11

Representation Learning

> *It can scarcely be denied that the supreme goal of all theory is to make the irreducible basic elements as simple and as few as possible without having to surrender the adequate representation of a single datum of experience.*
>
> Albert Einstein in a 1933 lecture, "On the Method of Theoretical Physics" [65].

Last chapter, we understood that linear models cannot learn features from data. Thus, the infinite-width limit is too simple to provide an adequate representation of deep learning: in order to include its irreducible basic element – representation learning – it is *qualitatively* important to study finite-width networks.

In the first half of this chapter, we'll analyze the leading correction to the gradient-descent update to the network output by extending its Taylor expansion to second order in the global learning rate η. After further seeing that a similar contribution arises in the first-order Taylor expansion of the update to the NTK, we'll then show that this correction is a finite-width effect. This upgrade of the NTK from fixed to dynamical indicates that for finite-width networks, the feature functions that comprise the NTK are themselves learning from the data over the course of training.

Unfortunately, the complete $O(1/n)$ contribution to the dynamics further includes terms that arise from Taylor-expanding the update to the network output to third order in the global learning rate η, and similarly Taylor-expanding the update to the NTK to second order in η. While it's *necessary* to include these contributions in order to actually compute the distribution of fully-trained finite-width networks, the $O(\eta^2)$ expansion of the network output and the $O(\eta)$ expansion of the NTK are *sufficient* to qualitatively investigate the mechanism for representation learning in these models.

With that in mind, in order to separate the pedagogy of representation learning from the messy phenomenological details of the real MLP, we'll spend the second half of this chapter focusing on a simplified model that's equivalent to this $O(\eta^2)$ truncation and gives a minimal qualitative picture of representation learning. These minimal models that we discuss form a valid and potentially useful class of machine learning models that perform representation learning, though annoyingly finite-width MLPs are not in this

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

class. Listening carefully to these real MLPs, we'll spend all of the next chapter (§∞) working out their $O(1/n)$ training dynamics in full intricate detail.

To begin, in §11.1 we'll work out this second-order-in-η contribution to the update to preactivations and the first-order-in-η contribution to the update to the NTK. This lets us source all representation learning from a single irreducible basic element, the *differential of the neural tangent kernel* (dNTK): just as the NTK governs the leading-order-in-η dynamics of the preactivations, the dNTK governs the leading-order-in-η dynamics of the NTK.

After thusly identifying the dNTK as a driver of representation learning, in §11.2 we'll recursively determine its correlation with the preactivations as a function of network layer ℓ. In detail, we'll first derive a stochastic forward equation for the dNTK and then evaluate the remaining recursions needed to determine the statistics of the joint preactivation–NTK–dNTK distribution at initialization. As such, this section mirrors the structure of our *RG-flow* analysis in §4 and §8. Importantly, we'll see that all the statistics involving the dNTK are $O(1/n)$ and thus only contribute at finite width.

In §11.3 we'll apply the principles of criticality and universality to analyze the new dNTK recursions. Since all of our hyperparameters have already been fixed by the parallel analysis of the preactivations in §5 – fixing the initialization hyperparameters – and the NTK in §9 – fixing the training hyperparameters – our focus here will be on evaluating the depth and width scaling of the dNTK statistics with these fixed hyperparameters. As you might guess, we'll find across our two universality classes (§11.3.1 and §11.3.2) that the effect of the dNTK – and therefore one source of representation learning – is proportional to our effective theory cutoff, the depth-to-width ratio L/n.

Having now firmly established that the NTK evolves at finite width – and having worked out an important contribution to its dynamics – in §11.4 we'll take a step back and look for a broader context, mirroring our discussion in §10.4 for infinite-width networks. To that end, in §11.4.1 we'll introduce a class of *nonlinear models* – with a particular focus on the *quadratic model* – and thus minimally extend the traditional workhorse of machine learning, the linear model. This quadratic model provides a *minimal model* of representation learning, independent of any neural-network abstraction. Moreover, these models are simple and completely analyzable, and yet are able to capture the essence of representation learning.

After solving the implied *nearly-linear quadratic regression* problem, in §11.4.2 we'll further provide a dual description of the quadratic model solution, which we'll call *nearly-kernel methods*. This will let us identify an object that corresponds to the dNTK in this minimal setting and show us how to make test-set predictions with a *trained* kernel that learns from the data. Overall, we hope this framework will be of further theoretical and practical interest as a new class of nearly-simple machine learning models that learn representations.

At this point, the connection between these nearly-kernel methods and finite-width networks – at least at order η^2 – will be nearly manifest, and in §11.4.3 we'll make it explicit. By doing so, we'll understand precisely how deep learning is a *nonminimal* model of representation learning. Ultimately, we'll conclude that the power of deep learning is the *deep* – the inductive bias of the network architecture induced by the layer-to-layer

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

RG flow – providing a particularly good choice of initial features as a starting point for *learning*. These observations will be quite helpful for us in interpreting our somewhat messy finite-width solution in the following chapter.

11.1 Differential of the Neural Tangent Kernel

Recall that in the first step of gradient descent, the change in the ℓ-th-layer parameters of any particular network is given by (7.11):

$$đ\theta_\mu^{(\ell)} \equiv \theta_\mu^{(\ell)}(t=1) - \theta_\mu^{(\ell)}(t=0) = -\eta \sum_\nu \lambda_{\mu\nu}^{(\ell)} \left(\sum_{k=1}^{n_L} \sum_{\tilde{\alpha}\in\mathcal{A}} \frac{\partial \mathcal{L}_\mathcal{A}}{\partial z_{k;\tilde{\alpha}}^{(L)}} \frac{dz_{k;\tilde{\alpha}}^{(L)}}{d\theta_\nu^{(\ell)}} \right) . \tag{11.1}$$

In this section, it will be helpful to specify explicitly which layer each parameter comes from. In particular, here $\theta_\mu^{(\ell)}$ denotes either an ℓ-th-layer bias $\theta_\mu^{(\ell)} \equiv b_i^{(\ell)}$ or an ℓ-th-layer weight $\theta_\mu^{(\ell)} \equiv W_{ij}^{(\ell)}$, and the ℓ-th-layer model-parameter indices μ, ν run over all the components of the bias vector $b_i^{(\ell)}$ *and* the weight matrix $W_{ij}^{(\ell)}$ in the ℓ-th layer *only*. Additionally, to emphasize that the learning-rate tensor $\lambda_{\mu\nu}^{(\ell)}$ only connects the parameters within a given layer ℓ, we've decorated it with a layer index for clarity. For now we'll let $\lambda_{\mu\nu}^{(\ell)}$ act arbitrarily within a layer, though ultimately we'll be interested in the case where it's diagonal, with two training hyperparameters per layer, $\lambda_b^{(\ell)}$ and $\lambda_W^{(\ell)}$, as usual.

As a further reminder, quantities without any explicit step argument are taken to be evaluated at initialization – though sometimes we may also explicitly denote $t = 0$ for extra emphasis – and our sample-index notation is *alpha-with-tilde* for the inputs in the training set, $\tilde{\alpha} \in \mathcal{A}$, *beta-with-dot* for inputs in the test set, $\dot{\beta} \in \mathcal{B}$, and *delta-with-no-decoration* for inputs that could be in either set, $\delta \in \mathcal{D} = \mathcal{A} \cup \mathcal{B}$.

Now, to go beyond the infinite-width limit, we'll need to expand the change in ℓ-th-layer preactivations to *second order* in the parameter update:

$$\begin{aligned} đz_{i;\delta}^{(\ell)} &\equiv z_{i;\delta}^{(\ell)}(t=1) - z_{i;\delta}^{(\ell)}(t=0) \\ &= \sum_{\ell_1=1}^{\ell} \sum_\mu \frac{dz_{i;\delta}^{(\ell)}}{d\theta_\mu^{(\ell_1)}} đ\theta_\mu^{(\ell_1)} + \frac{1}{2} \sum_{\ell_1,\ell_2=1}^{\ell} \sum_{\mu_1,\mu_2} \frac{d^2 z_{i;\delta}^{(\ell)}}{d\theta_{\mu_1}^{(\ell_1)} d\theta_{\mu_2}^{(\ell_2)}} đ\theta_{\mu_1}^{(\ell_1)} đ\theta_{\mu_2}^{(\ell_2)} + \ldots . \end{aligned} \tag{11.2}$$

Note that the ℓ-th-layer preactivations $z_{i;\delta}^{(\ell)}$ cannot depend on model parameters $\theta_\mu^{(\ell')}$ from layers ℓ' that are deeper than the ℓ-th layer. Thus, when $\ell' > \ell$, we have $dz_{i;\delta}^{(\ell)}/d\theta_\mu^{(\ell')} = 0$, and so we truncated our layer sums in the above expression at ℓ.

Next, we are going to slightly rewrite the parameter update equation (11.1) for the parameters $\theta_\mu^{(\ell_a)}$ appearing in our preactivation expansion (11.2), i.e., for those parameters in layers $\ell_a \leq \ell$ that contribute. To do so, we'll make use of the chain rule to decompose the derivative of the output-layer preactivations $z_{k;\tilde{\alpha}}^{(L)}$ with respect to the ℓ_a-th-layer model parameters as

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

$$\frac{dz^{(L)}_{k;\tilde{\alpha}}}{d\theta^{(\ell_a)}_{\nu}} = \sum_{j} \frac{dz^{(L)}_{k;\tilde{\alpha}}}{dz^{(\ell)}_{j;\tilde{\alpha}}} \frac{dz^{(\ell)}_{j;\tilde{\alpha}}}{d\theta^{(\ell_a)}_{\nu}}, \tag{11.3}$$

for an intermediate layer ℓ such that $\ell_a \leq \ell$. Using this decomposition, we can rewrite our parameter update (11.1) as

$$\text{đ}\theta^{(\ell_a)}_{\mu} = -\eta \sum_{\nu} \lambda^{(\ell_a)}_{\mu\nu} \left(\sum_{j,k,\tilde{\alpha}} \frac{\partial \mathcal{L}_{\mathcal{A}}}{\partial z^{(L)}_{k;\tilde{\alpha}}} \frac{dz^{(L)}_{k;\tilde{\alpha}}}{dz^{(\ell)}_{j;\tilde{\alpha}}} \frac{dz^{(\ell)}_{j;\tilde{\alpha}}}{d\theta^{(\ell_a)}_{\nu}} \right) = -\eta \sum_{\nu,j,\tilde{\alpha}} \lambda^{(\ell_a)}_{\mu\nu} \, \epsilon^{(\ell)}_{j;\tilde{\alpha}} \frac{dz^{(\ell)}_{j;\tilde{\alpha}}}{d\theta^{(\ell_a)}_{\nu}}, \tag{11.4}$$

where in the last equality we introduced an *ℓ-th-layer error factor*:

$$\epsilon^{(\ell)}_{j;\tilde{\alpha}} \equiv \sum_{k=1}^{n_L} \frac{\partial \mathcal{L}_{\mathcal{A}}}{\partial z^{(L)}_{k;\tilde{\alpha}}} \frac{dz^{(L)}_{k;\tilde{\alpha}}}{dz^{(\ell)}_{j;\tilde{\alpha}}} = \frac{d\mathcal{L}_{\mathcal{A}}}{dz^{(\ell)}_{j;\tilde{\alpha}}}. \tag{11.5}$$

Substituting this form of the parameter update (11.4) into the ℓ-th-layer preactivation update (11.2), our second-order expansion becomes

$$\begin{aligned} \text{đ}z^{(\ell)}_{i;\delta} = &-\eta \sum_{j,\tilde{\alpha}} \left(\sum_{\ell_1=1}^{\ell} \sum_{\mu,\nu} \lambda^{(\ell_1)}_{\mu\nu} \frac{dz^{(\ell)}_{i;\delta}}{d\theta^{(\ell_1)}_{\mu}} \frac{dz^{(\ell)}_{j;\tilde{\alpha}}}{d\theta^{(\ell_1)}_{\nu}} \right) \epsilon^{(\ell)}_{j;\tilde{\alpha}} \\ &+ \frac{\eta^2}{2} \sum_{j_1,j_2,\tilde{\alpha}_1,\tilde{\alpha}_2} \left(\sum_{\ell_1,\ell_2=1}^{\ell} \sum_{\substack{\mu_1,\nu_1,\\ \mu_2,\nu_2}} \lambda^{(\ell_1)}_{\mu_1\nu_1} \lambda^{(\ell_2)}_{\mu_2\nu_2} \frac{d^2 z^{(\ell)}_{i;\delta}}{d\theta^{(\ell_1)}_{\mu_1} d\theta^{(\ell_2)}_{\mu_2}} \frac{dz^{(\ell)}_{j_1;\tilde{\alpha}_1}}{d\theta^{(\ell_1)}_{\nu_1}} \frac{dz^{(\ell)}_{j_2;\tilde{\alpha}_2}}{d\theta^{(\ell_2)}_{\nu_2}} \right) \epsilon^{(\ell)}_{j_1;\tilde{\alpha}_1} \epsilon^{(\ell)}_{j_2;\tilde{\alpha}_2} \\ &+ \cdots , \end{aligned} \tag{11.6}$$

which is *quadratic* in such error factors. Here, it was essential that we treated the parameters in a per-layer manner and that each learning-rate tensor $\lambda^{(\ell_a)}_{\mu\nu}$ was restricted to a single layer ℓ_a; had we not done that, our decomposition (11.4) and update equation (11.6) would have been far more complicated.

Naturally, the object in the first parenthesis of the update equation (11.6) is the stochastic *ℓ-th-layer NTK* (8.4)

$$\widehat{H}^{(\ell)}_{i_1 i_2;\delta_1\delta_2} \equiv \sum_{\ell_1=1}^{\ell} \sum_{\mu,\nu} \lambda^{(\ell_1)}_{\mu\nu} \frac{dz^{(\ell)}_{i_1;\delta_1}}{d\theta^{(\ell_1)}_{\mu}} \frac{dz^{(\ell)}_{i_2;\delta_2}}{d\theta^{(\ell_1)}_{\nu}}, \tag{11.7}$$

as you know quite well by now, though in this version of the definition we represent the sum over layers explicitly, and the sum over parameter indices μ, ν runs per layer.

In contrast, the object in the second parenthesis is new.[1] Let's call this object the stochastic **ℓ-th-layer differential of the neural tangent kernel** (dNTK) and symbolize it as

[1] This object first appeared, unnamed, in both [66] and [67] around the same time. Here, we'll compute its recursion, determine its scaling with depth, and emphasize its physical importance by highlighting its connection to representation learning.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

$$\widehat{\mathrm{d}H}^{(\ell)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2} \equiv \sum_{\ell_1,\ell_2=1}^{\ell} \sum_{\substack{\mu_1,\nu_1,\\ \mu_2,\nu_2}} \lambda^{(\ell_1)}_{\mu_1\nu_1}\lambda^{(\ell_2)}_{\mu_2\nu_2} \frac{d^2 z^{(\ell)}_{i_0;\delta_0}}{d\theta^{(\ell_1)}_{\mu_1} d\theta^{(\ell_2)}_{\mu_2}} \frac{dz^{(\ell)}_{i_1;\delta_1}}{d\theta^{(\ell_1)}_{\nu_1}} \frac{dz^{(\ell)}_{i_2;\delta_2}}{d\theta^{(\ell_2)}_{\nu_2}}. \tag{11.8}$$

Here, the hats on both the NTK and the dNTK remind us that these objects are stochastic, depending on the particular realization of the model parameters at initialization. Also, from its definition note that the dNTK is symmetric in its second and third paired sets of indices $(i_1,\delta_1) \leftrightarrow (i_2,\delta_2)$, while the first neural-sample index (i_0,δ_0) is distinguished from the other two.

Using both definitions (11.7) and (11.8), our second-order expansion (11.6) can be more compactly written as

$$đz^{(\ell)}_{i;\delta} = -\eta \sum_{j,\tilde{\alpha}} \widehat{H}^{(\ell)}_{ij;\delta\tilde{\alpha}} \epsilon^{(\ell)}_{j;\tilde{\alpha}} + \frac{\eta^2}{2} \sum_{j_1,j_2,\tilde{\alpha}_1,\tilde{\alpha}_2} \widehat{\mathrm{d}H}^{(\ell)}_{ij_1j_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2} \epsilon^{(\ell)}_{j_1;\tilde{\alpha}_1} \epsilon^{(\ell)}_{j_2;\tilde{\alpha}_2} + \cdots . \tag{11.9}$$

In other words, we have a power series in error factors. To ultimately understand how the preactivations evolve under gradient descent at leading order in $1/n$, we'll actually need to extend this expansion to order η^3, which in turn will require that we introduce a few additional tensors. Rather than worry about that now, we'll put it off to §∞. Regardless of those additional higher-order terms, from (11.9) we already see that we'll need to know the joint statistics of the preactivations – encoding the error factors $\epsilon^{(\ell)}_{j;\tilde{\alpha}}$ – the NTK $\widehat{H}^{(\ell)}_{ij;\delta\tilde{\alpha}}$, and the dNTK $\widehat{\mathrm{d}H}^{(\ell)}_{ij_1j_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2}$.

