridge University Press

where in the last step we recalled the expansion (3.28) for the finite-width difference. As promised, $\mathcal{O}^{(\ell)}$ is deterministic at infinite width, since this variance is suppressed by $1/n$ and vanishes identically in the infinite-width limit. However, as we back off the infinite-width limit, the variance grows linearly with depth at criticality due to the finite-width correction (3.28). As such depth increases, the fluctuation becomes larger, meaning that the *typical* magnitude of the preactivations $\mathcal{O}^{(\ell)}$ measured on any given realization of the deep linear network may deviate more from the mean value $\mathbb{E}\left[\mathcal{O}^{(\ell)}\right] = G_2^{(\ell)}$.

All these finite-width effects – be they non-Gaussianities, intralayer interactions, or finite-width fluctuations – are proportional to the depth-to-width ratio of the network. This is perhaps the most important recurring theme of the book: the leading finite-width contributions at criticality grow linearly with depth, despite being suppressed by the inverse of the layer widths. Since the depths of real networks with at least one hidden layer are bounded from below as $L \geq 2$ – that is, at minimum such networks have one hidden layer and one output layer – in practice, networks of any finite size will express some amount of finite-width effects in their output distribution proportional to their aspect ratio L/n. As we will see later in §5, this emergent scaling will hold very generally at criticality for networks with any particular activation function.

Thus, the deeper a network is, the less the infinite-width Gaussian description will apply, due to accumulation of finite-width fluctuations. This is actually a good thing because, as we shall emphasize more in §11, infinite-width networks do not have correlations among neurons within a layer and cannot learn nontrivial representations from input data. Real, useful deep learning systems that are used in practice do both of these things, and our later analysis will show that deeper networks have the capacity to do more of these things.

Depth, however, is a double-edged sword. As the overall depth L of a network becomes comparable to its hidden-layer width, fluctuations can begin to dominate. In particular, such extremely deep networks will have a huge variation in observables from instantiation to instantiation. Thus, even if we choose the critical initialization hyperparameter $C_W = 1$, in some instantiations signals blow up, in other instantiations signals decay, and rarely do they stay tamed to be of order one. From a practical point of view, these networks are pretty useless.

This set of circumstances is actually very fortuitous from a theorist's vantage point: our *effective theory of deep learning* is most accurate when the aspect ratio L/n of the network is small but nonzero – due to the applicability of the perturbative large-width expansion – and this is exactly the setting of these architecture hyperparameters where networks work best in practice. In fact, one could expect that balancing the utility of nonzero depth for learning features against the cost of growing fluctuations could result in some optimal aspect ratio L/n for MLPs of a particular activation, just as we saw that there is a correct tuning for the initialization hyperparameter C_W for deep linear networks. We will return to this question of tuning L/n when we discuss inductive bias in §6 after first redoing our analysis of criticality and fluctuations for arbitrary activation functions in the following two chapters, §4 and §5. In particular, in these chapters we will understand how the statistics of the preactivations run with depth and see the emergence

https://doi.org/10.1017/9781009023405.005 Published online by Cambridge University Press

of the depth-to-width ratio as a scale that controls the validity of the perturbative $1/n$ expansion, as was the case here for deep linear networks.

Quite generally, in the regime where perturbation theory works, the finite-width corrections grow linearly – *not* exponentially – with depth, and the network remains well behaved. By contrast, when the depth-to-width ratio becomes large, perturbation theory breaks down, making it very difficult to analyze such networks. However, in the special case of deep linear networks, a nonperturbative analysis is possible. In the next section we'll illustrate explicitly what happens to deep linear networks when the depth-to-width ratio grows very large in order to paint an intuitive picture of the way networks behave in this regime.

3.4 Chaos

In the last two sections we used the method of Wick contractions to derive recursions for the two-point and four-point correlators of deep linear networks, which we then easily solved. Now, we will use this same method to compute all the higher-point correlators in order to complete our goal of determining the full distribution $p\left(z^{(\ell)}\middle|\mathcal{D}\right)$. So that we may first simplify the algebra and then focus on the interesting properties of this distribution, we'll again evaluate the correlators only on a single input $x_\alpha = x$ and drop the sample indices in all of the following equations. Math then physics.

Math: Recursions for Six-Point and Higher-Point Correlators

Starting with the first layer, let's compute a general $2m$-point *full* correlator. As this involves many Wick contractions, it might be helpful to remind yourself of the formal statement of Wick's theorem by flipping back to §1.1 and consulting (1.45)... Good.

Now, using the defining equation (3.7) to express the first-layer preactivations in terms of the input, we get

$$\begin{aligned}
&\mathbb{E}\left[z_{i_1}^{(1)} z_{i_2}^{(1)} \cdots z_{i_{2m-1}}^{(1)} z_{i_{2m}}^{(1)}\right] \\
&\quad = \sum_{j_1,\ldots,j_{2m}=1}^{n_0} \mathbb{E}\left[W_{i_1 j_1}^{(1)} W_{i_2 j_2}^{(1)} \cdots W_{i_{2m-1} j_{2m-1}}^{(1)} W_{i_{2m} j_{2m}}^{(1)}\right] x_{j_1} x_{j_2} \cdots x_{j_{2m-1}} x_{j_{2m}} \\
&\quad = \left(\sum_{\text{all parings}} \delta_{i_{k_1} i_{k_2}} \cdots \delta_{i_{k_{2m-1}} i_{k_{2m}}}\right) C_W^m \left(G_2^{(0)}\right)^m \\
&\quad = \left(\sum_{\text{all parings}} \delta_{i_{k_1} i_{k_2}} \cdots \delta_{i_{k_{2m-1}} i_{k_{2m}}}\right) \left(G_2^{(1)}\right)^m ,
\end{aligned} \tag{3.34}$$

where, as before we used Wick's theorem to determine the Wick contractions and then evaluated each contraction by substituting in (3.4) for the variance. Here, the sum is over all the possible pairing of the $2m$ auxiliary indices, $k_1, \ldots, k_{2m}$, resulting in $(2m-1)!!$ distinct terms, and on the final line we substituted in the solution (3.15) for the first-layer covariance.

https://doi.org/10.1017/9781009023405.005 Published online by Cambridge University Press

The result (3.34) confirms what we suspected in the last section, that the preactivation distribution for the first layer is completely Gaussian. If this isn't clear by inspection, it's easy to check directly – via basically the same application of Wick's theorem – that the correlators (3.34) are precisely the $2m$-point correlators of a Gaussian distribution with zero mean and variance $\delta_{i_1 i_2} G_2^{(1)}$. In other words, the preactivation distribution in the first layer is governed by the quadratic action

$$S\left(z^{(1)}\right) = \frac{1}{2G_2^{(1)}} \sum_{i=1}^{n_1} z_i^{(1)} z_i^{(1)}. \tag{3.35}$$

Before presenting a recursion for general $2m$-point correlators, let us work out the recursion for the six-point correlator in detail. Beginning with the iteration equation (3.1) with the bias set to zero, we find

$$\begin{aligned}
&\mathbb{E}\left[z_{i_1}^{(\ell+1)} z_{i_2}^{(\ell+1)} z_{i_3}^{(\ell+1)} z_{i_4}^{(\ell+1)} z_{i_5}^{(\ell+1)} z_{i_6}^{(\ell+1)}\right] \qquad (3.36)\\
&= \sum_{j_1,j_2,j_3,j_4,j_5,j_6=1}^{n_\ell} \mathbb{E}\left[W_{i_1 j_1}^{(\ell+1)} W_{i_2 j_2}^{(\ell+1)} W_{i_3 j_3}^{(\ell+1)} W_{i_4 j_4}^{(\ell+1)} W_{i_5 j_5}^{(\ell+1)} W_{i_6 j_6}^{(\ell+1)}\right] \mathbb{E}\left[z_{j_1}^{(\ell)} z_{j_2}^{(\ell)} z_{j_3}^{(\ell)} z_{j_4}^{(\ell)} z_{j_5}^{(\ell)} z_{j_6}^{(\ell)}\right]\\
&= C_W^3 \Big(\delta_{i_1 i_2}\delta_{i_3 i_4}\delta_{i_5 i_6} + \delta_{i_1 i_3}\delta_{i_2 i_4}\delta_{i_5 i_6} + \delta_{i_1 i_4}\delta_{i_2 i_3}\delta_{i_5 i_6}\\
&\quad + \delta_{i_1 i_2}\delta_{i_3 i_5}\delta_{i_4 i_6} + \delta_{i_1 i_3}\delta_{i_2 i_5}\delta_{i_4 i_6} + \delta_{i_1 i_5}\delta_{i_2 i_3}\delta_{i_4 i_6}\\
&\quad + \delta_{i_1 i_2}\delta_{i_5 i_4}\delta_{i_3 i_6} + \delta_{i_1 i_5}\delta_{i_2 i_4}\delta_{i_3 i_6} + \delta_{i_1 i_4}\delta_{i_2 i_5}\delta_{i_3 i_6}\\
&\quad + \delta_{i_1 i_5}\delta_{i_3 i_4}\delta_{i_2 i_6} + \delta_{i_1 i_3}\delta_{i_5 i_4}\delta_{i_2 i_6} + \delta_{i_1 i_4}\delta_{i_5 i_3}\delta_{i_2 i_6}\\
&\quad + \delta_{i_5 i_2}\delta_{i_3 i_4}\delta_{i_1 i_6} + \delta_{i_5 i_3}\delta_{i_2 i_4}\delta_{i_1 i_6} + \delta_{i_5 i_4}\delta_{i_2 i_3}\delta_{i_1 i_6}\Big) \frac{1}{n_\ell^3} \sum_{i,j,k=1}^{n_\ell} \mathbb{E}\left[z_i^{(\ell)} z_i^{(\ell)} z_j^{(\ell)} z_j^{(\ell)} z_k^{(\ell)} z_k^{(\ell)}\right],
\end{aligned}$$