Finally, as an explanation for our choice of name and symbol for the dNTK, consider the leading-order update to the ℓ-th-layer NTK after a step of gradient descent:

$$\begin{aligned}
đH^{(\ell)}_{i_1i_2;\delta_1\delta_2} &\equiv H^{(\ell)}_{i_1i_2;\delta_1\delta_2}(t=1) - H^{(\ell)}_{i_1i_2;\delta_1\delta_2}(t=0) \\
&= \sum_{\ell_1=1}^{\ell} \sum_{\mu_1} \frac{dH^{(\ell)}_{i_1i_2;\delta_1\delta_2}}{d\theta^{(\ell_1)}_{\mu_1}} đ\theta^{(\ell_1)}_{\mu_1} + \cdots \\
&= -\eta \sum_{\ell_1=1}^{\ell} \sum_{\mu_1} \left[\frac{d}{d\theta^{(\ell_1)}_{\mu_1}} \left(\sum_{\ell_2=1}^{\ell} \sum_{\mu_2,\nu_2} \lambda^{(\ell_2)}_{\mu_2\nu_2} \frac{dz^{(\ell)}_{i_1;\delta_1}}{d\theta^{(\ell_2)}_{\mu_2}} \frac{dz^{(\ell)}_{i_2;\delta_2}}{d\theta^{(\ell_2)}_{\nu_2}} \right) \right] \\
&\qquad \times \left[\sum_{\nu_1} \lambda^{(\ell_1)}_{\mu_1\nu_1} \sum_{j,\tilde{\alpha}} \frac{dz^{(\ell)}_{j;\tilde{\alpha}}}{d\theta^{(\ell_1)}_{\nu_1}} \epsilon^{(\ell)}_{j;\tilde{\alpha}} \right] + \cdots \\
&= -\eta \sum_{j,\tilde{\alpha}} \left[\sum_{\ell_1,\ell_2=1}^{\ell} \sum_{\substack{\mu_1,\nu_1,\\ \mu_2,\nu_2}} \lambda^{(\ell_1)}_{\mu_1\nu_1}\lambda^{(\ell_2)}_{\mu_2\nu_2} \frac{d^2 z^{(\ell)}_{i_1;\delta_1}}{d\theta^{(\ell_1)}_{\mu_1} d\theta^{(\ell_2)}_{\mu_2}} \frac{dz^{(\ell)}_{i_2;\delta_2}}{d\theta^{(\ell_2)}_{\nu_2}} \frac{dz^{(\ell)}_{j;\tilde{\alpha}}}{d\theta^{(\ell_1)}_{\nu_1}} \right] \epsilon^{(\ell)}_{j;\tilde{\alpha}} \\
&\quad - \eta \sum_{j,\tilde{\alpha}} \left[\sum_{\ell_1,\ell_2=1}^{\ell} \sum_{\substack{\mu_1,\nu_1,\\ \mu_2,\nu_2}} \lambda^{(\ell_1)}_{\mu_1\nu_1}\lambda^{(\ell_2)}_{\mu_2\nu_2} \frac{dz^{(\ell)}_{i_1;\delta_1}}{d\theta^{(\ell_2)}_{\mu_2}} \frac{d^2 z^{(\ell)}_{i_2;\delta_2}}{d\theta^{(\ell_1)}_{\mu_1} d\theta^{(\ell_2)}_{\nu_2}} \frac{dz^{(\ell)}_{j;\tilde{\alpha}}}{d\theta^{(\ell_1)}_{\nu_1}} \right] \epsilon^{(\ell)}_{j;\tilde{\alpha}} + \cdots \\
&= -\eta \sum_{j,\tilde{\alpha}} \left(\widehat{\mathrm{d}H}^{(\ell)}_{i_1i_2j;\delta_1\delta_2\tilde{\alpha}} + \widehat{\mathrm{d}H}^{(\ell)}_{i_2i_1j;\delta_2\delta_1\tilde{\alpha}} \right) \epsilon^{(\ell)}_{j;\tilde{\alpha}} + \cdots .
\end{aligned} \tag{11.10}$$

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

Here, in the third line we inserted the definition of the NTK (11.7) and the parameter update (11.4) for $\ell_1 \leq \ell$, and on the final line we used the definition of the dNTK (11.8). Thus we see that the dNTK – when multiplied by the global learning rate and contracted with an ℓ-th-layer error factor – gives the update to the ℓ-th-layer NTK after a step of gradient descent.[2]

Since we know that the infinite-width NTK is *frozen*, $\widehat{H}^{(\ell)} \to \Theta^{(\ell)}$, the relation between the NTK update and the dNTK implies that the dNTK must be a finite-width effect, vanishing in the strict infinite-width limit $\widehat{\mathrm{d}H}^{(\ell)} \to 0$. Similarly, at infinite width we truncated the preactivation updates (11.9) to be linear in the global learning rate η, cf. (10.2). In the next section, we will verify all of this by computing the dNTK recursively and showing explicitly that $\widehat{\mathrm{d}H}^{(\ell)} = O(1/n)$.

11.2 RG Flow of the dNTK

As its title suggests, the structure of this section parallels §4 – where we worked out the layer-to-layer representation group (RG) flow of the preactivation distribution $p\big(z^{(\ell)}\big|\mathcal{D}\big)$ – and §8 – where we worked out the layer-to-layer RG flow of the NTK-preactivation joint distribution $p\big(z^{(\ell)}, \widehat{H}^{(\ell)}\big|\mathcal{D}\big)$. Specifically, we will now work out the effective ℓ-th-layer joint distribution of the preactivations, the NTK, and the dNTK:

$$p\Big(z^{(\ell)}, \widehat{H}^{(\ell)}, \widehat{\mathrm{d}H}^{(\ell)}\Big|\mathcal{D}\Big). \tag{11.11}$$

This analysis is important for two reasons: *(i)* understanding the statistics of this ℓ-th-layer joint distribution at order $1/n$ is a necessary prerequisite for understanding the leading nontrivial finite-width corrections for deep neural networks trained with gradient-based learning; *(ii)* in §11.4 we will see that a nonvanishing dNTK is sufficient for a network to exhibit representation learning, and thus by showing that the dNTK is of order $1/n$, we will firmly establish that the leading-order finite-width effective theory is able to describe this essential property of deep learning.

Zeroth, we'll establish the stochastic iteration equation for the dNTK (§11.2.0). Then, beginning our statistical analysis, first we'll see that the dNTK vanishes identically in the first layer (§11.2.1). Second, we'll see that there's a nontrivial cross correlation between the dNTK and the preactivations in the second layer (§11.2.2). Third and finally, we'll work out a general recursion that controls the accumulation of such dNTK–preactivation cross correlations in deeper layers (§11.2.3).

[2] Please don't confuse our italicized, crossed, and unhatted notation, $đH^{(\ell)}_{i_1 i_2;\delta_1\delta_2}$, representing the first update to the NTK, with our unitalicized, uncrossed, and hatted notation, $\widehat{\mathrm{d}H}^{(\ell)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2}$, representing the dNTK. In this chapter we will focus on the statistics of the dNTK, and we will not use this notation when evaluating the NTK dynamics in the following chapter.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

11.2.0 Forward Equation for the dNTK

Just as we needed to derive a stochastic forward iteration equation for the NTK (8.12) in §8.0 before working out recursions for its statistics, here we'll derive such an equation for the dNTK.

Let's start by writing out the definition of the dNTK (11.8) at layer $(\ell+1)$:

$$\widehat{\mathrm{d}H}^{(\ell+1)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2} \equiv \sum_{\ell_1,\ell_2=1}^{\ell+1} \left[\sum_{\substack{\mu_1,\nu_1,\\ \mu_2,\nu_2}} \lambda^{(\ell_1)}_{\mu_1\nu_1} \lambda^{(\ell_2)}_{\mu_2\nu_2} \frac{d^2 z^{(\ell+1)}_{i_0;\delta_0}}{d\theta^{(\ell_1)}_{\mu_1} d\theta^{(\ell_2)}_{\mu_2}} \frac{dz^{(\ell+1)}_{i_1;\delta_1}}{d\theta^{(\ell_1)}_{\nu_1}} \frac{dz^{(\ell+1)}_{i_2;\delta_2}}{d\theta^{(\ell_2)}_{\nu_2}} \right]. \tag{11.12}$$

To determine its forward equation, we need to explicitly evaluate the derivatives with respect to the $(\ell+1)$-th-layer parameters and also rewrite all the $(\ell+1)$-th-layer quantities in terms of the ℓ-th-layer quantities using the chain rule. Depending on the values of ℓ_1 and ℓ_2, there are thus three cases to consider for the double summation over layers.

First, when both layers are maximal $\ell_1 = \ell_2 = \ell+1$, there is no contribution. Recalling for one final time the preactivation forward equation,

$$z^{(\ell+1)}_{i;\delta} = b^{(\ell+1)}_i + \sum_{j=1}^{n_\ell} W^{(\ell+1)}_{ij} \sigma^{(\ell)}_{j;\delta}, \tag{11.13}$$

we see that the $(\ell+1)$-th-layer preactivations are always linear in the $(\ell+1)$-th-layer model parameters $\theta^{(\ell+1)}_\mu$. Thus, in this case the second derivative in the dNTK definition (11.12) will vanish.

Second, when $\ell_1 = \ell+1$ and $\ell_2 < \ell+1$, there is a contribution from the $(\ell+1)$-th-layer weights but not from the $(\ell+1)$-th-layer biases. Considering the bias $\theta^{(\ell_1)}_{\mu_1} = b^{(\ell+1)}_j$, the $(\ell+1)$-th-layer derivative gives a Kronecker delta,

$$\frac{dz^{(\ell+1)}_{i;\delta}}{db^{(\ell+1)}_j} = \delta_{ij}, \tag{11.14}$$

and so the second derivative again vanishes,

$$\frac{d^2 z^{(\ell+1)}_{i;\delta}}{db^{(\ell+1)}_j d\theta^{(\ell_2)}_{\mu_2}} = 0. \tag{11.15}$$

Instead considering the weight matrix $\theta^{(\ell_1)}_{\mu_1} = W^{(\ell+1)}_{jk}$, the $(\ell+1)$-th-layer derivative is not a constant,

$$\frac{dz^{(\ell+1)}_{i;\delta}}{dW^{(\ell+1)}_{jk}} = \delta_{ij} \sigma^{(\ell)}_{k;\delta}. \tag{11.16}$$

Thus, the second derivative evaluates to something nontrivial,

$$\frac{d^2 z^{(\ell+1)}_{i;\delta}}{dW^{(\ell+1)}_{jk} d\theta^{(\ell_2)}_{\mu_2}} = \delta_{ij} \sigma'^{(\ell)}_{k;\delta} \frac{dz^{(\ell)}_{k;\delta}}{d\theta^{(\ell_2)}_{\mu_2}}, \tag{11.17}$$

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

while the remaining first derivative gives

$$\frac{dz^{(\ell+1)}_{i;\delta}}{d\theta^{(\ell_2)}_{\nu_2}} = \sum_k W^{(\ell+1)}_{ik} \sigma'^{(\ell)}_{k;\delta} \frac{dz^{(\ell)}_{k;\delta}}{d\theta^{(\ell_2)}_{\nu_2}}\,, \tag{11.18}$$

with the use of the chain rule. Plugging in these three derivative evaluations (11.16), (11.17), and (11.18) to evaluate terms in the dNTK definition (11.12) with $\ell_1 = \ell + 1$ and $\ell_2 < \ell + 1$, we find

$$\begin{aligned}
&\sum_{\ell_2=1}^{\ell} \sum_{j,k} \lambda_{W^{(\ell+1)}_{jk} W^{(\ell+1)}_{jk}} \sum_{\mu_2,\nu_2} \lambda^{(\ell_2)}_{\mu_2\nu_2} \frac{d^2 z^{(\ell+1)}_{i_0;\delta_0}}{dW^{(\ell+1)}_{jk} d\theta^{(\ell_2)}_{\mu_2}} \frac{dz^{(\ell+1)}_{i_1;\delta_1}}{dW^{(\ell+1)}_{jk}} \frac{dz^{(\ell+1)}_{i_2;\delta_2}}{d\theta^{(\ell_2)}_{\nu_2}} \\
&\quad = \frac{\lambda^{(\ell+1)}_W}{n_\ell} \sum_{\ell_2=1}^{\ell} \sum_{j,k} \sum_{\mu_2,\nu_2} \lambda^{(\ell_2)}_{\mu_2\nu_2} \left(\delta_{i_0 j}\, \sigma'^{(\ell)}_{k;\delta_0} \frac{dz^{(\ell)}_{k;\delta_0}}{d\theta^{(\ell_2)}_{\mu_2}} \right) \left(\delta_{i_1 j}\, \sigma^{(\ell)}_{k;\delta_1} \right) \left(\sum_{k_2} W^{(\ell+1)}_{i_2 k_2} \sigma'^{(\ell)}_{k_2;\delta_2} \frac{dz^{(\ell)}_{k_2;\delta_2}}{d\theta^{(\ell_2)}_{\nu_2}} \right) \\
&\quad = \frac{\lambda^{(\ell+1)}_W}{n_\ell} \delta_{i_0 i_1} \sum_{k_0,k_2} W^{(\ell+1)}_{i_2 k_2} \sigma'^{(\ell)}_{k_0;\delta_0} \sigma^{(\ell)}_{k_0;\delta_1} \sigma'^{(\ell)}_{k_2;\delta_2} \widehat{H}^{(\ell)}_{k_0 k_2;\delta_0\delta_2}.
\end{aligned} \tag{11.19}$$

To get this result, on the second line we implemented our choice of a single intralayer learning rate for the weights (8.5),

$$\lambda_{W^{(\ell+1)}_{j_1 k_1} W^{(\ell+1)}_{j_2 k_2}} = \delta_{j_1 j_2} \delta_{k_1 k_2} \frac{\lambda^{(\ell+1)}_W}{n_\ell}, \tag{11.20}$$

importantly rescaled by n_ℓ, and on the third line we used the definition of the stochastic NTK (11.7) and relabeled a dummy index. By symmetry, there must be a similar contribution when instead $\ell_2 = \ell + 1$ and $\ell_1 < \ell + 1$. This term is given by (11.19) after swapping neural-sample index pairs $(i_1, \delta_1) \leftrightarrow (i_2, \delta_2)$.

Third and finally, when both $\ell_1 < \ell+1$ and $\ell_2 < \ell+1$, both the biases and the weights contribute to the second derivative. When $\theta^{(\ell_1)}_\mu$ and $\theta^{(\ell_2)}_\nu$ are not from the $(\ell+1)$-th layer, we computed their first derivative in (11.18), and their second derivative is given by

$$\frac{d^2 z^{(\ell+1)}_{i;\delta}}{d\theta^{(\ell_1)}_{\mu_1} d\theta^{(\ell_2)}_{\mu_2}} = \sum_k W^{(\ell+1)}_{ik} \sigma''^{(\ell)}_{k;\delta} \frac{dz^{(\ell)}_{k;\delta}}{d\theta^{(\ell_1)}_{\mu_1}} \frac{dz^{(\ell)}_{k;\delta}}{d\theta^{(\ell_2)}_{\mu_2}} + \sum_k W^{(\ell+1)}_{ik} \sigma'^{(\ell)}_{k;\delta} \frac{d^2 z^{(\ell)}_{k;\delta}}{d\theta^{(\ell_1)}_{\mu_1} d\theta^{(\ell_2)}_{\mu_2}}. \tag{11.21}$$

Multiplying these second-derivative terms by the learning-rate tensors $\lambda^{(\ell_1)}_{\mu_1\nu_1}\, \lambda^{(\ell_2)}_{\mu_2\nu_2}$ and by the appropriate first derivatives (11.18), and implementing all the sums over ℓ_1, ℓ_2, μ_1, ν_1, μ_2, ν_2 in the dNTK definition (11.12), the first term from (11.21) gives a contribution of

$$\sum_{k_0,k_1,k_2} W^{(\ell+1)}_{i_0 k_0} W^{(\ell+1)}_{i_1 k_1} W^{(\ell+1)}_{i_2 k_2} \sigma''^{(\ell)}_{k_0;\delta_0} \sigma'^{(\ell)}_{k_1;\delta_1} \sigma'^{(\ell)}_{k_2;\delta_2} \widehat{H}^{(\ell)}_{k_0 k_1;\delta_0\delta_1} \widehat{H}^{(\ell)}_{k_0 k_2;\delta_0\delta_2}, \tag{11.22}$$

where we made use of the NTK definition (11.7) twice, while the second term from (11.21) gives a contribution of

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

$$\sum_{k_0,k_1,k_2} W^{(\ell+1)}_{i_0k_0} W^{(\ell+1)}_{i_1k_1} W^{(\ell+1)}_{i_2k_2} \sigma'^{(\ell)}_{k_0;\delta_0} \sigma'^{(\ell)}_{k_1;\delta_1} \sigma'^{(\ell)}_{k_2;\delta_2} \widehat{\mathrm{d}H}^{(\ell)}_{k_0k_1k_2;\delta_0\delta_1\delta_2}, \tag{11.23}$$

where we made use of the dNTK definition (11.8) once.

Combining our three types of contributions (11.19), (11.22), and (11.23), we get a rather involved stochastic iteration equation:

$$\begin{aligned}
\widehat{\mathrm{d}H}^{(\ell+1)}_{i_0i_1i_2;\delta_0\delta_1\delta_2} = &\sum_{k_0,k_1,k_2} W^{(\ell+1)}_{i_0k_0} W^{(\ell+1)}_{i_1k_1} W^{(\ell+1)}_{i_2k_2} \sigma'^{(\ell)}_{k_0;\delta_0} \sigma'^{(\ell)}_{k_1;\delta_1} \sigma'^{(\ell)}_{k_2;\delta_2} \widehat{\mathrm{d}H}^{(\ell)}_{k_0k_1k_2;\delta_0\delta_1\delta_2} \\
&+ \sum_{k_0,k_1,k_2} W^{(\ell+1)}_{i_0k_0} W^{(\ell+1)}_{i_1k_1} W^{(\ell+1)}_{i_2k_2} \sigma''^{(\ell)}_{k_0;\delta_0} \sigma'^{(\ell)}_{k_1;\delta_1} \sigma'^{(\ell)}_{k_2;\delta_2} \widehat{H}^{(\ell)}_{k_0k_1;\delta_0\delta_1} \widehat{H}^{(\ell)}_{k_0k_2;\delta_0\delta_2} \\
&+ \frac{\lambda_W^{(\ell+1)}}{n_\ell} \delta_{i_0i_1} \sum_{k_0,k_2} W^{(\ell+1)}_{i_2k_2} \sigma'^{(\ell)}_{k_0;\delta_0} \sigma^{(\ell)}_{k_0;\delta_1} \sigma'^{(\ell)}_{k_2;\delta_2} \widehat{H}^{(\ell)}_{k_0k_2;\delta_0\delta_2} \\
&+ \frac{\lambda_W^{(\ell+1)}}{n_\ell} \delta_{i_0i_2} \sum_{k_0,k_1} W^{(\ell+1)}_{i_1k_1} \sigma'^{(\ell)}_{k_0;\delta_0} \sigma'^{(\ell)}_{k_1;\delta_1} \sigma^{(\ell)}_{k_0;\delta_2} \widehat{H}^{(\ell)}_{k_0k_1;\delta_0\delta_1}.
\end{aligned} \tag{11.24}$$

This is the **forward equation for the dNTK**, and we're next going to work out the recursions that determine its statistics.

11.2.1 First Layer: Zero dNTK

Recall from §4.1 and §8.1 that at initialization the first-layer preactivations,

$$z^{(1)}_{i;\delta} \equiv b^{(1)}_i + \sum_{k=1}^{n_0} W^{(1)}_{ik} x_{k;\delta}\,, \tag{11.25}$$

are distributed according to a zero-mean Gaussian distribution (4.23) and that the NTK $\widehat{H}^{(1)}_{i_1i_2;\delta_1\delta_2} = \delta_{i_1i_2} H^{(1)}_{\delta_1\delta_2}$ is deterministic (8.23).

As we discussed just before, since the preactivations are linear in the model parameters, their second derivative must vanish. Thus, the dNTK trivially vanishes in the first layer:

$$\widehat{\mathrm{d}H}^{(1)}_{i_0i_1i_2;\delta_0\delta_1\delta_2} \equiv \sum_{\substack{\mu_1,\nu_1,\\ \mu_2,\nu_2}} \lambda^{(1)}_{\mu_1\nu_1} \lambda^{(1)}_{\mu_2\nu_2} \frac{d^2 z^{(1)}_{i_0;\delta_0}}{d\theta^{(1)}_{\mu_1} d\theta^{(1)}_{\mu_2}} \frac{dz^{(1)}_{i_1;\delta_1}}{d\theta^{(1)}_{\nu_1}} \frac{dz^{(1)}_{i_2;\delta_2}}{d\theta^{(1)}_{\nu_2}} = 0. \tag{11.26}$$

This gives the initial condition for our recursions.