noting again the independence of the $(\ell+1)$-th-layer weights from the ℓ-th-layer preactivations. On the final line, we see that there were fifteen distinct ways to make the three Wick contractions of six weights.

As we saw for the four-point correlator, the structure of neural indices for the full six-point correlator is the same for *any* layer and proportional to a constant tensor, given by the object in the parenthesis above with all those Kronecker deltas. This suggests a decomposition of the six-point correlator as

$$\mathbb{E}\left[z_{i_1}^{(\ell)} z_{i_2}^{(\ell)} z_{i_3}^{(\ell)} z_{i_4}^{(\ell)} z_{i_5}^{(\ell)} z_{i_6}^{(\ell)}\right] \equiv \left(\delta_{i_1 i_2}\delta_{i_3 i_4}\delta_{i_5 i_6} + \cdots + \delta_{i_5 i_4}\delta_{i_2 i_3}\delta_{i_1 i_6}\right) G_6^{(\ell)}, \tag{3.37}$$

with the neural-dependence encapsulated by that complicated sum-over-products of Kronecker deltas and the layer dependence captured solely by $G_6^{(\ell)}$.

Now, to find a recursion for $G_6^{(\ell)}$, we need to perform the sum

$$\frac{1}{n_\ell^3} \sum_{i,j,k=1}^{n_\ell} \mathbb{E}\left[z_i^{(\ell)} z_i^{(\ell)} z_j^{(\ell)} z_j^{(\ell)} z_k^{(\ell)} z_k^{(\ell)}\right] \tag{3.38}$$

https://doi.org/10.1017/9781009023405.005 Published online by Cambridge University Press

after substituting in the decomposition (3.37). With the given pattern of neural indices, there are really only three types of terms in the sum. In particular, there is one term that looks like this

$$\frac{1}{n_\ell^3}\sum_{i,j,k=1}^{n_\ell}\delta_{ii}\delta_{jj}\delta_{kk}=1, \tag{3.39}$$

six terms that look like this

$$\frac{1}{n_\ell^3}\sum_{i,j,k=1}^{n_\ell}\delta_{ij}\delta_{ji}\delta_{kk}=\frac{1}{n_\ell}, \tag{3.40}$$

and eight terms that look like this

$$\frac{1}{n_\ell^3}\sum_{i,j,k=1}^{n_\ell}\delta_{ij}\delta_{jk}\delta_{ki}=\frac{1}{n_\ell^2}. \tag{3.41}$$

Putting all these terms together, we find a recursion for the layer-dependence of the full six-point correlator

$$G_6^{(\ell+1)}=C_W^3\left(1+\frac{6}{n_\ell}+\frac{8}{n_\ell^2}\right)G_6^{(\ell)}, \tag{3.42}$$

which has a simple solution

$$\begin{aligned}G_6^{(\ell)}&=C_W^{3\ell}\left[\prod_{\ell'=1}^{\ell-1}\left(1+\frac{6}{n_{\ell'}}+\frac{8}{n_{\ell'}^2}\right)\right]\left(G_2^{(0)}\right)^3\\&=\left[\prod_{\ell'=1}^{\ell-1}\left(1+\frac{6}{n_{\ell'}}+\frac{8}{n_{\ell'}^2}\right)\right]\left(G_2^{(\ell)}\right)^3.\end{aligned} \tag{3.43}$$

Here, we used the initial condition (3.34) $G_6^{(0)}=\left(G_2^{(0)}\right)^3$, and on the final line we substituted in our solution for the variance of a single input (3.15).

Similarly, we can decompose an arbitrary $2m$-point correlator as

$$\mathbb{E}\left[z_{i_1}^{(\ell)}z_{i_2}^{(\ell)}\cdots z_{i_{2m-1}}^{(\ell)}z_{i_{2m}}^{(\ell)}\right]=\left(\sum_{\text{all parings}}\delta_{i_{k_1}i_{k_2}}\cdots\delta_{i_{k_{2m-1}}i_{k_{2m}}}\right)G_{2m}^{(\ell)} \tag{3.44}$$

and use a similar set of manipulations to show that the layer dependence $G_{2m}^{(\ell)}$ obeys a recursion

$$G_{2m}^{(\ell+1)}=c_{2m}(n_\ell)\,C_W^m G_{2m}^{(\ell)}, \tag{3.45}$$

with the combinatorial factor $c_{2m}(n)$ given by

$$c_{2m}(n)=\left(1+\frac{2}{n}\right)\left(1+\frac{4}{n}\right)\cdots\left(1+\frac{2m-2}{n}\right)=\frac{\left(\frac{n}{2}-1+m\right)!}{\left(\frac{n}{2}-1\right)!}\left(\frac{2}{n}\right)^m. \tag{3.46}$$

https://doi.org/10.1017/9781009023405.005 Published online by Cambridge University Press

We included the explicit form of this factor only for completeness. If you insist on checking this factor, note that it reproduces the right combinatorial factors for $2m = 2, 4, 6$, though we strongly suggest that you do not explicitly write out all of the terms for any other particular value of m. Overall, this recursion is still just a simple sequence of multiplications, with a simple solution

$$G_{2m}^{(\ell)} = \left[\prod_{\ell'=1}^{\ell-1} c_{2m}(n'_\ell)\right] \left(G_2^{(\ell)}\right)^m \ . \tag{3.47}$$

Enough with the math; time for the physics.[7]

Physics: Breakdown of Perturbation Theory and the Emergence of Chaos

Let's play with this formula (3.47) a bit by taking various limits. For simplicity, let's set all the hidden layer widths to be equal, $n_1 = n_2 = \cdots = n_{L-1} \equiv n$, and also focus only on output distribution $p\left(z^{(L)}\middle|x\right)$.

- On the one hand, if we send the network width to infinity, $n \to \infty$, while keeping the depth L fixed, then all the combinatorial factors (3.46) become unity:

  $$\lim_{n\to\infty} c_{2m}(n) = 1. \tag{3.48}$$

  In this infinite-width limit, all the correlators (3.47) are given by their Gaussian values

  $$G_{2m}^{(L)} = \left(G_2^{(L)}\right)^m, \tag{3.49}$$

  and the output distribution $p\left(z^{(L)}\middle|x\right)$ is precisely Gaussian. More generally, even for multiple inputs the output distribution $p\left(z^{(L)}\middle|\mathcal{D}\right)$ remains Gaussian, with covariance $G^{(L)}_{\alpha_1\alpha_2} = C_W^L\left(\frac{1}{n_0}\sum_{i=1}^{n_0} x_{i;\alpha_1}x_{i;\alpha_2}\right)$. As this distribution is equivalent to that of one-layer networks initialized with weight variance C_W^L, we see that such networks are not really deep after all.

- On the other hand, if we send the depth to infinity, $L \to \infty$, while keeping the width n fixed, then all the combinatorial factors are fixed and greater than one, $c_{2m} > 1$. This means that the higher-point correlators for $2m > 2$ will all blow up exponentially as

  $$G_{2m}^{(L)} = \left[c_{2m}(n)\right]^{L-1} \left(G_2^{(L)}\right)^m. \tag{3.50}$$

[7] If you *do* want more math, check out [26] for an alternative derivation of these $2m$-point correlators and a nonperturbative expression for the distribution $p\left(z^{(\ell)}\middle|x\right)$.

https://doi.org/10.1017/9781009023405.005 Published online by Cambridge University Press

Note that this behavior persists *even if* we tune the network to criticality by setting $C_W = 1$ so that the two-point correlator is fixed $G_2^{(\ell)} = G_2^{(0)}$. This shows explicitly how our large-width analysis from the last section can break down if the network depth becomes too large. Furthermore, the distribution implied by these correlators is extremely non-Gaussian, to say the least, and in practice the outputs of these networks will fluctuate chaotically from instantiation to instantiation. Such networks are entirely unusable.