Note that this result should have been expected as the first-layer NTK (8.23) is independent of the model parameters and thus cannot change with any training. As we saw before for the first-layer preactivations and first-layer NTK – zero-mean Gaussian and fixed, respectively – this first-layer result for the dNTK will be representative of its infinite-width limit for all layers.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

11.2.2 Second Layer: Nonzero dNTK

Now, let's analyze the dNTK (11.24) in the second layer. Remembering again that the first-layer NTK (8.23) is deterministic and diagonal in its neural indices as $\widehat{H}^{(1)}_{i_1 i_2;\delta_1\delta_2} = \delta_{i_1 i_2} H^{(1)}_{\delta_1\delta_2}$, and remembering for the first time that the dNTK vanishes in the first layer $\widehat{\mathrm{d}H}^{(1)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2} = 0$ from (11.26), the forward equation (11.24) in the second layer simplifies to

$$\begin{aligned}\widehat{\mathrm{d}H}^{(2)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2} &= H^{(1)}_{\delta_0\delta_1} H^{(1)}_{\delta_0\delta_2} \sum_{k=1}^{n_1} W^{(2)}_{i_0 k} W^{(2)}_{i_1 k} W^{(2)}_{i_2 k} \sigma''^{(1)}_{k;\delta_0} \sigma'^{(1)}_{k;\delta_1} \sigma'^{(1)}_{k;\delta_2} \\ &\quad + \delta_{i_0 i_1} \frac{\lambda^{(2)}_W}{n_1} H^{(1)}_{\delta_0\delta_2} \sum_{k=1}^{n_1} W^{(2)}_{i_2 k} \sigma'^{(1)}_{k;\delta_0} \sigma^{(1)}_{k;\delta_1} \sigma'^{(1)}_{k;\delta_2} \\ &\quad + \delta_{i_0 i_2} \frac{\lambda^{(2)}_W}{n_1} H^{(1)}_{\delta_0\delta_1} \sum_{k=1}^{n_1} W^{(2)}_{i_1 k} \sigma'^{(1)}_{k;\delta_0} \sigma'^{(1)}_{k;\delta_1} \sigma^{(1)}_{k;\delta_2} . \end{aligned} \tag{11.27}$$

Interestingly, since each term has an odd number of weights, the mean of the dNTK will vanish, and we'll have to look at cross correlations to find leading dNTK statistics that are nonvanishing.

The simplest cross correlation is with a single preactivation. Considering the product of the second-layer dNTK (11.27) with second-layer preactivations,

$$z^{(2)}_{i;\delta} = b^{(2)}_i + \sum_{k=1}^{n_1} W^{(2)}_{ik} \sigma^{(1)}_{k;\delta} , \tag{11.28}$$

and taking an expectation, we find

$$\begin{aligned}&\mathbb{E}\left[\widehat{\mathrm{d}H}^{(2)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2} z^{(2)}_{i_3;\delta_3}\right] \\ &\quad = H^{(1)}_{\delta_0\delta_1} H^{(1)}_{\delta_0\delta_2} \left(\frac{C^{(2)}_W}{n_1}\right)^2 (\delta_{i_0 i_3}\delta_{i_1 i_2} + \delta_{i_0 i_1}\delta_{i_2 i_3} + \delta_{i_0 i_2}\delta_{i_1 i_3}) \sum_{k=1}^{n_1} \mathbb{E}\left[\sigma''^{(1)}_{k;\delta_0} \sigma'^{(1)}_{k;\delta_1} \sigma'^{(1)}_{k;\delta_2} \sigma^{(1)}_{k;\delta_3}\right] \\ &\qquad + \frac{\lambda^{(2)}_W}{n_1} H^{(1)}_{\delta_0\delta_2} \delta_{i_0 i_1}\delta_{i_2 i_3} \frac{C^{(2)}_W}{n_1} \sum_{k=1}^{n_1} \mathbb{E}\left[\sigma'^{(1)}_{k;\delta_0} \sigma^{(1)}_{k;\delta_1} \sigma'^{(1)}_{k;\delta_2} \sigma^{(1)}_{k;\delta_3}\right] \\ &\qquad + \frac{\lambda^{(2)}_W}{n_1} H^{(1)}_{\delta_0\delta_1} \delta_{i_0 i_2}\delta_{i_1 i_3} \frac{C^{(2)}_W}{n_1} \sum_{k=1}^{n_1} \mathbb{E}\left[\sigma'^{(1)}_{k;\delta_0} \sigma'^{(1)}_{k;\delta_1} \sigma^{(1)}_{k;\delta_2} \sigma^{(1)}_{k;\delta_3}\right] \\ &\quad = \frac{1}{n_1} (\delta_{i_0 i_3}\delta_{i_1 i_2} + \delta_{i_0 i_1}\delta_{i_2 i_3} + \delta_{i_0 i_2}\delta_{i_1 i_3}) C^{(2)}_W H^{(1)}_{\delta_0\delta_1} C^{(2)}_W H^{(1)}_{\delta_0\delta_2} \left\langle \sigma''_{\delta_0} \sigma'_{\delta_1} \sigma'_{\delta_2} \sigma_{\delta_3} \right\rangle_{G^{(1)}} \\ &\qquad + \frac{1}{n_1} \delta_{i_0 i_1}\delta_{i_2 i_3} \lambda^{(2)}_W C^{(2)}_W H^{(1)}_{\delta_0\delta_2} \left\langle \sigma'_{\delta_0} \sigma_{\delta_1} \sigma'_{\delta_2} \sigma_{\delta_3} \right\rangle_{G^{(1)}} \\ &\qquad + \frac{1}{n_1} \delta_{i_0 i_2}\delta_{i_1 i_3} \lambda^{(2)}_W C^{(2)}_W H^{(1)}_{\delta_0\delta_1} \left\langle \sigma'_{\delta_0} \sigma'_{\delta_1} \sigma_{\delta_2} \sigma_{\delta_3} \right\rangle_{G^{(1)}} . \end{aligned} \tag{11.29}$$

To get this final result, in the first equality we dropped the bias term from (11.28), since it vanishes under the expectation, and performed various Wick contractions of the weights

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

using $\mathbb{E}\left[W^{(2)}_{i_1 j_1} W^{(2)}_{i_2 j_2}\right] = \delta_{i_1 i_2}\delta_{j_1 j_2} C^{(2)}_W / n_1$ (2.20). For the second equality, we remembered that the first-layer preactivation distribution is a zero-mean Gaussian with a two-point correlator that's diagonal in neural indices, $\mathbb{E}\left[z^{(1)}_{i_1;\delta_1} z^{(1)}_{i_2;\delta_2}\right] = \delta_{i_1 i_2} G^{(1)}_{\delta_1 \delta_2}$ (4.23), and used this to swap full expectations for Gaussian expectations and then performed the sums.

As we did before for the NTK variance (8.31) and the NTK–preactivation cross correlation (8.37), it is convenient to decompose this dNTK–preactivation cross correlation (11.29) into two tensors with sample indices only:

$$\mathbb{E}\left[\widehat{\mathrm{d}H}^{(2)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2} z^{(2)}_{i_3;\delta_3}\right] \equiv \frac{1}{n_1}\left[\delta_{i_0 i_3}\delta_{i_1 i_2} P^{(2)}_{\delta_0\delta_1\delta_2\delta_3} + \delta_{i_0 i_1}\delta_{i_2 i_3} Q^{(2)}_{\delta_0\delta_1\delta_2\delta_3} + \delta_{i_0 i_2}\delta_{i_1 i_3} Q^{(2)}_{\delta_0\delta_2\delta_1\delta_3}\right]. \tag{11.30}$$

Comparing this with our explicit formula for the second-layer cross correlation (11.29), we see that these tensors have the following definitions:

$$P^{(2)}_{\delta_0\delta_1\delta_2\delta_3} \equiv \left(C^{(2)}_W\right)^2 H^{(1)}_{\delta_0\delta_1} H^{(1)}_{\delta_0\delta_2} \left\langle \sigma''_{\delta_0} \sigma'_{\delta_1} \sigma'_{\delta_2} \sigma_{\delta_3} \right\rangle_{G^{(1)}} , \tag{11.31}$$

$$Q^{(2)}_{\delta_0\delta_1\delta_2\delta_3} \equiv \left(C^{(2)}_W\right)^2 H^{(1)}_{\delta_0\delta_1} H^{(1)}_{\delta_0\delta_2} \left\langle \sigma''_{\delta_0} \sigma'_{\delta_1} \sigma'_{\delta_2} \sigma_{\delta_3} \right\rangle_{G^{(1)}} + \lambda^{(2)}_W C^{(2)}_W H^{(1)}_{\delta_0\delta_2} \left\langle \sigma'_{\delta_0} \sigma_{\delta_1} \sigma'_{\delta_2} \sigma_{\delta_3} \right\rangle_{G^{(1)}} , \tag{11.32}$$

and that they are manifestly of order one.[3] Overall, the dNTK–preactivation cross correlation (11.30) is of order $1/n_1$, vanishing in the strict infinite-width limit $n_1 \to \infty$.

These tensors (11.31) and (11.32) – and their deeper-layer siblings – control all the leading finite-width correlation between the preactivations and the dNTK, and in fact encapsulate the entire effect of the dNTK at order $1/n$. As we'll show next, any other dNTK–preactivation cross correlators, e.g., $\mathbb{E}\left[\widehat{\mathrm{d}H}^{(\ell)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2} z^{(\ell)}_{j_3;\delta_3} z^{(\ell)}_{j_4;\delta_4} z^{(\ell)}_{j_5;\delta_5}\right]$, can always be expressed in terms of a combination of $P^{(\ell)}$ and $Q^{(\ell)}$ at this order.

11.2.3 Deeper Layers: Growing dNTK

As before with §4.1 ‖ §8.1 ‖ §11.2.1 and §4.2 ‖ §8.2 ‖ §11.2.2, this section parallels our other sections analyzing the RG flow in deeper layers (§4.3 ‖ §8.3).

To proceed forward, we'll first need to evaluate an interlayer formula with three weight insertions (extending our work in §8.3.0), and then we'll immediately put it to use in order to obtain recursions for the dNTK–preactivation cross correlation.

*Inter*lude 2: *Inter*layer Correlations Reloaded

Since the forward equation for the dNTK (11.24) has terms with one or three $(\ell+1)$-th-layer weight matrices, we'll need interlayer formulae with one or three weight insertions.

[3] The cross-correlation tensor $P^{(2)}_{\delta_0\delta_1\delta_2\delta_3}$ (11.31) – and more generally $P^{(\ell)}_{\delta_0\delta_1\delta_2\delta_3}$ in deeper layers – is only symmetric under the exchange of its middle sample indices $\delta_1 \leftrightarrow \delta_2$. Meanwhile, it's manifestly clear from (11.32) that the other cross-correlation tensor $Q^{(2)}_{\delta_0\delta_1\delta_2\delta_3}$ has no symmetry whatsoever.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

Let's start by recalling our generating function for interlayer correlations (8.53):

$$\mathbb{E}\left[\mathcal{O}\left(z^{(\ell+1)}\right) e^{\sum_{i,j}\mathcal{J}_{ij}W_{ij}^{(\ell+1)}}\mathcal{Q}\left(z^{(\ell)},\widehat{H}^{(\ell)},\widehat{\mathrm{d}H}^{(\ell)}\right)\right] \tag{11.33}$$
$$= \exp\left(\frac{C_W^{(\ell+1)}}{2n_\ell}\sum_{i,j}\mathcal{J}_{ij}^2\right)\mathbb{E}\left[\left\langle\!\!\left\langle \mathcal{O}\left(z_{i;\delta}^{(\ell+1)}+\frac{C_W^{(\ell+1)}}{n_\ell}\sum_{j=1}^{n_\ell}\mathcal{J}_{ij}\sigma_{j;\delta}^{(\ell)}\right)\right\rangle\!\!\right\rangle_{\widehat{G}^{(\ell+1)}}\mathcal{Q}\left(z^{(\ell)},\widehat{H}^{(\ell)},\widehat{\mathrm{d}H}^{(\ell)}\right)\right].$$

In this formula, $\mathcal{O}$ is a generic function of $(\ell+1)$-th-layer preactivations only, and $\mathcal{Q}$ is a function of any of our ℓ-th-layer objects; in particular, since the original derivation of this formula didn't depend on specifics of $\mathcal{Q}$, we've also included the ℓ-th-layer dNTK as part of its argument.

With that in mind, we first wrote down an interlayer formula with one insertion as (10.10) when analyzing (the lack of) representation learning in the infinite-width limit. To save you the need to flip back and refresh your memory, we'll reprint it here after making some minor notational adjustments:

$$\mathbb{E}\left[\mathcal{O}\left(z^{(\ell+1)}\right) W_{ij}^{(\ell+1)}\mathcal{Q}\left(z^{(\ell)},\widehat{H}^{(\ell)},\widehat{\mathrm{d}H}^{(\ell)}\right)\right] \tag{11.34}$$
$$= \frac{C_W^{(\ell+1)}}{n_\ell}\sum_{\delta\in\mathcal{D}}\mathbb{E}\left[\left\langle\!\!\left\langle\frac{\partial\mathcal{O}}{\partial z_{i;\delta}^{(\ell+1)}}\right\rangle\!\!\right\rangle_{\widehat{G}^{(\ell+1)}}\sigma_{j;\delta}^{(\ell)}\,\mathcal{Q}\left(z^{(\ell)},\widehat{H}^{(\ell)},\widehat{\mathrm{d}H}^{(\ell)}\right)\right].$$

This can be readily rederived by differentiating the generating function (11.33) with respect to the source as $\frac{d}{d\mathcal{J}_{ij}}$ once and then setting the source to zero.

In contrast, the three-weight insertion formula will be new. Thrice-differentiating the generating function (11.33) with respect to the source as $\frac{d}{d\mathcal{J}_{i_0j_0}}\frac{d}{d\mathcal{J}_{i_1j_1}}\frac{d}{d\mathcal{J}_{i_2j_2}}$ and then setting the source to zero $\mathcal{J}=0$, we find

$$\mathbb{E}\left[\mathcal{O}\left(z^{(\ell+1)}\right) W_{i_0j_0}^{(\ell+1)}W_{i_1j_1}^{(\ell+1)}W_{i_2j_2}^{(\ell+1)}\mathcal{Q}\left(z^{(\ell)},\widehat{H}^{(\ell)},\widehat{\mathrm{d}H}^{(\ell)}\right)\right] \tag{11.35}$$
$$= \left(\frac{C_W^{(\ell+1)}}{n_\ell}\right)^2\delta_{i_0i_1}\delta_{j_0j_1}\sum_{\delta\in\mathcal{D}}\mathbb{E}\left[\left\langle\!\!\left\langle\frac{\partial\mathcal{O}}{\partial z_{i_2;\delta}^{(\ell+1)}}\right\rangle\!\!\right\rangle_{\widehat{G}^{(\ell+1)}}\sigma_{j_2;\delta}^{(\ell)}\,\mathcal{Q}\left(z^{(\ell)},\widehat{H}^{(\ell)},\widehat{\mathrm{d}H}^{(\ell)}\right)\right]$$
$$+\left(\frac{C_W^{(\ell+1)}}{n_\ell}\right)^2\delta_{i_0i_2}\delta_{j_0j_2}\sum_{\delta\in\mathcal{D}}\mathbb{E}\left[\left\langle\!\!\left\langle\frac{\partial\mathcal{O}}{\partial z_{i_1;\delta}^{(\ell+1)}}\right\rangle\!\!\right\rangle_{\widehat{G}^{(\ell+1)}}\sigma_{j_1;\delta}^{(\ell)}\,\mathcal{Q}\left(z^{(\ell)},\widehat{H}^{(\ell)},\widehat{\mathrm{d}H}^{(\ell)}\right)\right]$$
$$+\left(\frac{C_W^{(\ell+1)}}{n_\ell}\right)^2\delta_{i_1i_2}\delta_{j_1j_2}\sum_{\delta\in\mathcal{D}}\mathbb{E}\left[\left\langle\!\!\left\langle\frac{\partial\mathcal{O}}{\partial z_{i_0;\delta}^{(\ell+1)}}\right\rangle\!\!\right\rangle_{\widehat{G}^{(\ell+1)}}\sigma_{j_0;\delta}^{(\ell)}\,\mathcal{Q}\left(z^{(\ell)},\widehat{H}^{(\ell)},\widehat{\mathrm{d}H}^{(\ell)}\right)\right]$$
$$+\left(\frac{C_W^{(\ell+1)}}{n_\ell}\right)^3\sum_{\delta_0,\delta_1,\delta_2\in\mathcal{D}}$$
$$\mathbb{E}\left[\left\langle\!\!\left\langle\frac{\partial^3\mathcal{O}}{\partial z_{i_0;\delta_0}^{(\ell+1)}\partial z_{i_1;\delta_1}^{(\ell+1)}\partial z_{i_2;\delta_2}^{(\ell+1)}}\right\rangle\!\!\right\rangle_{\widehat{G}^{(\ell+1)}}\sigma_{j_0;\delta_0}^{(\ell)}\sigma_{j_1;\delta_1}^{(\ell)}\sigma_{j_2;\delta_2}^{(\ell)}\,\mathcal{Q}\left(z^{(\ell)},\widehat{H}^{(\ell)},\widehat{\mathrm{d}H}^{(\ell)}\right)\right].$$

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

Intuitively, we can understand this formula as follows: each of the first three terms comes from forming one Wick contraction with two of the weight insertions and then forming another contraction between the remaining weight and a weight hidden inside the $z^{(\ell+1)}$ in $\mathcal{O}$, while the final term comes from all three weight insertions each forming a contraction with other weights inside the observable $\mathcal{O}$.

dNTK–Preactivation Cross Correlations

Let's first use the interlayer formulae derived above to show that all of the dNTK's contributions to the statistics of the joint distribution $p\Big(z^{(\ell)}, \widehat{H}^{(\ell)}, \widehat{\mathrm{d}H}^{(\ell)}\Big|\mathcal{D}\Big)$ at order $1/n$ are captured by the cross correlation of the dNTK with a single preactivation, $\mathbb{E}\left[\widehat{\mathrm{d}H}^{(\ell)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2} z^{(\ell+1)}_{i_3;\delta_3}\right]$. To that end, we'll examine a dNTK–preactivation cross correlator of a very general form:

$$
\begin{aligned}
&\mathbb{E}\left[\mathcal{O}\Big(z^{(\ell+1)}\Big)\, \widehat{\mathrm{d}H}^{(\ell+1)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2}\right] \qquad (11.36)\\
&\quad = \delta_{i_0 i_1}\frac{\lambda_W^{(\ell+1)}}{n_\ell}\sum_{k_0,k_2}\mathbb{E}\left[\mathcal{O}\Big(z^{(\ell+1)}\Big) W^{(\ell+1)}_{i_2k_2}\sigma'^{(\ell)}_{k_0;\delta_0}\sigma^{(\ell)}_{k_0;\delta_1}\sigma'^{(\ell)}_{k_2;\delta_2}\widehat{H}^{(\ell)}_{k_0k_2;\delta_0\delta_2}\right]\\
&\qquad + \delta_{i_0 i_2}\frac{\lambda_W^{(\ell+1)}}{n_\ell}\sum_{k_0,k_1}\mathbb{E}\left[\mathcal{O}\Big(z^{(\ell+1)}\Big) W^{(\ell+1)}_{i_1k_1}\sigma'^{(\ell)}_{k_0;\delta_0}\sigma'^{(\ell)}_{k_1;\delta_1}\sigma^{(\ell)}_{k_0;\delta_2}\widehat{H}^{(\ell)}_{k_0k_1;\delta_0\delta_1}\right]\\
&\qquad + \sum_{k_0,k_1,k_2}\mathbb{E}\left[\mathcal{O}\Big(z^{(\ell+1)}\Big) W^{(\ell+1)}_{i_0k_0}W^{(\ell+1)}_{i_1k_1}W^{(\ell+1)}_{i_2k_2}\sigma''^{(\ell)}_{k_0;\delta_0}\sigma'^{(\ell)}_{k_1;\delta_1}\sigma'^{(\ell)}_{k_2;\delta_2}\widehat{H}^{(\ell)}_{k_0k_1;\delta_0\delta_1}\widehat{H}^{(\ell)}_{k_0k_2;\delta_0\delta_2}\right]\\
&\qquad + \sum_{k_0,k_1,k_2}\mathbb{E}\left[\mathcal{O}\Big(z^{(\ell+1)}\Big) W^{(\ell+1)}_{i_0k_0}W^{(\ell+1)}_{i_1k_1}W^{(\ell+1)}_{i_2k_2}\sigma'^{(\ell)}_{k_0;\delta_0}\sigma'^{(\ell)}_{k_1;\delta_1}\sigma'^{(\ell)}_{k_2;\delta_2}\widehat{\mathrm{d}H}^{(\ell)}_{k_0k_1k_2;\delta_0\delta_1\delta_2}\right].
\end{aligned}
$$

Here, we took the expectation of the dNTK forward equation (11.24) multiplied by a generic observable $\mathcal{O}\Big(z^{(\ell+1)}\Big)$ of $(\ell+1)$-th-layer preactivations. We also ordered the four terms to reflect the order in which we will subsequently evaluate them.

First, let's simplify the first two terms. Using our interlayer formula with one weight insertion (11.34) on the first term in (11.36), we get

$$
\begin{aligned}
&\frac{\lambda_W^{(\ell+1)}}{n_\ell}\delta_{i_0 i_1}\sum_{k_0,k_2}\mathbb{E}\left[\mathcal{O}\Big(z^{(\ell+1)}\Big) W^{(\ell+1)}_{i_2k_2}\sigma'^{(\ell)}_{k_0;\delta_0}\sigma^{(\ell)}_{k_0;\delta_1}\sigma'^{(\ell)}_{k_2;\delta_2}\widehat{H}^{(\ell)}_{k_0k_2;\delta_0\delta_2}\right] \qquad (11.37)\\
&\quad = \frac{\lambda_W^{(\ell+1)}C_W^{(\ell+1)}}{n_\ell^2}\sum_{k_0,k_2}\sum_{\delta\in\mathcal{D}}\delta_{i_0 i_1}\mathbb{E}\left[\left\langle\!\!\left\langle\frac{\partial\mathcal{O}}{\partial z^{(\ell+1)}_{i_2;\delta}}\right\rangle\!\!\right\rangle_{\widehat{G}^{(\ell+1)}}\sigma^{(\ell)}_{k_2;\delta}\sigma'^{(\ell)}_{k_0;\delta_0}\sigma^{(\ell)}_{k_0;\delta_1}\sigma'^{(\ell)}_{k_2;\delta_2}\widehat{H}^{(\ell)}_{k_0k_2;\delta_0\delta_2}\right]\\
&\quad = \frac{\lambda_W^{(\ell+1)}C_W^{(\ell+1)}}{n_\ell^2}\sum_{k,m}\sum_{\delta\in\mathcal{D}}\delta_{i_0 i_1}\left\langle\!\!\left\langle\frac{\partial\mathcal{O}}{\partial z^{(\ell+1)}_{i_2;\delta}}\right\rangle\!\!\right\rangle_{G^{(\ell+1)}}\mathbb{E}\left[\sigma^{(\ell)}_{k;\delta_1}\sigma^{(\ell)}_{m;\delta}\sigma'^{(\ell)}_{k;\delta_0}\sigma'^{(\ell)}_{m;\delta_2}\widehat{H}^{(\ell)}_{km;\delta_0\delta_2}\right]+O\Big(\frac{1}{n^2}\Big)\\
&\quad = \frac{1}{n_\ell}\frac{\lambda_W^{(\ell+1)}}{C_W^{(\ell+1)}}\sum_{\delta\in\mathcal{D}}\delta_{i_0 i_1}\left\langle\!\!\left\langle\frac{\partial\mathcal{O}}{\partial z^{(\ell+1)}_{i_2;\delta}}\right\rangle\!\!\right\rangle_{G^{(\ell+1)}}F^{(\ell+1)}_{\delta_1\delta_0\delta_3\delta_2}+O\Big(\frac{1}{n^2}\Big),
\end{aligned}
$$

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

where in the third line we used the Schwinger–Dyson formula (8.70) to expand the metric fluctuation around its mean $G^{(\ell+1)}$, noting that the second term with the fluctuation is subleading, and in the last line we identified the remaining expectation as the definition of the NTK–preactivation cross correlation tensor $F^{(\ell+1)}$ from (8.69) up to constants.[4]

Similarly, for the second term in (11.36) we get an identical contribution up to the swapping of neural-sample index pairs as $(i_1, \delta_1) \leftrightarrow (i_2, \delta_2)$.

Next, let's tackle the third term in (11.36). Applying our interlayer formula with three weight insertions (11.35), we get

$$\begin{aligned}
&\sum_{k_0,k_1,k_2} \mathbb{E}\left[\mathcal{O}\left(z^{(\ell+1)}\right) W^{(\ell+1)}_{i_0k_0} W^{(\ell+1)}_{i_1k_1} W^{(\ell+1)}_{i_2k_2} \sigma''^{(\ell)}_{k_0;\delta_0} \sigma'^{(\ell)}_{k_1;\delta_1} \sigma'^{(\ell)}_{k_2;\delta_2} \widehat{H}^{(\ell)}_{k_0k_1;\delta_0\delta_1} \widehat{H}^{(\ell)}_{k_0k_2;\delta_0\delta_2}\right] \\
&= \frac{\left(C_W^{(\ell+1)}\right)^2}{n_\ell} \delta_{i_0i_1} \sum_{\delta\in\mathcal{D}} \left\langle\!\!\left\langle \frac{\partial \mathcal{O}}{\partial z^{(\ell+1)}_{i_2;\delta}} \right\rangle\!\!\right\rangle_{G^{(\ell+1)}} \left\{ \frac{1}{n_\ell} \sum_{k,m} \mathbb{E}\left[\sigma^{(\ell)}_{m;\delta} \sigma''^{(\ell)}_{k;\delta_0} \sigma'^{(\ell)}_{k;\delta_1} \sigma'^{(\ell)}_{m;\delta_2} \widehat{H}^{(\ell)}_{kk;\delta_0\delta_1} \widehat{H}^{(\ell)}_{km;\delta_0\delta_2}\right] \right\} \\
&\quad + \frac{\left(C_W^{(\ell+1)}\right)^2}{n_\ell} \delta_{i_0i_2} \sum_{\delta\in\mathcal{D}} \left\langle\!\!\left\langle \frac{\partial \mathcal{O}}{\partial z^{(\ell+1)}_{i_1;\delta}} \right\rangle\!\!\right\rangle_{G^{(\ell+1)}} \left\{ \frac{1}{n_\ell} \sum_{k,m} \mathbb{E}\left[\sigma^{(\ell)}_{m;\delta} \sigma''^{(\ell)}_{k;\delta_0} \sigma'^{(\ell)}_{k;\delta_2} \sigma'^{(\ell)}_{m;\delta_1} \widehat{H}^{(\ell)}_{kk;\delta_0\delta_2} \widehat{H}^{(\ell)}_{km;\delta_0\delta_1}\right] \right\} \\
&\quad + \frac{\left(C_W^{(\ell+1)}\right)^2}{n_\ell} \delta_{i_1i_2} \sum_{\delta\in\mathcal{D}} \left\langle\!\!\left\langle \frac{\partial \mathcal{O}}{\partial z^{(\ell+1)}_{i_0;\delta}} \right\rangle\!\!\right\rangle_{G^{(\ell+1)}} \left\{ \frac{1}{n_\ell} \sum_{k,m} \mathbb{E}\left[\sigma^{(\ell)}_{m;\delta} \sigma''^{(\ell)}_{m;\delta_0} \sigma'^{(\ell)}_{k;\delta_1} \sigma'^{(\ell)}_{k;\delta_2} \widehat{H}^{(\ell)}_{mk;\delta_0\delta_1} \widehat{H}^{(\ell)}_{mk;\delta_0\delta_2}\right] \right\} \\
&\quad + O\left(\frac{1}{n^2}\right),
\end{aligned} \tag{11.38}$$

where for each term we again used the Schwinger–Dyson formula (8.70) to expand the metric fluctuation around its mean $G^{(\ell+1)}$, picking up the leading contribution from the mean metric, and then took the Gaussian expectation outside the full ℓ-th-layer expectation. Note that the final would-be term proportional to $\partial^3\mathcal{O}$ is subleading:

$$\begin{aligned}
&\left(C_W^{(\ell+1)}\right)^3 \sum_{\delta,\delta_4,\delta_5\in\mathcal{D}} \left\langle\!\!\left\langle \frac{\partial^3 \mathcal{O}}{\partial z^{(\ell+1)}_{i_0;\delta} \partial z^{(\ell+1)}_{i_1;\delta_4} \partial z^{(\ell+1)}_{i_2;\delta_5}} \right\rangle\!\!\right\rangle_{G^{(\ell+1)}} \\
&\quad \times \frac{1}{n_\ell^3} \sum_{k_0,k_1,k_2} \mathbb{E}\left[\sigma^{(\ell)}_{k_0;\delta} \sigma^{(\ell)}_{k_1;\delta_4} \sigma^{(\ell)}_{k_2;\delta_5} \sigma''^{(\ell)}_{k_0;\delta_0} \sigma'^{(\ell)}_{k_1;\delta_1} \sigma'^{(\ell)}_{k_2;\delta_2} \widehat{H}^{(\ell)}_{k_0k_1;\delta_0\delta_1} \widehat{H}^{(\ell)}_{k_0k_2;\delta_0\delta_2}\right] = O\left(\frac{1}{n^2}\right).
\end{aligned} \tag{11.39}$$

To see why, decompose each stochastic NTK into a mean and fluctuation as $\widehat{H}^{(\ell)}_{i_1i_2;\delta_1\delta_2} = \delta_{i_1i_2} H^{(\ell)}_{\delta_1\delta_2} + \widehat{\Delta H}^{(\ell)}_{i_1i_2;\delta_1\delta_2}$ (8.58), and evaluate the resulting four terms. In each case, you'll find the terms are $O(1/n^2)$ suppressed due to the Kronecker deltas constraining the triple sum and/or the additional $1/n$ suppression coming from the fluctuations.

[4] Note that this is an $(\ell+1)$-th-layer quantity, rather than an ℓ-th-layer quantity as we typically have on the right-hand side of recursions. If you prefer, you can use the F-recursion (8.79) to re-express it in terms of a more complicated collection of ℓ-th-layer quantities.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

Lastly, let's process the fourth and final term in our general cross correlation (11.36). Again applying our interlayer formula with three weight insertions (11.35) and taking only the leading-order pieces, we get

$$\begin{aligned}
&\sum_{k_0,k_1,k_2} \mathbb{E}\left[\mathcal{O}\left(z^{(\ell+1)}\right) W^{(\ell+1)}_{i_0k_0} W^{(\ell+1)}_{i_1k_1} W^{(\ell+1)}_{i_2k_2} \sigma'^{(\ell)}_{k_0;\delta_0} \sigma'^{(\ell)}_{k_1;\delta_1} \sigma'^{(\ell)}_{k_2;\delta_2} \widehat{\mathrm{d}H}^{(\ell)}_{k_0k_1k_2;\delta_0\delta_1\delta_2}\right] \\
&= \frac{\left(C_W^{(\ell+1)}\right)^2}{n_\ell} \sum_{\delta\in\mathcal{D}} \delta_{i_0i_1} \left\langle\!\!\left\langle \frac{\partial\mathcal{O}}{\partial z^{(\ell+1)}_{i_2;\delta}} \right\rangle\!\!\right\rangle_{G^{(\ell+1)}} \left\{ \frac{1}{n_\ell} \sum_{k,m} \mathbb{E}\left[\sigma'^{(\ell)}_{k;\delta_0} \sigma'^{(\ell)}_{k;\delta_1} \sigma'^{(\ell)}_{m;\delta_2} \sigma^{(\ell)}_{m;\delta} \widehat{\mathrm{d}H}^{(\ell)}_{kkm;\delta_0\delta_1\delta_2}\right] \right\} \\
&\quad + \frac{\left(C_W^{(\ell+1)}\right)^2}{n_\ell} \sum_{\delta\in\mathcal{D}} \delta_{i_0i_2} \left\langle\!\!\left\langle \frac{\partial\mathcal{O}}{\partial z^{(\ell+1)}_{i_1;\delta}} \right\rangle\!\!\right\rangle_{G^{(\ell+1)}} \left\{ \frac{1}{n_\ell} \sum_{k,m} \mathbb{E}\left[\sigma'^{(\ell)}_{k;\delta_0} \sigma'^{(\ell)}_{m;\delta_1} \sigma'^{(\ell)}_{k;\delta_2} \sigma^{(\ell)}_{m;\delta} \widehat{\mathrm{d}H}^{(\ell)}_{kmk;\delta_0\delta_1\delta_2}\right] \right\} \\
&\quad + \frac{\left(C_W^{(\ell+1)}\right)^2}{n_\ell} \sum_{\delta\in\mathcal{D}} \delta_{i_1i_2} \left\langle\!\!\left\langle \frac{\partial\mathcal{O}}{\partial z^{(\ell+1)}_{i_0;\delta}} \right\rangle\!\!\right\rangle_{G^{(\ell+1)}} \left\{ \frac{1}{n_\ell} \sum_{k,m} \mathbb{E}\left[\sigma'^{(\ell)}_{m;\delta_0} \sigma'^{(\ell)}_{k;\delta_1} \sigma'^{(\ell)}_{k;\delta_2} \sigma^{(\ell)}_{m;\delta} \widehat{\mathrm{d}H}^{(\ell)}_{mkk;\delta_0\delta_1\delta_2}\right] \right\} \\
&\quad + O\left(\frac{1}{n^2}\right) .
\end{aligned} \tag{11.40}$$

Note that the factors in the curly brackets are actually of order one since – as we saw in the second layer and will recursively show next for general layers ℓ – the leading dNTK–preactivation cross correlation is of order $1/n$. Meanwhile, again the final would-be term proportional to $\partial^3\mathcal{O}$ is subleading:

$$\begin{aligned}
&\left(C_W^{(\ell+1)}\right)^3 \sum_{\delta_3,\delta_4,\delta_5\in\mathcal{D}} \left\langle\!\!\left\langle \frac{\partial^3\mathcal{O}}{\partial z^{(\ell+1)}_{i_0;\delta_3} \partial z^{(\ell+1)}_{i_1;\delta_4} \partial z^{(\ell+1)}_{i_2;\delta_5}} \right\rangle\!\!\right\rangle_{K^{(\ell+1)}} \\
&\quad \times \frac{1}{n_\ell^3} \sum_{k_0,k_1,k_2} \mathbb{E}\left[\left(\sigma'^{(\ell)}_{k_0;\delta_0} \sigma^{(\ell)}_{k_0;\delta_3}\right) \left(\sigma'^{(\ell)}_{k_1;\delta_1} \sigma^{(\ell)}_{k_1;\delta_4}\right) \left(\sigma'^{(\ell)}_{k_2;\delta_2} \sigma^{(\ell)}_{k_2;\delta_5}\right) \widehat{\mathrm{d}H}^{(\ell)}_{k_0k_1k_2;\delta_0\delta_1\delta_2}\right] = O\left(\frac{1}{n^2}\right) .
\end{aligned} \tag{11.41}$$

To see why, note that the expectation is another dNTK–preactivation cross correlator and thus is at most of order $1/n$. Further, we only get such an order-$1/n$ contribution when two out of three neural indices k_0, k_1, k_2 coincide: this should be clear from the pattern of Kronecker deltas that arise when we evaluate such dNTK–preactivation cross correlations in terms of our P-Q decomposition; cf. (11.30) for the second layer, or look ahead a paragraph to (11.42) for deeper layers. This means that the sum over all three neural indices will be restricted to be only two independent sums, and, taking the prefactor of $1/n^3$ into account, the overall contribution of this term will thus go as $\sim (1/n^3)(n^2)(1/n) \sim 1/n^2$.