- Clearly these limits do not commute, i.e.,

$$\lim_{n\to\infty}\lim_{L\to\infty} G_{2m}^{(L)} \neq \lim_{L\to\infty}\lim_{n\to\infty} G_{2m}^{(L)}. \tag{3.51}$$

However, we can construct an *interpolating solution* by sending both the width and depth to infinity, $n, L \to \infty$, while keeping their ratio fixed:

$$r \equiv \frac{L}{n}. \tag{3.52}$$

Noting that we can expand the combinatorial factors as

$$c_{2m}(n) = 1 + \frac{1}{n}\left(\sum_{s=1}^{m-1} 2s\right) + O\left(\frac{1}{n^2}\right) = 1 + \frac{m(m-1)}{n} + O\left(\frac{1}{n^2}\right) \tag{3.53}$$

and then using the well-known formula for the exponential,

$$\lim_{L\to\infty}\left[1 + \frac{a}{L} + O\left(\frac{1}{L^2}\right)\right]^L = e^a, \tag{3.54}$$

we can construct a limiting value for any correlator at a given value of m and fixed aspect ratio r:

$$G_{2m}^{(L)} \to e^{m(m-1)r}\left(G_2^{(L)}\right)^m. \tag{3.55}$$

This solution interpolates between the two extreme limits: by sending $r \to 0$ we recover the Gaussian limit (3.49), and by sending $r \to \infty$ we recover the chaotic limit (3.50) that demonstrates the breakdown of criticality.[8]

[8] This *double-scaling limit* corresponds to neglecting terms that scale like $\frac{L}{n^2}$, $\frac{L^3}{n^5}$, $\frac{L^{120}}{n^{157}}$, etc., which are all subleading when the depth and the width are large, $n, L \to \infty$, but their ratio r is fixed.

Furthermore, there is a very subtle point in using this interpolating solution – albeit a theoretical subtlety – when we consider not just a particular correlator at a given $2m$ but the set of *all* the correlators. Namely, for any *finite* n, L – no matter how big – there always exist higher-point correlators for which the exponential approximation (3.55) is invalid because the factor of $m(m-1)$ becomes too big. That is, since we constructed this interpolating solution assuming fixed m, such a solution can break down if m is large enough.

https://doi.org/10.1017/9781009023405.005 Published online by Cambridge University Press

Let us play a tiny bit more with the last interpolating formula (3.55) at criticality where $G_2^{(L)} = G_2^{(0)}$. Here, the finite-width difference (3.28) that governs the connected four-point correlator (3.29) becomes

$$\begin{aligned} G_4^{(L)} - \left(G_2^{(L)}\right)^2 &= \left(e^{2r} - 1\right)\left(G_2^{(0)}\right)^2 \\ &= 2r\left(G_2^{(0)}\right)^2 + O\left(r^2\right). \end{aligned} \tag{3.56}$$

This reproduces the *running* of the quartic coupling with the depth-to-width ratio (3.28). Similarly, the corresponding quantity governing the layer dependence of the connected six-point correlator (1.61) is given by

$$\begin{aligned} G_6^{(L)} - 3G_2^{(L)}G_4^{(L)} + 2\left(G_2^{(L)}\right)^3 &= \left(e^{6r} - 3e^{2r} + 2\right)\left(G_2^{(0)}\right)^3 \\ &= 12r^2\left(G_2^{(0)}\right)^3 + O\left(r^3\right), \end{aligned} \tag{3.57}$$

which scales like the depth-to-width ratio *squared*. Therefore, the connected six-point correlator is even more suppressed than the connected four-point correlator for large networks with sufficiently small depth-to-width ratio r. This is in accord with the comments we made in §1.3: neural networks obey *nearly-Gaussian* statistics, and the connected correlators have a hierarchical structure. In particular, we see here that the scaling of the correlators is controlled by the same small parameter r, with the higher-point connected correlators suppressed by a higher power of that parameter. This means that for small r, we should be able to consistently truncate our distribution and only compute up to a fixed order in r.

https://doi.org/10.1017/9781009023405.005 Published online by Cambridge University Press