Substituting all our evaluated contributions (11.37) (twice), (11.38), and (11.40) into our expression for a general dNTK–preactivation cross correlator (11.36), we get

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

$$\mathbb{E}\left[\mathcal{O}\left(z^{(\ell+1)}\right)\,\widehat{\mathrm{d}H}^{(\ell+1)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2}\right] \tag{11.42}$$
$$= \frac{1}{n_\ell}\sum_{\delta\in\mathcal{D}}\left[\delta_{i_1 i_2}\left\langle\!\!\left\langle \frac{\partial\mathcal{O}}{\partial z^{(\ell+1)}_{i_0;\delta}}\right\rangle\!\!\right\rangle_{G^{(\ell+1)}} P^{(\ell+1)}_{\delta_0\delta_1\delta_2\delta} + \delta_{i_0 i_1}\left\langle\!\!\left\langle \frac{\partial\mathcal{O}}{\partial z^{(\ell+1)}_{i_2;\delta}}\right\rangle\!\!\right\rangle_{G^{(\ell+1)}} Q^{(\ell+1)}_{\delta_0\delta_1\delta_2\delta} + \delta_{i_0 i_2}\left\langle\!\!\left\langle \frac{\partial\mathcal{O}}{\partial z^{(\ell+1)}_{i_1;\delta}}\right\rangle\!\!\right\rangle_{G^{(\ell+1)}} Q^{(\ell+1)}_{\delta_0\delta_2\delta_1\delta}\right] + O\left(\frac{1}{n^2}\right),$$

where we've introduced the $(\ell+1)$-th-layer generalizations of the second-layer tensors $P^{(2)}$ (11.31) and $Q^{(2)}$ (11.32):

$$P^{(\ell+1)}_{\delta_0\delta_1\delta_2\delta_3} \equiv \left(C_W^{(\ell+1)}\right)^2 \frac{1}{n_\ell}\sum_{k,m=1}^{n_\ell}\mathbb{E}\left[\sigma''^{(\ell)}_{m;\delta_0}\sigma'^{(\ell)}_{k;\delta_1}\sigma'^{(\ell)}_{k;\delta_2}\sigma^{(\ell)}_{m;\delta_3}\widehat{H}^{(\ell)}_{mk;\delta_0\delta_1}\widehat{H}^{(\ell)}_{mk;\delta_0\delta_2}\right] \tag{11.43}$$
$$+ \left(C_W^{(\ell+1)}\right)^2 \frac{1}{n_\ell}\sum_{k,m=1}^{n_\ell}\mathbb{E}\left[\sigma'^{(\ell)}_{m;\delta_0}\sigma'^{(\ell)}_{k;\delta_1}\sigma'^{(\ell)}_{k;\delta_2}\sigma^{(\ell)}_{m;\delta_3}\widehat{\mathrm{d}H}^{(\ell)}_{mkk;\delta_0\delta_1\delta_2}\right] + O\left(\frac{1}{n}\right),$$

$$Q^{(\ell+1)}_{\delta_0\delta_1\delta_2\delta_3} \equiv \left(C_W^{(\ell+1)}\right)^2 \frac{1}{n_\ell}\sum_{k,m=1}^{n_\ell}\mathbb{E}\left[\sigma''^{(\ell)}_{k;\delta_0}\sigma'^{(\ell)}_{k;\delta_1}\sigma'^{(\ell)}_{m;\delta_2}\sigma^{(\ell)}_{m;\delta_3}\widehat{H}^{(\ell)}_{kk;\delta_0\delta_1}\widehat{H}^{(\ell)}_{km;\delta_0\delta_2}\right] + \frac{\lambda_W^{(\ell+1)}}{C_W^{(\ell+1)}}F^{(\ell+1)}_{\delta_1\delta_0\delta_3\delta_2}$$
$$+ \left(C_W^{(\ell+1)}\right)^2 \frac{1}{n_\ell}\sum_{k,m=1}^{n_\ell}\mathbb{E}\left[\sigma'^{(\ell)}_{k;\delta_0}\sigma'^{(\ell)}_{k;\delta_1}\sigma'^{(\ell)}_{m;\delta_2}\sigma^{(\ell)}_{m;\delta_3}\widehat{\mathrm{d}H}^{(\ell)}_{kkm;\delta_0\delta_1\delta_2}\right] + O\left(\frac{1}{n}\right). \tag{11.44}$$

To see how these general expressions reduce to the ones we had for $P^{(2)}_{\delta_0\delta_1\delta_2\delta_3}$ and $Q^{(2)}_{\delta_0\delta_1\delta_2\delta_3}$ in the second layer, (11.31) and (11.32), recall that the first-layer NTK is deterministic and diagonal in neural indices $\widehat{H}^{(1)}_{i_1 i_2;\delta_1\delta_2} = \delta_{i_1 i_2}H^{(1)}_{\delta_1\delta_2}$, that the first-layer dNTK vanishes $\widehat{\mathrm{d}H}^{(1)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2} = 0$, and that in the second layer we had for the NTK–preactivation cross correlation tensor $F^{(2)}_{\delta_1\delta_0\delta_3\delta_2} = \left(C_W^{(2)}\right)^2 H^{(1)}_{\delta_0\delta_2}\left\langle\sigma_{\delta_1}\sigma_{\delta_3}\sigma'_{\delta_0}\sigma'_{\delta_2}\right\rangle_{G^{(1)}}$ (8.39).

Finally, note that by setting our observable to $\mathcal{O} = z^{(\ell+1)}_{i_3;\delta_3}$, we get

$$\mathbb{E}\left[\widehat{\mathrm{d}H}^{(\ell+1)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2} z^{(\ell+1)}_{i_3;\delta_3}\right] = \frac{1}{n_\ell}\left[\delta_{i_0 i_3}\delta_{i_1 i_2}P^{(\ell+1)}_{\delta_0\delta_1\delta_2\delta_3} + \delta_{i_0 i_1}\delta_{i_2 i_3}Q^{(\ell+1)}_{\delta_0\delta_1\delta_2\delta_3} + \delta_{i_0 i_2}\delta_{i_1 i_3}Q^{(\ell+1)}_{\delta_0\delta_2\delta_1\delta_3}\right]. \tag{11.45}$$

Importantly, this means that at leading nonvanishing order in $1/n$, the tensors in the decomposition of our elementary dNTK–preactivation cross correlator with a single preactivation (11.45) completely fix the general dNTK–preactivation cross correlation with more complicated observables (11.42).[5] Thus, to completely incorporate the

[5] In other words, at our order in $1/n$ we can always replace the ℓ-th-layer dNTK inside any expectations by the following differential operator:

$$\widehat{\mathrm{d}H}^{(\ell)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2} \to \frac{1}{n_{\ell-1}}\sum_{\delta\in\mathcal{D}}\left[\delta_{i_1 i_2}P^{(\ell)}_{\delta_0\delta_1\delta_2\delta}\frac{\partial}{\partial z^{(\ell)}_{i_0;\delta}} + \delta_{i_0 i_1}Q^{(\ell)}_{\delta_0\delta_1\delta_2\delta}\frac{\partial}{\partial z^{(\ell)}_{i_2;\delta}} + \delta_{i_0 i_2}Q^{(\ell)}_{\delta_0\delta_2\delta_1\delta}\frac{\partial}{\partial z^{(\ell)}_{i_1;\delta}}\right], \tag{11.46}$$

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

leading effects of the dNTK in our analysis, we only need to evaluate recursions for $P^{(\ell)}$ and $Q^{(\ell)}$.[6]

P-recursion

To find a recursion for $P^{(\ell)}$, we need to evaluate the two expectations in (11.43).

For the first expectation, making a decomposition of the NTK into a mean and fluctuation as $\widehat{H}^{(\ell)}_{i_1 i_2;\delta_1\delta_2} = \delta_{i_1 i_2} H^{(\ell)}_{\delta_1\delta_2} + \widehat{\Delta H}^{(\ell)}_{i_1 i_2;\delta_1\delta_2}$, we get

$$\begin{aligned}
&\frac{1}{n_\ell^2}\sum_{k,m}\mathbb{E}\left[\sigma''^{(\ell)}_{m;\delta_0}\sigma'^{(\ell)}_{k;\delta_1}\sigma'^{(\ell)}_{k;\delta_2}\sigma^{(\ell)}_{m;\delta_3}\widehat{H}^{(\ell)}_{mk;\delta_0\delta_1}\widehat{H}^{(\ell)}_{mk;\delta_0\delta_2}\right] \qquad (11.47)\\
&= \frac{1}{n_\ell}\left\langle \sigma''_{\delta_0}\sigma'_{\delta_1}\sigma'_{\delta_2}\sigma_{\delta_3}\right\rangle_{G^{(\ell)}} H^{(\ell)}_{\delta_0\delta_1}H^{(\ell)}_{\delta_0\delta_2} + \frac{1}{n_{\ell-1}}\left\langle \sigma''_{\delta_0}\sigma_{\delta_3}\right\rangle_{G^{(\ell)}}\left\langle \sigma'_{\delta_1}\sigma'_{\delta_2}\right\rangle_{G^{(\ell)}} B^{(\ell)}_{\delta_0\delta_0\delta_1\delta_2} + O\left(\frac{1}{n^2}\right).
\end{aligned}$$

In particular, the cross terms consisting of an NTK mean and an NTK fluctuation dropped out because the Kronecker delta from the mean constrained the double sum, and the fluctuation gave an additional $1/n$ suppression. If this explanation was a little too fast, you should review our slower derivation of the B-recursion from (8.83) to (8.89), which is identical in form to (11.47) above up to where the sample indices and the derivatives go.

For the second expectation in (11.43), we can just apply our general cross-correlation formula (11.42) for layer ℓ, letting us simplify it as

$$\begin{aligned}
&\frac{1}{n_\ell}\sum_{k,m=1}^{n_\ell}\mathbb{E}\left[\sigma'^{(\ell)}_{m;\delta_0}\sigma'^{(\ell)}_{k;\delta_1}\sigma'^{(\ell)}_{k;\delta_2}\sigma^{(\ell)}_{m;\delta_3}\widehat{\mathrm{d}H}^{(\ell)}_{mkk;\delta_0\delta_1\delta_2}\right] \qquad (11.48)\\
&= \frac{1}{n_\ell n_{\ell-1}}\sum_{k,m=1}^{n_\ell}\sum_{\delta_4\in\mathcal{D}}\left[\left\langle\!\!\left\langle \frac{\partial}{\partial z^{(\ell)}_{m;\delta_4}}\left(\sigma'^{(\ell)}_{m;\delta_0}\sigma'^{(\ell)}_{k;\delta_1}\sigma'^{(\ell)}_{k;\delta_2}\sigma^{(\ell)}_{m;\delta_3}\right)\right\rangle\!\!\right\rangle_{G^{(\ell)}} P^{(\ell)}_{\delta_0\delta_1\delta_2\delta_4} + \delta_{mk}\times O(1)\right]\\
&\quad + O\left(\frac{1}{n}\right)\\
&= \left(\frac{n_\ell}{n_{\ell-1}}\right)\left[\left\langle \sigma''_{\delta_0}\sigma_{\delta_3}\right\rangle_{G^{(\ell)}}\left\langle \sigma'_{\delta_1}\sigma'_{\delta_2}\right\rangle_{G^{(\ell)}} P^{(\ell)}_{\delta_0\delta_1\delta_2\delta_0} + \left\langle \sigma'_{\delta_0}\sigma'_{\delta_3}\right\rangle_{K^{(\ell)}}\left\langle \sigma'_{\delta_1}\sigma'_{\delta_2}\right\rangle_{K^{(\ell)}} P^{(\ell)}_{\delta_0\delta_1\delta_2\delta_3}\right]\\
&\quad + O\left(\frac{1}{n}\right).
\end{aligned}$$

with nonfluctuating coefficients $P^{(\ell)}$ and $Q^{(\ell)}$. When we use this replacement, we must remember that the derivatives act on all of the ℓ-th-layer preactivations multiplying the dNTK; i.e., move the dNTK all the way to the left side of the expectation before making such a replacement.

[6] Any preactivation–NTK–dNTK cross correlators, such as $\mathbb{E}\left[\widehat{\Delta H}^{(\ell)}_{i_1 i_2;\delta_1\delta_2}\widehat{\mathrm{d}H}^{(\ell)}_{i_3 i_4 i_5;\delta_3\delta_4\delta_5} z^{(\ell)}_{i_6;\delta_6}\right]$, and any higher-order dNTK correlators, such as $\mathbb{E}\left[\widehat{\mathrm{d}H}^{(\ell)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2}\widehat{\mathrm{d}H}^{(\ell)}_{i_3 i_4 i_5;\delta_3\delta_4\delta_5}\right]$, are subleading. We won't show this explicitly, but you may find additional tranquility in working it out for yourself; both examples are relatively simple to work out for the second layer, $L = 2$.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

Here in the second line, we've simply written the $Q^{(\ell)}$ terms proportional to δ_{mk} as $O(1)$; the details of these terms do not matter because when we perform the double sum with the restriction $m = k$, they will be subleading. As for the term proportional to $P^{(\ell)}$, the diagonal contribution with $k = m$ is similarly subleading; the leading contribution comes from the $(n_\ell^2 - n_\ell)$ off-diagonal pieces with $k \neq m$, for which the derivative acts only on two activations out of the four, and then we can further use Gaussian factorization to write each term as a product of single-neuron Gaussian expectations.

Plugging these two simplified expectations back into our expression for $P^{(\ell+1)}$ (11.43), we get a recursion:

$$\begin{aligned} P^{(\ell+1)}_{\delta_0\delta_1\delta_2\delta_3} &= \left(C_W^{(\ell+1)}\right)^2 \left\langle \sigma''_{\delta_0}\sigma'_{\delta_1}\sigma'_{\delta_2}\sigma_{\delta_3}\right\rangle_{G^{(\ell)}} H^{(\ell)}_{\delta_0\delta_1} H^{(\ell)}_{\delta_0\delta_2} \\ &\quad + \left(\frac{n_\ell}{n_{\ell-1}}\right)\left(C_W^{(\ell+1)}\right)^2 \left\langle \sigma'_{\delta_1}\sigma'_{\delta_2}\right\rangle_{G^{(\ell)}} \\ &\quad \times \left[\left\langle \sigma''_{\delta_0}\sigma_{\delta_3}\right\rangle_{G^{(\ell)}} P^{(\ell)}_{\delta_0\delta_1\delta_2\delta_0} + \left\langle \sigma'_{\delta_0}\sigma'_{\delta_3}\right\rangle_{G^{(\ell)}} P^{(\ell)}_{\delta_0\delta_1\delta_2\delta_3} + \left\langle \sigma''_{\delta_0}\sigma_{\delta_3}\right\rangle_{G^{(\ell)}} B^{(\ell)}_{\delta_0\delta_0\delta_1\delta_2}\right] + O\left(\frac{1}{n}\right). \end{aligned} \tag{11.49}$$

Interestingly, we see that this dNTK tensor $P^{(\ell)}$ mixes with the NTK mean $H^{(\ell)}$ as well as the NTK-variance tensor $B^{(\ell)}$. Since $H^{(\ell)}$ and $B^{(\ell)}$ are of order one, and since $P^{(1)} = 0$, this recursion shows that $P^{(\ell)}$ will recursively stay of order one for all layers ℓ.

Q-recursion

To find a recursion for $Q^{(\ell)}$, we need to work out the two expectations in (11.44).

For the first expectation, again making a decomposition of the NTK into a mean and fluctuation as $\widehat{H}^{(\ell)}_{i_1 i_2;\delta_1\delta_2} = \delta_{i_1 i_2} H^{(\ell)}_{\delta_1\delta_2} + \widehat{\Delta H}^{(\ell)}_{i_1 i_2;\delta_1\delta_2}$, we get

$$\begin{aligned} &\frac{1}{n_\ell^2}\sum_{k,m} \mathbb{E}\left[\sigma''^{(\ell)}_{k;\delta_0}\sigma'^{(\ell)}_{k;\delta_1}\sigma'^{(\ell)}_{m;\delta_2}\sigma^{(\ell)}_{m;\delta_3}\widehat{H}^{(\ell)}_{kk;\delta_0\delta_1}\widehat{H}^{(\ell)}_{km;\delta_0\delta_2}\right] \\ &= \frac{1}{n_\ell^2}\sum_{k} \mathbb{E}\left[\sigma''^{(\ell)}_{k;\delta_0}\sigma'^{(\ell)}_{k;\delta_1}\sigma'^{(\ell)}_{k;\delta_2}\sigma^{(\ell)}_{k;\delta_3}\right] H^{(\ell)}_{\delta_0\delta_1} H^{(\ell)}_{\delta_0\delta_2} \\ &\quad + \frac{1}{n_\ell^2}\sum_{k} \mathbb{E}\left[\sigma^{(\ell)}_{k;\delta_3}\sigma''^{(\ell)}_{k;\delta_0}\sigma'^{(\ell)}_{k;\delta_1}\sigma'^{(\ell)}_{k;\delta_2}\widehat{\Delta H}^{(\ell)}_{kk;\delta_0\delta_1}\right] H^{(\ell)}_{\delta_0\delta_2} \\ &\quad + \frac{1}{n_\ell^2}\sum_{k,m} \mathbb{E}\left[\sigma''^{(\ell)}_{k;\delta_0}\sigma'^{(\ell)}_{k;\delta_1}\sigma'^{(\ell)}_{m;\delta_2}\sigma^{(\ell)}_{m;\delta_3}\widehat{\Delta H}^{(\ell)}_{km;\delta_0\delta_2}\right] H^{(\ell)}_{\delta_0\delta_1} \\ &\quad + \frac{1}{n_\ell^2}\sum_{k,m} \mathbb{E}\left[\sigma''^{(\ell)}_{k;\delta_0}\sigma'^{(\ell)}_{k;\delta_1}\sigma'^{(\ell)}_{m;\delta_2}\sigma^{(\ell)}_{m;\delta_3}\widehat{\Delta H}^{(\ell)}_{kk;\delta_0\delta_1}\widehat{\Delta H}^{(\ell)}_{km;\delta_0\delta_2}\right] \\ &= \frac{1}{n_\ell}\left\langle \sigma''_{\delta_0}\sigma'_{\delta_1}\sigma'_{\delta_2}\sigma_{\delta_3}\right\rangle_{G^{(\ell)}} H^{(\ell)}_{\delta_0\delta_1} H^{(\ell)}_{\delta_0\delta_2} \\ &\quad + \frac{1}{n_{\ell-1}}\sum_{\delta_4,\delta_5,\delta_6,\delta_7} H^{(\ell)}_{\delta_0\delta_1}\left\langle z_{\delta_4}\sigma''_{\delta_0}\sigma'_{\delta_1}\right\rangle_{G^{(\ell)}}\left\langle z_{\delta_5}\sigma'_{\delta_2}\sigma_{\delta_3}\right\rangle_{G^{(\ell)}} G^{\delta_4\delta_6}_{(\ell)} G^{\delta_5\delta_7}_{(\ell)} F^{(\ell)}_{\delta_6\delta_0\delta_7\delta_2} + O\left(\frac{1}{n^2}\right). \end{aligned} \tag{11.50}$$

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

Here, to go from the second expression to the third expression, we had to evaluate four expectations: the first expectation gives a single-neuron Gaussian expectation at leading order; the second expectation is subleading, cf. (8.71); the third expectation can also be evaluated with that same general NTK–preactivation cross-correlation formula (8.71), but in this case gives a leading term proportional to $F^{(\ell)}$; and the final expectation vanishes due to the unpaired m neural index, cf. similar manipulations in (8.87) and then the decomposition (8.82).

To simplify the second expectation in (11.44), we can again apply our general cross-correlation formula (11.42) for layer ℓ:

$$\begin{aligned}
&\frac{1}{n_\ell}\sum_{k,m=1}^{n_\ell}\mathbb{E}\left[\sigma'^{(\ell)}_{k;\delta_0}\sigma'^{(\ell)}_{k;\delta_1}\sigma'^{(\ell)}_{m;\delta_2}\sigma^{(\ell)}_{m;\delta_3}\widehat{\mathrm{d}H}^{(\ell)}_{kkm;\delta_0\delta_1\delta_2}\right] \qquad (11.51)\\
&\quad=\frac{1}{n_\ell n_{\ell-1}}\sum_{k,m=1}^{n_\ell}\sum_{\delta_4\in\mathcal{D}}\left[\left\langle\!\!\left\langle\frac{\partial}{\partial z^{(\ell)}_{m;\delta_4}}\left(\sigma'^{(\ell)}_{k;\delta_0}\sigma'^{(\ell)}_{k;\delta_1}\sigma'^{(\ell)}_{m;\delta_2}\sigma^{(\ell)}_{m;\delta_3}\right)\right\rangle\!\!\right\rangle_{G^{(\ell)}}Q^{(\ell)}_{\delta_0\delta_1\delta_2\delta_4}\right]+O\left(\frac{1}{n}\right)\\
&\quad=\left(\frac{n_\ell}{n_{\ell-1}}\right)\left[\left\langle\sigma''_{\delta_2}\sigma_{\delta_3}\right\rangle_{K^{(\ell)}}\left\langle\sigma'_{\delta_0}\sigma'_{\delta_1}\right\rangle_{G^{(\ell)}}Q^{(\ell)}_{\delta_0\delta_1\delta_2\delta_2}+\left\langle\sigma'_{\delta_2}\sigma'_{\delta_3}\right\rangle_{G^{(\ell)}}\left\langle\sigma'_{\delta_1}\sigma'_{\delta_2}\right\rangle_{G^{(\ell)}}Q^{(\ell)}_{\delta_0\delta_1\delta_2\delta_3}\right]\\
&\qquad+O\left(\frac{1}{n}\right).
\end{aligned}$$

Here in the second line, this time we didn't even write the terms proportional to δ_{mk}; just as we saw before when working out the P-recursion, the restriction $k=m$ for the double sum will make such terms subleading. In the final equality – again similarly to the P-recursion – we kept the off-diagonal terms with $k\neq m$, took the derivative, used Gaussian factorization, and performed the double sum.

Plugging these two simplified expectations back into our expression for $Q^{(\ell+1)}$ (11.44), we get our nearly-final recursion of the book:

$$\begin{aligned}
Q^{(\ell+1)}_{\delta_0\delta_1\delta_2\delta_3}&=\left(C_W^{(\ell+1)}\right)^2\left\langle\sigma''_{\delta_0}\sigma'_{\delta_1}\sigma'_{\delta_2}\sigma_{\delta_3}\right\rangle_{G^{(\ell)}}H^{(\ell)}_{\delta_0\delta_1}H^{(\ell)}_{\delta_0\delta_2}+\frac{\lambda_W^{(\ell+1)}}{C_W^{(\ell+1)}}F^{(\ell+1)}_{\delta_1\delta_0\delta_3\delta_2}\\
&\quad+\left(\frac{n_\ell}{n_{\ell-1}}\right)\left(C_W^{(\ell+1)}\right)^2\left\langle\sigma'_{\delta_0}\sigma'_{\delta_1}\right\rangle_{G^{(\ell)}}\left[\left\langle\sigma''_{\delta_2}\sigma_{\delta_3}\right\rangle_{G^{(\ell)}}Q^{(\ell)}_{\delta_0\delta_1\delta_2\delta_2}+\left\langle\sigma'_{\delta_2}\sigma'_{\delta_3}\right\rangle_{G^{(\ell)}}Q^{(\ell)}_{\delta_0\delta_1\delta_2\delta_3}\right]\\
&\quad+\left(\frac{n_\ell}{n_{\ell-1}}\right)\left(C_W^{(\ell+1)}\right)^2H^{(\ell)}_{\delta_0\delta_1}\sum_{\delta_4,\ldots,\delta_7\in\mathcal{D}}\left\langle z_{\delta_4}\sigma''_{\delta_0}\sigma'_{\delta_1}\right\rangle_{G^{(\ell)}}\left\langle z_{\delta_5}\sigma'_{\delta_2}\sigma_{\delta_3}\right\rangle_{G^{(\ell)}}G^{\delta_4\delta_6}_{(\ell)}G^{\delta_5\delta_7}_{(\ell)}F^{(\ell)}_{\delta_6\delta_0\delta_7\delta_2}\\
&\quad+O\left(\frac{1}{n}\right). \qquad (11.52)
\end{aligned}$$

Interestingly, in this case we see that this dNTK tensor $Q^{(\ell)}$ mixes with the NTK mean $H^{(\ell)}$ as well as the NTK–preactivation cross-correlation tensor $F^{(\ell)}$.[7] Again, since $H^{(\ell)}$

[7] Also of possible interest, while the recursions for $F^{(\ell)}$ (8.79) and $B^{(\ell)}$ (8.89) were sourced by the NTK mean $H^{(\ell)}$ but otherwise didn't mix with any finite-width tensors, the recursion for NTK–preactivation

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

and $F^{(\ell)}$ are both of order one, and since $Q^{(1)} = 0$, this recursion shows that $Q^{(\ell)}$ will also recursively stay of order one for all layers ℓ.

In conclusion, together with the general dNTK–preactivation cross-correlation formula (11.42), the P-Q recursions, (11.49) and (11.52), show that the leading dNTK–preactivation cross correlation is $1/n$-suppressed. In other words, the effects of the dNTK are visible *at finite width only*.

11.3 Effective Theory of the dNTK at Initialization

This section parallels our previous effective theory work on preactivation statistics and NTK–preactivation joint statistics (§5 ‖ §9). In particular, since we already know so many different reasons why criticality is essential, cf. §5, §6, §9, and §10, we'll spend less time on the disastrous consequences of not picking critical initialization hyperparameters and more time on finding asymptotic solutions to the P- and Q-recursions at criticality.

As we did in our discussion of preactivation criticality in §5 and NTK criticality in §9, throughout this section we'll set the bias variance $C_b^{(\ell)}$ and the rescaled weight variance $C_W^{(\ell)}$ to be uniform across layers,

$$C_b^{(\ell)} = C_b\,, \qquad C_W^{(\ell)} = C_W. \tag{11.53}$$

Further mirroring §5.4 and §9.1–§9.3, we'll consider MLPs with uniform hidden layer widths,

$$n_1 = \cdots = n_{L-1} \equiv n. \tag{11.54}$$

Finally, analogous to §9, we're only going to focus on single-input statistics, leaving the evaluation of the multi-input recursions as an adventure for thrill seekers.[8]

With these choices made, let's write down the leading single-input recursions for $P^{(\ell)}$ and $Q^{(\ell)}$. Dropping the sample indices and contributions that are subleading in $1/n$, in particular replacing the mean metric by the kernel $G^{(\ell)} \to K^{(\ell)}$ and the NTK mean by the frozen NTK $H^{(\ell)} \to \Theta^{(\ell)}$, the recursions (11.49) and (11.52) reduce to

$$\begin{aligned} P^{(\ell+1)} = {} & C_W^2 \left\langle \sigma''\sigma'\sigma'\sigma \right\rangle_{K^{(\ell)}} \left(\Theta^{(\ell)}\right)^2 + C_W \chi_\perp^{(\ell)} \left\langle \sigma''\sigma \right\rangle_{K^{(\ell)}} B^{(\ell)} \\ & + \left[C_W \chi_\perp^{(\ell)} \left\langle \sigma''\sigma \right\rangle_{K^{(\ell)}} + \left(\chi_\perp^{(\ell)}\right)^2 \right] P^{(\ell)}\,, \end{aligned} \tag{11.55}$$

cross-correlation tensor $D^{(\ell)}$ (8.77) mixed with the four-point vertex $V^{(\ell)}$, and the recursion for NTK-variance tensor $A^{(\ell)}$ (8.97) mixed with both $V^{(\ell)}$ and $D^{(\ell)}$. Thus, at least at this order, it seems like the correlations and fluctuations of the type $F^{(\ell)}$ and $B^{(\ell)}$ are potentially useful for the representation learning that the dNTK induces, while $V^{(\ell)}$, $D^{(\ell)}$, and $A^{(\ell)}$ may be more associated with instantiation-to-instantiation fluctuations.

[8] You'll have to generalize the $\gamma^{[a]}$ into a tensor product $\gamma^{[a]} \otimes \gamma^{[b]}$ and then further decompose such a basis according to the symmetries of the finite-width tensors you'll want to expand.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

$$Q^{(\ell+1)} = C_W^2 \left\langle \sigma'' \sigma' \sigma' \sigma \right\rangle_{K^{(\ell)}} \left(\Theta^{(\ell)}\right)^2 + \frac{\lambda_W^{(\ell+1)}}{C_W} F^{(\ell+1)} + 2h^{(\ell)} \chi_{\parallel}^{(\ell)} \Theta^{(\ell)} F^{(\ell)} + \left[C_W \chi_{\perp}^{(\ell)} \left\langle \sigma'' \sigma \right\rangle_{K^{(\ell)}} + \left(\chi_{\perp}^{(\ell)}\right)^2 \right] Q^{(\ell)}, \tag{11.56}$$

with the initial conditions (cf. §11.2.1)

$$P^{(1)} = Q^{(1)} = 0. \tag{11.57}$$

To simplify these expressions, we have recalled our two susceptibilities, the parallel susceptibility (5.50) and the perpendicular susceptibility (5.51), which are given by

$$\chi_{\parallel}^{(\ell)} \equiv \frac{C_W}{K^{(\ell)}} \left\langle \sigma' \sigma z \right\rangle_{K^{(\ell)}}, \tag{11.58}$$

$$\chi_{\perp}^{(\ell)} \equiv C_W \left\langle \sigma' \sigma' \right\rangle_{K^{(\ell)}}, \tag{11.59}$$

and we have also recalled our least favorite helper function (5.52),

$$h^{(\ell)} \equiv \frac{1}{2} \frac{d}{dK^{(\ell)}} \chi_{\perp}^{(\ell)} = \frac{C_W}{2K^{(\ell)}} \left\langle \sigma'' \sigma' z \right\rangle_{K^{(\ell)}}, \tag{11.60}$$

though we've given a new expression for it on the right-hand side, obtained through integration by parts, cf. (5.53).

To solve these recursions at criticality, we need to remember our scaling ansatz for observables (5.93),

$$\mathcal{O}^{(\ell)} = \left(\frac{1}{\ell}\right)^{p_{\mathcal{O}}} \left[c_{0,0} + c_{1,1} \left(\frac{\log \ell}{\ell}\right) + c_{1,0} \left(\frac{1}{\ell}\right) + c_{2,2} \left(\frac{\log^2 \ell}{\ell^2}\right) + \cdots \right]. \tag{11.61}$$

Here, solving the dNTK recursions will yield new critical exponents p_P and p_Q that describe the asymptotic depth scaling of $P^{(\ell)}$ and $Q^{(\ell)}$, respectively.

Additionally, in order to understand the relative size of the dNTK–preactivation cross correlation, we will need to identify appropriate dimensionless quantities just as we did before in §5.4 and §9.1. In this case, it will turn out that we should normalize by two factors of the (frozen) NTK:

$$\frac{P^{(\ell)}}{n \left(\Theta^{(\ell)}\right)^2} \sim \frac{1}{n} \left(\frac{1}{\ell}\right)^{p_P - 2p_\Theta}, \qquad \frac{Q^{(\ell)}}{n \left(\Theta^{(\ell)}\right)^2} \sim \frac{1}{n} \left(\frac{1}{\ell}\right)^{p_Q - 2p_\Theta}, \tag{11.62}$$

where on the right-hand side p_Θ is the critical exponent for the frozen NTK.

To see why these are the appropriate quantities to consider, recall our discussion of *dimensional analysis* in footnote 15 of §1.3 and that our notation of $[z]$ means "z is measured in units of $[z]$." Looking at our second-order update for preactivations in (11.9), and remembering that we can only add terms that have the same dimensions, we must have

$$[z] = [\eta]\,[\epsilon]\,[\widehat{H}] = [\eta]^2\,[\epsilon]^2\,[\widehat{\mathrm{d}H}], \tag{11.63}$$

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

from which it's clear that $[\eta]\,[\epsilon] = [z]\,[\widehat{H}]^{-1}$, and subsequently we see that P and Q have dimensions of NTK squared:

$$[P] = [Q] \equiv [z]\,[\widehat{\mathrm{d}H}] = [\widehat{H}]^2. \tag{11.64}$$

If this still seems a little counterintuitive, the utility of considering these particular ratios (11.62) will become even more apparent when we analyze the stochastic prediction of a fully-trained finite-width network in §∞.2.3.

After solving the P- and Q-recursions for both universality classes and looking at these dimensionless quantities (11.62), we'll again find *scaling laws* that transcend universality class:

$$p_P - 2p_\Theta = -1\,, \quad p_Q - 2p_\Theta = -1. \tag{11.65}$$

Specifically, we'll see that these laws hold for both the scale-invariant and $K^\star = 0$ universality classes.[9] Thus, we'll be able to conclude that all the leading finite-width effects of the preactivation–NTK–dNTK joint distribution are *relevant* – in the sense of RG flow – controlled by the same ℓ/n perturbative cutoff.

Now that we're fully prepared for what we're going to see, let's actually solve the dNTK recursions for our two important universality classes.

11.3.1 Scale-Invariant Universality Class

First, let's recall some previous results that we'll need in order to evaluate the recursions (11.55) and (11.56). For the scale-invariant universality class, we know from (5.62), (9.33), and (9.37) that

$$\chi_\perp^{(\ell)} = C_W A_2 \equiv \chi\,, \qquad h^{(\ell)} = 0\,, \qquad \left\langle \sigma'\sigma'\sigma\sigma \right\rangle_{K^{(\ell)}} = A_4 K^{(\ell)}, \tag{11.66}$$

where as a reminder $A_2 \equiv (a_+^2 + a_-^2)/2$ and $A_4 \equiv (a_+^4 + a_-^4)/2$, and the $a_\pm$ are the respective slopes of the positive/negative linear pieces of the activation function, cf. (5.59).

Next, we see that all the new Gaussian expectations in the recursions (11.55) and (11.56) involve the second derivative of the activation. For these, we need to be somewhat careful with nonlinear scale-invariant activation functions, since they have an undifferentiable kink at the origin. (For linear activation functions, $\sigma''(z) = 0$, and there's no subtlety.) For the first of these new expectations, note that we can integrate it by parts as

$$\left\langle \sigma''\sigma \right\rangle_K = \frac{1}{\sqrt{2\pi K}} \int_{-\infty}^{\infty} dz e^{-\frac{z^2}{2K}} \left(\frac{d}{dz}\sigma'\right)\sigma = \frac{1}{K}\left\langle z\sigma'\sigma \right\rangle_K - \left\langle \sigma'\sigma' \right\rangle_K = 0\,, \tag{11.67}$$

where in the final equality we used

$$\frac{1}{K}\left\langle z\sigma'\sigma \right\rangle_K = \left\langle \sigma'\sigma' \right\rangle_K = A_2. \tag{11.68}$$

[9] To be more specific, for the scale-invariant class, we'll find that $P^{(\ell)}$ identically vanishes and that $Q^{(\ell)}$ solely determines the leading finite-width dNTK–preactivation cross correlation.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

Similarly, integrating the other new Gaussian expectation by parts, we find

$$\left\langle \sigma''\sigma'\sigma'\sigma\right\rangle_K = \frac{1}{K}\left\langle z\sigma'\sigma'\sigma'\sigma\right\rangle_K - 2\left\langle \sigma''\sigma'\sigma'\sigma\right\rangle_K - \left\langle \sigma'\sigma'\sigma'\sigma'\right\rangle_K . \tag{11.69}$$

Rearranging this, we easily see that

$$\left\langle \sigma''\sigma'\sigma'\sigma\right\rangle_K = \frac{1}{3K}\left\langle z\sigma'\sigma'\sigma'\sigma\right\rangle_K - \frac{1}{3}\left\langle \sigma'\sigma'\sigma'\sigma'\right\rangle_K = 0 , \tag{11.70}$$

where in the final equality we used

$$\frac{1}{K}\left\langle z\sigma'\sigma'\sigma'\sigma\right\rangle_K = \left\langle \sigma'\sigma'\sigma'\sigma'\right\rangle_K = A_4. \tag{11.71}$$

Thus, we can safely ignore both of these new expectations.

Substituting in all of (11.66) and ignoring ignorable expectations, our single-input recursions (11.55) and (11.56) are extremely simple:

$$P^{(\ell+1)} = \chi^2 P^{(\ell)}, \tag{11.72}$$

$$Q^{(\ell+1)} = \chi^2 Q^{(\ell)} + \frac{\lambda_W^{(\ell+1)}}{C_W} F^{(\ell+1)}. \tag{11.73}$$

In particular, since our initial condition (11.57) is $P^{(1)} = 0$, we see immediately that $P^{(\ell)}$ vanishes identically for all layers:

$$P^{(\ell)} = 0. \tag{11.74}$$

In contrast, for $Q^{(\ell)}$ we see that the susceptibility χ is going to generically lead to exponential behavior.

Now, let's tune to scale-invariant criticality by setting the initialization hyperparameters as $C_b = 0$ and $C_W = 1/A_2$. As a consequence, this fixes the susceptibility to unity, $\chi = 1$, and leaves the kernel fixed for all layers as $K^{(\ell)} = K^\star$. Additionally, let's pick our training hyperparameters according to the learning rate equivalence principle, for which we're instructed to choose layer-independent learning rates (9.94) as

$$\lambda_b^{(\ell)} = \frac{\widetilde{\lambda}_b}{L}, \qquad \lambda_W^{(\ell)} = \frac{\widetilde{\lambda}_W}{L}. \tag{11.75}$$

Finally, with these hyperparameter choices, the single-input solution for the frozen NTK (9.44) and the single-input solution for the NTK–preactivation cross correlation $F^{(\ell)}$ (9.50) are given by

$$\Theta^{(\ell)} = \left(\widetilde{\lambda}_b + \widetilde{\lambda}_W A_2 K^\star\right)\frac{\ell}{L}, \tag{11.76}$$

$$F^{(\ell)} = \frac{\ell(\ell-1)}{2L}\left[\frac{A_4}{A_2^2}\left(\widetilde{\lambda}_b + \widetilde{\lambda}_W A_2 K^\star\right)K^\star\right]. \tag{11.77}$$

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

Plugging in the critical initialization hyperparameters, the fixed kernel, the learning rates (11.75), and the expression for $F^{(\ell)}$ (11.77), the Q-recursion (11.73) becomes

$$Q^{(\ell+1)} = Q^{(\ell)} + \frac{\ell(\ell+1)}{2L^2}\left[\frac{A_4}{A_2}\left(\widetilde{\lambda}_b + \widetilde{\lambda}_W A_2 K^\star\right)\widetilde{\lambda}_W K^\star\right]. \tag{11.78}$$

This simple recursion is exactly solved by

$$Q^{(\ell)} = \frac{\ell(\ell^2-1)}{6L^2}\left[\frac{A_4}{A_2}\left(\widetilde{\lambda}_b + \widetilde{\lambda}_W A_2 K^\star\right)\widetilde{\lambda}_W K^\star\right], \tag{11.79}$$

which satisfies the initial condition $Q^{(1)} = 0$.

With this solution, we can identify the critical exponent associated with the large-ℓ behavior of $Q^{(\ell)}$: $p_Q = -3$. Further, we see that our dimensionless quantity (11.62) will satisfy the scaling law (11.65) as promised,

$$p_Q - 2p_\Theta = -1, \tag{11.80}$$

where we have also used $p_\Theta = -1$ from the scale-invariant frozen NTK solution reprinted above in (11.76). More specifically, substituting in the solution for $\Theta^{(\ell)}$ (11.76) and the solution for $Q^{(\ell)}$ (11.79) into the dimensionless ratio (11.62), we find

$$\frac{Q^{(\ell)}}{n\left(\Theta^{(\ell)}\right)^2} = \frac{A_4}{6A_2}\left[\frac{\widetilde{\lambda}_W K^\star}{\widetilde{\lambda}_b + \widetilde{\lambda}_W A_2 K^\star}\right]\frac{\ell}{n} + \cdots. \tag{11.81}$$

Thus, we have verified the ℓ/n scaling of the leading dNTK–preactivation cross correlation for scale-invariant activation functions.

11.3.2 $K^\star = 0$ Universality Class

For the $K^\star = 0$ universality class, we'll begin again by recalling some previous results. First, we know from (5.84), (5.85), (9.64), and (11.60) the following:

$$\chi_\parallel(K) = \left(C_W\sigma_1^2\right)\left[1 + 2a_1 K + O\left(K^2\right)\right], \tag{11.82}$$

$$\chi_\perp(K) = \left(C_W\sigma_1^2\right)\left[1 + b_1 K + O\left(K^2\right)\right], \tag{11.83}$$

$$C_W^2\left\langle\sigma'\sigma'\sigma\sigma\right\rangle_K = \left(C_W\sigma_1^2\right)^2\left[K + O|1\left(K^2\right)\right], \tag{11.84}$$

$$h(K) = \frac{1}{2}\frac{d}{dK}\chi_\perp(K) = \left(C_W\sigma_1^2\right)\left[\frac{b_1}{2} + O\left(K^1\right)\right]. \tag{11.85}$$

To interpret these results, remember that we Taylor-expanded the activation function as

$$\sigma(z) = \sum_{p=0}^{\infty}\frac{\sigma_p}{p!}z^p, \tag{11.86}$$

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

defined the following combination of Taylor coefficients for convenience

$$a_1 \equiv \left(\frac{\sigma_3}{\sigma_1}\right) + \frac{3}{4}\left(\frac{\sigma_2}{\sigma_1}\right)^2 , \tag{11.87}$$

$$b_1 \equiv \left(\frac{\sigma_3}{\sigma_1}\right) + \left(\frac{\sigma_2}{\sigma_1}\right)^2 , \tag{11.88}$$

and required that all activation functions in this class satisfy $\sigma_0 = 0$ and $\sigma_1 \neq 0$. Then, making analogous Taylor expansions and performing Gaussian integrations order by order, we can evaluate the new Gaussian expectations in the recursions (11.55) and (11.56) as

$$C_W \left\langle \sigma'' \sigma \right\rangle_K = \left(C_W \sigma_1^2\right) \left[(2a_1 - b_1)K + O\left(K^2\right)\right] , \tag{11.89}$$

$$C_W^2 \left\langle \sigma'' \sigma' \sigma' \sigma \right\rangle_K = \left(C_W \sigma_1^2\right)^2 \left[(-6a_1 + 7b_1)K + O\left(K^2\right)\right] . \tag{11.90}$$

Now, let's jump right into $K^\star = 0$ criticality (5.90) by tuning the initialization hyperparameters as $C_b = 0$ and $C_W = 1/\sigma_1^2$. At the same time, let's also tune our training hyperparameters according to the learning rate equivalence principle, which for $K^\star = 0$ activation functions is given by (9.95),

$$\lambda_b^{(\ell)} = \widetilde{\lambda}_b \left(\frac{1}{\ell}\right)^{p_\perp} L^{p_\perp - 1} , \qquad \lambda_W^{(\ell)} = \widetilde{\lambda}_W \left(\frac{L}{\ell}\right)^{p_\perp - 1} , \tag{11.91}$$

where the critical exponent for perpendicular perturbations is defined as $p_\perp \equiv b_1/a_1$. With these hyperparameter settings, let's also record the other single-input solutions that we need for the recursions, the kernel $K^{(\ell)}$ (5.92), the frozen NTK $\Theta^{(\ell)}$ (9.71), the NTK–preactivation cross correlation $F^{(\ell)}$ (9.81), and the NTK variance $B^{(\ell)}$ (9.82):

$$K^{(\ell)} = \left[\frac{1}{(-a_1)}\right] \frac{1}{\ell} + \cdots , \tag{11.92}$$

$$\Theta^{(\ell)} = \left[\widetilde{\lambda}_b + \frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)}\right] \left(\frac{L}{\ell}\right)^{p_\perp - 1} + \cdots , \tag{11.93}$$

$$F^{(\ell)} = \frac{1}{(5 - p_\perp)} \left[\frac{1}{(-a_1)}\right] \left[\widetilde{\lambda}_b + \frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)}\right] \left(\frac{L}{\ell}\right)^{p_\perp - 1} + \cdots , \tag{11.94}$$

$$B^{(\ell)} = \frac{L^{2p_\perp - 2}}{3} \left[\widetilde{\lambda}_b + \frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)}\right]^2 \left(\frac{1}{\ell}\right)^{2p_\perp - 3} + \cdots . \tag{11.95}$$

Finally, plugging all our collected results and tunings (11.82)–(11.85) and (11.89)–(11.95) into the P-recursion (11.55) and the Q-recursion (11.56), we get

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

$$P^{(\ell+1)} = \left[1 - \frac{(p_\perp + 2)}{\ell} + \cdots\right] P^{(\ell)} + \frac{(p_\perp - 2)}{3}\left[\widetilde{\lambda}_b + \frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)}\right]^2 \left(\frac{L}{\ell}\right)^{2p_\perp - 2} + \cdots , \tag{11.96}$$

$$\begin{aligned} Q^{(\ell+1)} = &\left[1 - \frac{(p_\perp + 2)}{\ell} + \cdots\right] Q^{(\ell)} \\ &+ \frac{1}{(5 - p_\perp)}\left[(1 - p_\perp)\frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)} - p_\perp \widetilde{\lambda}_b\right]\left[\widetilde{\lambda}_b + \frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)}\right]\left(\frac{L}{\ell}\right)^{2p_\perp - 2} + \cdots . \end{aligned} \tag{11.97}$$

Let's tackle the P-recursion first. Plugging our scaling ansatz (11.61) into the recursion (11.96) and matching the terms, we find an asymptotic solution

$$P^{(\ell)} = -\frac{L^{2p_\perp - 2}}{3}\left(\frac{2 - p_\perp}{5 - p_\perp}\right)\left[\widetilde{\lambda}_b + \frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)}\right]^2 \left(\frac{1}{\ell}\right)^{2p_\perp - 3} + \cdots . \tag{11.98}$$

Thus, the critical exponent for this dNTK–preactivation cross correlation is $p_P = 2p_\perp - 3$, and we obtain our promised scaling law (11.65)

$$p_P - 2p_\Theta = -1, \tag{11.99}$$

after substituting in $p_\Theta = p_\perp - 1$ from the $K^\star = 0$ frozen NTK solution (11.93). More specifically, substituting in the solution for $\Theta^{(\ell)}$ (11.93) and the solution for $P^{(\ell)}$ (11.98) into the dimensionless ratio (11.62), we get

$$\frac{P^{(\ell)}}{n\left(\Theta^{(\ell)}\right)^2} = -\frac{1}{3}\left(\frac{2 - p_\perp}{5 - p_\perp}\right)\frac{\ell}{n} + \cdots , \tag{11.100}$$

which *(i)* scales as ℓ/n, *(ii)* is independent of the training hyperparameters, and *(iii)* is manifestly negative, given $p_\perp \leq 1$.[10]

Similarly for the Q-recursion (11.97), plugging our scaling ansatz (11.61) into the recursion (11.97) and matching the terms one final time, we find

$$Q^{(\ell)} = \frac{L^{2p_\perp - 2}}{(5 - p_\perp)^2}\left[1 - p_\perp - \frac{\widetilde{\lambda}_b}{\widetilde{\lambda}_b + \widetilde{\lambda}_W \sigma_1^2/(-a_1)}\right]\left[\widetilde{\lambda}_b + \frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)}\right]^2 \left(\frac{1}{\ell}\right)^{2p_\perp - 3} . \tag{11.101}$$

This gives a critical exponent of $p_Q = 2p_\perp - 3$ and a dimensionless ratio of

$$\frac{Q^{(\ell)}}{n\left(\Theta^{(\ell)}\right)^2} = \frac{1}{(5 - p_\perp)^2}\left[1 - p_\perp - \frac{\widetilde{\lambda}_b}{\widetilde{\lambda}_b + \widetilde{\lambda}_W \sigma_1^2/(-a_1)}\right]\frac{\ell}{n} + \cdots . \tag{11.102}$$

[10] To recall why $p_\perp \leq 1$, without flipping back to footnote 33 of §10, *(i)* remember that we must have $a_1 < 0$ in order for the kernel $K^{(\ell)}$ to stay positive when asymptotically approaching the $K^\star = 0$ fixed point, cf. (11.92), *(ii)* note to yourself that $b_1 \geq a_1$, cf. (11.87) and (11.88), and *(iii)* realize that $a_1 < 0$ and $b_1 \geq a_1$ together imply that $p_\perp \equiv b_1/a_1 \leq 1$.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

Thus, we've now fully verified our scaling law (11.65):

$$p_Q - 2p_\Theta = -1. \tag{11.103}$$

♦♦♦♦

In conclusion, all the nonzero dimensionless dNTK–preactivation cross correlations will grow as ℓ/n at leading order in the finite-width expansion. As the dNTK leads to a dynamical NTK, this is sufficient to see that deep finite-width networks are representation learners.

In the next section, we're going to set aside our direct investigation of deep MLPs and instead focus on a pedagogical model of representation learning that directly incorporates the effect of a nonzero dNTK. In the following chapter, we'll return to our finite-width networks and complete our goal of computing the effective distribution over wide and deep finite-width MLPs after training. The simplified model presented in the next section will make it easier to understand the results of our later analysis of such fully-trained networks.

11.4 Nonlinear Models and Nearly-Kernel Methods

In §10.4, we placed infinite-width networks into a broader context of machine learning models. There, we discussed how the infinite-width network can be understood as either a *linear model* with fixed random features or, dually, as a *kernel method* with the kernel given by the frozen NTK. Now we are ready to find a broader context for finite-width networks.

In this chapter, we've seen that the statistics needed to compute the dynamics of finite-width networks (§11.2 and §11.3) are far more complicated than for infinite-width networks, nontrivially incorporating the second derivative of the network function. This additional complexity is irreducibly encapsulated by a single object: the dNTK (§11.1). The dNTK enables the NTK to evolve during training, leading to nontrivial *representation learning* from the training data.

The goal of this section is to abstract this property away from the deep learning framework and distill it into a **minimal model of representation learning** that captures all of its important features. Such a model provides a framework for studying the type of representation learning exhibited by deep neural networks, but more succinctly and more broadly. In other words, we hope that this endeavor will extend the standard toolbox of machine learning.

First, in §11.4.1, we'll extend the linear models discussed in §10.4.1 to *nonlinear models*. Then, in §11.4.2, we'll give a dual description of these nonlinear models, *nearly-kernel methods*, which will extend the standard kernel methods that we discussed in §10.4.2. Finally, in §11.4.3, as we did analogously for infinite-width networks before in §10.4.3, we'll see how finite-width networks can be understood in terms of this new framework.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

11.4.1 Nonlinear Models

As a reminder from §10.4.1, a *linear model* is given by (10.119)

$$z_i(x_\delta;\theta) = \sum_{j=0}^{n_f} W_{ij}\phi_j(x_\delta), \tag{11.104}$$

where the model parameters are given by the weight matrix $\theta = W_{ij}$, the model's features are given by the *feature function* $\phi_j(x)$, and we again adopt the old-school convention for incorporating biases, including a constant feature $\phi_0(x) \equiv 1$ so that the bias vector is given by $W_{i0} \equiv b_i$. Note that since we're not discussing neural networks in particular, there are no neural indices or layer indices here. Instead, in this equation, δ is a *sample index*, running over the $N_{\mathcal{D}}$ samples in the dataset $\delta \in \mathcal{D}$; i is a *vectorial index*, running over the n_{out} vectorial components of the model output z_i; and j is a *feature index*, running over $(n_f + 1)$ different features. In this setup, a feature function $\phi_j(x)$ is computed on an input sample x, and the weight matrix W_{ij} determines the effect of the j-th feature on the i-th component of the output. Traditionally, feature functions $\phi_j(x)$ are often *designed* such that the linear model works well for the desired task after optimization or *linear regression*.

To go beyond this linear paradigm, let's slightly *deform* it to get a ***nonlinear* model**:

$$z_i(x_\delta;\theta) = \sum_{j=0}^{n_f} W_{ij}\phi_j(x_\delta) + \frac{\epsilon}{2}\sum_{j_1,j_2=0}^{n_f} W_{ij_1}W_{ij_2}\psi_{j_1j_2}(x_\delta) + \cdots . \tag{11.105}$$

Here, $\epsilon \ll 1$ is small parameter that controls the size of the deformation, and more importantly, we've introduced another set of feature functions, $\psi_{j_1j_2}(x)$, with *two* feature indices, making the model nonlinear in its weights. By definition, $\psi_{j_1j_2}(x)$ is symmetric under the exchange of the feature indices $j_1 \leftrightarrow j_2$, and the factor of $1/2$ is there because of the double counting of the double sum. For reasons that will be made clear shortly, we will call each component of $\psi_{j_1j_2}(x)$ a **meta feature function**, and so there are $(n_f + 1)(n_f + 2)/2$ different meta feature functions.

To familiarize ourselves with the features of this model, let's see how the model outputs change given a small change in the model parameters $W_{ij} \to W_{ij} + đW_{ij}$:

$$\begin{aligned}
&z_i(x_\delta;\theta + đ\theta) \\
&\quad = z_i(x_\delta;\theta) + \sum_{j=0}^{n_f} đW_{ij}\left[\phi_j(x_\delta) + \epsilon\sum_{j_1=0}^{n_f} W_{ij_1}\psi_{j_1j}(x_\delta)\right] \\
&\qquad + \frac{\epsilon}{2}\sum_{j_1,j_2=0}^{n_f} đW_{ij_1}đW_{ij_2}\psi_{j_1j_2}(x_\delta) + \cdots \\
&\quad = z_i(x_\delta;\theta) + \sum_{j=0}^{n_f} đW_{ij}\,\phi^{\text{E}}_{ij}(x_\delta;\theta) + \frac{\epsilon}{2}\sum_{j_1,j_2=0}^{n_f} đW_{ij_1}đW_{ij_2}\psi_{j_1j_2}(x_\delta) + \cdots ,
\end{aligned} \tag{11.106}$$

where in the last line we summarized the quantity in the square bracket in terms of an **effective feature function**,

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

$$\phi^{\mathrm{E}}_{ij}(x_\delta;\theta) \equiv \frac{dz_i(x_\delta;\theta)}{dW_{ij}} = \phi_j(x_\delta) + \epsilon \sum_{k=0}^{n_f} W_{ik}\psi_{kj}(x_\delta). \tag{11.107}$$

The utility of this is as follows: the linear response of the model to changing its parameters is as if it *effectively* has a feature function, $\phi^{\mathrm{E}}_{ij}(x_\delta;\theta)$, which itself depends on the value of the model parameters. Moreover, the change in the effective feature function given a small change in the model parameters $W_{ik} \to W_{ik} + đW_{ik}$ is given by

$$\phi^{\mathrm{E}}_{ij}(x_\delta;\theta + đ\theta) = \phi^{\mathrm{E}}_{ij}(x_\delta;\theta) + \epsilon \sum_{k=0}^{n_f} đW_{ik}\,\psi_{kj}(x_\delta). \tag{11.108}$$

Thus, the effective features $\phi^{\mathrm{E}}_{ij}(x_\delta;\theta)$ are *learnable*, evolving as a linear model, and for these effective features, the meta features $\psi_{kj}(x_\delta)$ play the role usually played by the features.[11] In this way, we can think of our nonlinear model as having a hierarchical structure, where the features evolve as if they are described by a linear model according to (11.108), while the model's output evolves in a more complicated nonlinear way according to (11.106).

Note that we could extend this hierarchical structure further, e.g., by further deforming our nonlinear model (11.105) with a term that's cubic in the parameters and includes a three-indexed *meta-meta feature function* that analogously allows the meta feature functions to learn.[12] However, as the quadratic term already suffices to provide a minimal model of representation learning – just as a nonzero dNTK sufficed to induce nontrivial representation learning in the MLP – from here on, we'll explicitly truncate the model (11.105) at the quadratic order, giving us a **quadratic model**.[13]

To understand how representation learning works in this model, we should find the optimal values for the weights $W^\star_{ij}$ given a training set $\mathcal{A}$. Mirroring what we did before for linear models, let's minimize the MSE loss, which for our quadratic model is given by

$$\begin{aligned}\mathcal{L}_{\mathcal{A}}(\theta) &= \frac{1}{2}\sum_{\tilde{\alpha}\in\mathcal{A}}\sum_{i=1}^{n_{\mathrm{out}}}\Big[y_{i;\tilde{\alpha}} - z_i(x_{\tilde{\alpha}};\theta)\Big]^2 \\ &= \frac{1}{2}\sum_{\tilde{\alpha}\in\mathcal{A}}\sum_{i=1}^{n_{\mathrm{out}}}\left[y_{i;\tilde{\alpha}} - \sum_{j=0}^{n_f} W_{ij}\,\phi_j(x_{\tilde{\alpha}}) - \frac{\epsilon}{2}\sum_{j_1,j_2=0}^{n_f} W_{ij_1}W_{ij_2}\psi_{j_1j_2}(x_{\tilde{\alpha}})\right]^2 .\end{aligned} \tag{11.109}$$

[11] This role of the meta features $\psi_{kj}(x_\delta)$ in the linear model for the effective features explains our choice of name. Note also that in this setup, we assume that both the feature function $\phi_j(x)$ and the meta feature function $\psi_{j_1j_2}(x)$ are picked by the model designer, just as the feature function was before for the linear model.

[12] Such deformations are essentially equivalent to incorporating further terms in the Taylor expansion of an arbitrary general model function $z_i(x_\delta;\theta)$, where the linear model is the base model, and the nonlinear model (11.105) gives the first improvement from the expansion.

[13] To leading order in the $1/n$ expansion, finite-width networks cannot self-consistently be described by a truncated quadratic model; instead, they fall into a class of *cubic models*, with evolving meta feature functions. This is one of the ways in which such finite-width networks are *nonminimal* representation learners and is the main reason for discussing quadratic models first before moving on to the more complicated analysis of finite-width networks with their dynamical dNTKs.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

Supervised learning with such a quadratic model (11.105) doesn't have a particular name, but if it did, we'd all probably agree that its name should be **quadratic regression**. However, unlike linear regression – where the MSE loss (10.120) was quadratic in the model parameters – quadratic regression – where the loss is now *quartic* in the parameters – does not in general yield analytical solutions. Nevertheless, we can perturbatively find a solution to the quadratic regression problem by expanding in our small parameter ϵ, for which the regression is *nearly linear*.

Nearly-Linear Quadratic Regression

Taking the derivative of the loss $\mathcal{L}_{\mathcal{A}}$ (11.109) with respect to the model parameter W_{ij_0} and setting it to zero, we find

$$0 = \sum_{\tilde{\alpha}} \phi^{\mathrm{E}}_{ij_0}(x_{\tilde{\alpha}}; \theta^\star) \left[z_i(x_{\tilde{\alpha}}; \theta^\star) - y_{i;\tilde{\alpha}} \right], \tag{11.110}$$

making clear that the effective features (11.107) give the linear response of the model. Next, substituting in for the quadratic model (11.104) and the effective feature function (11.107) and rearranging, we find

$$\sum_{\tilde{\alpha}\in\mathcal{A}} \left\{ \sum_{j_1=0}^{n_f} W^\star_{ij_1} \phi_{j_1}(x_{\tilde{\alpha}}) \phi_{j_0}(x_{\tilde{\alpha}}) + \epsilon \sum_{j_1,j_2=0}^{n_f} W^\star_{ij_1} W^\star_{ij_2} \left[\phi_{j_1}(x_{\tilde{\alpha}}) \psi_{j_2 j_0}(x_{\tilde{\alpha}}) + \frac{1}{2} \psi_{j_1 j_2}(x_{\tilde{\alpha}}) \phi_{j_0}(x_{\tilde{\alpha}}) \right] \right\}$$
$$= \sum_{\tilde{\alpha}\in\mathcal{A}} y_{i;\tilde{\alpha}} \left[\phi_{j_0}(x_{\tilde{\alpha}}) + \epsilon \sum_{j_1=0}^{n_f} W^\star_{ij_1} \psi_{j_1 j_0}(x_{\tilde{\alpha}}) \right] + O\left(\epsilon^2\right). \tag{11.111}$$

This expression contains the two terms we found for linear regression in (10.121) as well as three additional terms proportional to the meta features $\psi_{j_1 j_2}(x)$. Additionally, we truncated the $O(\epsilon^2)$ term since ϵ is assumed to be parametrically small. Importantly, we see clearly that the equation is overall nonlinear, with two terms quadratic in the weights. In the language of physics, the linear equation of linear regression is *free* and exactly solvable, while the nonlinear equation of quadratic regression is *interacting*. Since ϵ multiplies all the new terms associated with quadratic regression, the nonlinear terms are all small, and so (11.111) exhibits *weakly-interacting* dynamics. This means that we can systematically solve this nonlinear equation via perturbation theory.

With that in mind, let's decompose the optimal weight matrix into a free linear part and an interacting nonlinear part as

$$W^\star_{ij} \equiv W^{\mathrm{F}}_{ij} + W^{\mathrm{I}}_{ij}, \tag{11.112}$$

with the idea being that the free part W^{F}_{ij} will solve the free linear regression equation (10.121), while the interacting part W^{I}_{ij} will solve the remaining linearized equation after substituting back in the solution for W^{F}_{ij}. Given that the quadratic regression problem (11.111) becomes a linear regression problem (10.121) in the limit of $\epsilon \to 0$, we naturally

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

expect that the interacting part of the optimal weights should be proportional to the small parameter $W^{\mathrm{I}}_{ij} = O(\epsilon)$.

Let's quickly review our **direct optimization** solution to linear regression from §10.4.1 in the current context: defining an (n_f+1)-by-(n_f+1) symmetric matrix (10.122)

$$M_{j_1 j_2} \equiv \sum_{\tilde{\alpha}\in\mathcal{A}} \phi_{j_1}(x_{\tilde{\alpha}})\, \phi_{j_2}(x_{\tilde{\alpha}})\,, \tag{11.113}$$

the linear part of the quadratic regression problem (11.111) can be written as

$$\sum_{j_1=0}^{n_f} W^{\mathrm{F}}_{ij_1} M_{j_1 j_0} = \sum_{\tilde{\alpha}\in\mathcal{A}} y_{i;\tilde{\alpha}} \phi_{j_0}(x_{\tilde{\alpha}}), \tag{11.114}$$

which can be solved by multiplication by the inverse $(M^{-1})_{j_0 j}$,

$$W^{\mathrm{F}}_{ij} = \sum_{j_0=0}^{n_f} \sum_{\tilde{\alpha}\in\mathcal{A}} y_{i;\tilde{\alpha}} \phi_{j_0}(x_{\tilde{\alpha}}) \left(M^{-1}\right)_{j_0 j}. \tag{11.115}$$

Recall that the inverse $(M^{-1})_{j_0 j}$ will not uniquely exist if we're in the *overparameterized* regime with more features than training examples, $(n_f + 1) > N_{\mathcal{A}}$, but we can use our regularization trick (10.127) to pick out a particular inverse. Going forward, we will assume that we're in this overparameterized regime and that the inverse was picked in this way.

Next, plugging in our decomposition (11.112) into our equation (11.111) and collecting the terms of order ϵ, remembering also that $W^{\mathrm{I}}_{ij} = O(\epsilon)$, we find for our linearized interacting dynamics,

$$\begin{aligned}\sum_{j_1=0}^{n_f} W^{\mathrm{I}}_{ij_1} M_{j_1 j_0} =& \epsilon \sum_{j_1=0}^{n_f} W^{\mathrm{F}}_{ij_1} \sum_{\tilde{\alpha}\in\mathcal{A}} y_{i;\tilde{\alpha}} \psi_{j_1 j_0}(x_{\tilde{\alpha}}) - \epsilon \sum_{j_1,j_2=0}^{n_f} W^{\mathrm{F}}_{ij_1} W^{\mathrm{F}}_{ij_2} \sum_{\tilde{\alpha}\in\mathcal{A}} \phi_{j_1}(x_{\tilde{\alpha}}) \psi_{j_2 j_0}(x_{\tilde{\alpha}}) \\ & - \frac{\epsilon}{2} \sum_{j_1,j_2=0}^{n_f} W^{\mathrm{F}}_{ij_1} W^{\mathrm{F}}_{ij_2} \sum_{\tilde{\alpha}\in\mathcal{A}} \phi_{j_0}(x_{\tilde{\alpha}}) \psi_{j_1 j_2}(x_{\tilde{\alpha}}) + O\left(\epsilon^2\right). \end{aligned} \tag{11.116}$$

Here on the right-hand side, the first two terms actually cancel each other, since the free solution satisfies

$$\sum_{j=0}^{n_f} W^{\mathrm{F}}_{ij} \phi_j(x_{\tilde{\alpha}}) = y_{i;\tilde{\alpha}}, \tag{11.117}$$

i.e., for overparameterized models, the linear part of the optimized model can correctly predict all the training-set examples.[14] After making that cancellation, we can multiply by the inverse $(M^{-1})_{j_0 j}$ to find a solution:

[14] This is shown in detail in §10.4.2. Note that for *underparametrized* models, the solution can still be analyzed, but the details will be different. We're focusing on overparameterized models here since *(i)* deep learning models are typically overparameterized and *(ii)* we suspect that the sort of representation learning our model exhibits is most useful in that regime. We'll elaborate on this quite a bit more in Epilogue ε.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

$$W^{\mathrm{I}}_{ij} = -\frac{\epsilon}{2} \sum_{j_1,j_2,j_3=0}^{n_f} W^{\mathrm{F}}_{ij_1} W^{\mathrm{F}}_{ij_2} \sum_{\tilde{\alpha}\in\mathcal{A}} \left[\psi_{j_1 j_2}(x_{\tilde{\alpha}})\phi_{j_3}(x_{\tilde{\alpha}})\right] \left(M^{-1}\right)_{j_3 j} + O\left(\epsilon^2\right). \tag{11.118}$$

In particular, the free solution, (11.115), and the interacting solution, (11.118), together solve the nonlinear optimization problem (11.111) to order ϵ.[15]

Finally, having obtained the solution, we can throw away the training data and simply store the optimal parameters $W^{\star}_{ij} = W^{\mathrm{F}}_{ij} + W^{\mathrm{I}}_{ij}$, making predictions on novel test inputs $x_{\dot{\beta}}$ as

$$\begin{aligned} z_i(x_{\dot{\beta}};\theta^{\star}) &= \sum_{j=0}^{n_f} W^{\star}_{ij}\phi_j(x_{\dot{\beta}}) + \frac{\epsilon}{2} \sum_{j_1,j_2=0}^{n_f} W^{\star}_{ij_1} W^{\star}_{ij_2} \psi_{j_1 j_2}(x_{\dot{\beta}}) \\ &= \sum_{j=0}^{n_f} W^{\mathrm{F}}_{ij}\phi_j(x_{\dot{\beta}}) + \sum_{j=0}^{n_f} W^{\mathrm{I}}_{ij}\phi_j(x_{\dot{\beta}}) + \frac{\epsilon}{2} \sum_{j_1,j_2=0}^{n_f} W^{\mathrm{F}}_{ij_1} W^{\mathrm{F}}_{ij_2} \psi_{j_1 j_2}(x_{\dot{\beta}}) + O\left(\epsilon^2\right) \\ &= \frac{1}{2} \sum_{j=0}^{n_f} W^{\star}_{ij} \left[\phi_j(x_{\dot{\beta}}) + \phi^{\mathrm{E}}_{ij}(x_{\dot{\beta}};\theta^{\star})\right] + O\left(\epsilon^2\right). \end{aligned} \tag{11.119}$$

Here, we've given two different ways to think about the optimal output. On the first and second lines, we simply have the prediction of the quadratic model (11.105) expressed in terms of the fixed features $\phi_j(x_{\dot{\beta}})$ and the fixed meta features $\psi_{j_1 j_2}(x_{\dot{\beta}})$. This presentation makes the nonlinearity manifest. After regrouping the terms and using our definition (11.107), in the last line we instead wrote the model prediction in the form of a linear model, where we see that features in this interpretation are the mean of the fixed unlearned features and the learned effective features. From this perspective, representation learning is manifest: the effective features $\phi^{\mathrm{E}}_{ij}(x_{\dot{\beta}};\theta^{\star})$ depend on the training data through the optimal parameters, $W^{\star}_{ij} = W^{\star}_{ij}(x_{\tilde{\alpha}}, y_{\tilde{\alpha}})$.

In summary, our nonlinear quadratic model (11.105) serves as a minimal model of representation learning. As we will see soon, this captures the mechanism of feature evolution for a nonzero but fixed dNTK.

Aside: Model Comparison of Linear Regression and Quadratic Regression

Before we move on to the dual sample-space description of the quadratic model, let's briefly perform a model comparison between linear regression and quadratic regression. In particular, let's think about the *model complexity* of these classes of models.[16]

[15] When doing nearly-linear quadratic regression practically, it would probably make the most sense to first find the optimal linear parameters W^{F}_{ij} and then plug them back into (11.118) to find the additional nonlinear parameters W^{I}_{ij}.

[16] For a further discussion of model complexity, with a direct focus on overparameterized deep learning models, see Epilogue ε.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

For both linear and quadratic regression, the number of model parameters is given by the number of elements in the combined weight matrix W_{ij}:

$$P \equiv n_{\text{out}} \times (n_f + 1). \tag{11.120}$$

Since both models completely memorize the same training set for a fixed and equal number of parameters, we obviously cannot naively use the Occam's razor heuristic (§6.2.2) for model comparison. This makes our model comparison somewhat subtle.

On the one hand, there is a sense in which the quadratic model is more complicated, as it computes far more functions of the input per parameter. Specifically, on a per parameter basis, we need to specify a far greater number of underlying functions for the quadratic model than we do for the linear model: i.e., we need

$$(n_f + 1) + \left[\frac{1}{2}(n_f + 1)(n_f + 2)\right] = O\left(P^2\right) \tag{11.121}$$

numbers to specify $\phi_j(x)$ *and* $\psi_{j_1 j_2}(x)$, while we need *just*

$$(n_f + 1) = O(P) \tag{11.122}$$

numbers to specify $\phi_j(x)$. In particular, the counting of the model functions is dominated by the meta feature functions $\psi_{j_1 j_2}(x)$. As such, this type of complexity is not really captured by the counting of model parameters, P; instead, it is expressed in the structure of the model, with the addition of the meta feature functions $\psi_{j_1 j_2}(x)$.

On the other hand, we can interpret these additional meta feature functions as *constraining* the quadratic model according to an explicit inductive bias for representation learning.[17] In particular, this additional structure alters the linear model solution (11.115) with the addition of $O(\epsilon)$ tunings W^{I}_{ij}, constrained by the $O\left(P^2\right)$ meta features that are defined before any learning takes place. Assuming these meta feature functions are *useful*, we might expect that the quadratic model will overfit less and generalize better.[18] (In fact, that was the whole point of introducing them.)

This latter point is worth a little further discussion. One typical signature of overfitting is that the parameters are extremely **finely-tuned**; these tunings are in some sense *unnatural* as they can arise from the extreme flexibility afforded to overparameterized models, enabling models to pass through all the training points *exactly*, to the extreme

[17] Similarly, we could naively think of the addition of a *regularization* term such as $\sum_{\mu=1}^{P} a_\mu \theta_\mu^2$ to the loss as making a model more complex with its extra structure, despite being a well-known remedy for overfitting. Instead, it's probably better to think of this regularization term as an inductive bias for *constraining* the norm squared of the optimized model parameters.

[18] Interestingly, for the quadratic model, the number of *effective feature functions* (11.107) is actually the same as the number of model parameters: $n_{\text{out}} \times (n_f + 1) = P$. Since it's only through these effective features that the meta feature functions enter the model predictions, cf. (11.119), this further underscores that, despite the additional model structure, there aren't actually $O\left(P^2\right)$ independent degrees of freedom that can be applied toward fitting the training data.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

detriment of the test predictions.[19] Adding a regularization term on the parameter norm – i.e., the one we just discussed in footnote 17 – combats such tuning: the additional constraints on the optimization problem drive the norm of the parameters toward zero, effectively promoting parameter sparsity. Here, we see that since the nonlinear contribution to the optimal weights, W^{I}_{ij}, is fixed to be small, $O(\epsilon)$, it's adding constraints that – if they're useful – can combat any fine tunings that may appear in the linear solution, W^{F}_{ij}, and lead to better generalization.

11.4.2 Nearly-Kernel Methods

Now that we have some more parameter-space intuition for the potential advantages of nonlinear models over linear models, let's now develop a *dual* sample-space description of quadratic regression where a quadratic-model analog of the dNTK appears naturally.

Starting with the expression in the second line of the prediction formula (11.119) and plugging in the free solution (11.115) and the interacting solution (11.118), we get

$$\begin{aligned} z_i(x_{\dot\beta};\theta^\star) = &\sum_{\tilde\alpha\in\mathcal{A}} y_{i;\tilde\alpha}\left[\sum_{j_1,j_2=0}^{n_f}\phi_{j_1}(x_{\tilde\alpha})\left(M^{-1}\right)_{j_1j_2}\phi_{j_2}(x_{\dot\beta})\right] \\ &+\frac{\epsilon}{2}\sum_{\tilde\alpha_1,\tilde\alpha_2\in\mathcal{A}} y_{i;\tilde\alpha_1}y_{i;\tilde\alpha_2}\sum_{j_1,j_2,j_3,j_4=0}^{n_f}\phi_{j_1}(x_{\tilde\alpha_1})\left(M^{-1}\right)_{j_1j_3}\phi_{j_2}(x_{\tilde\alpha_2})\left(M^{-1}\right)_{j_2j_4} \\ &\times\left[\psi_{j_3j_4}(x_{\dot\beta})-\sum_{\tilde\alpha\in\mathcal{A}}\psi_{j_3j_4}(x_{\tilde\alpha})\sum_{j_5,j_6=0}^{n_f}\phi_{j_5}(x_{\tilde\alpha})\left(M^{-1}\right)_{j_5j_6}\phi_{j_6}(x_{\dot\beta})\right]+O\left(\epsilon^2\right). \end{aligned} \tag{11.123}$$

[19] This is very commonly illustrated by using a polynomial basis of feature functions for linear regression, which is sometimes called *polynomial regression*. In this case, consider a linear model of a scalar function $f(x)$ with a scalar input x:

$$z(x_\delta;\theta)=\sum_{j=0}^{n_f} w_j\phi_j(x_\delta)\equiv w_0+w_1x_\delta+w_2x_\delta^2+\cdots+w_{n_f}x_\delta^{n_f}. \tag{11.124}$$

If there's any noise at all in the data, when the model is overparameterized, $n_f+1>N_{\mathcal{A}}$, the plot of this one-dimensional function will make $\sim n_f$ wild turns to go through the $N_{\mathcal{A}}$ training points. (This is particularly evocative if the target function is a simple linear function with noise, i.e., $f(x)=ax+b+\varepsilon$, with ε a zero-mean Gaussian noise with small variance $\sigma_\varepsilon^2\ll 1$.) In order to make these turns, the optimal coefficients, $w_j^\star$, computed by (11.115), will be finely-tuned to many significant figures. This kind of fine-tuning problem in model parameters is indicative of the model being unnatural or wrong; in fact, the analog of this problem in high-energy theoretical physics is called *naturalness* (see, e.g., [68] for a nontechnical discussion).

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

To simplify this expression, recall formula (10.130) that we derived when discussing *kernel methods*,

$$\sum_{j_1,j_2=0}^{n_f} \phi_{j_1}(x_{\tilde{\alpha}}) \left(M^{-1}\right)_{j_1 j_2} \phi_{j_2}(x_{\dot{\beta}}) = k_{\dot{\beta}\tilde{\alpha}_1} \widetilde{k}^{\tilde{\alpha}_1 \tilde{\alpha}}, \tag{11.125}$$

where the *kernel* was defined in (10.126) as

$$k_{\delta_1 \delta_2} \equiv k(x_{\delta_1}, x_{\delta_2}) \equiv \sum_{j=0}^{n_f} \phi_j(x_{\delta_1})\, \phi_j(x_{\delta_2}) \tag{11.126}$$

and provided a measure of similarity between two inputs $x_{i;\delta_1}$ and $x_{i;\delta_2}$ in feature space. Plugging this formula (11.125) back into our quadratic regression prediction formula (11.123), we get

$$\begin{aligned} z_i(x_{\dot{\beta}}; \theta^\star) = & \sum_{\tilde{\alpha}_1, \tilde{\alpha}_2 \in \mathcal{A}} k_{\dot{\beta}\tilde{\alpha}_1} \widetilde{k}^{\tilde{\alpha}_1 \tilde{\alpha}_2} y_{i;\tilde{\alpha}_2} \\ & + \frac{\epsilon}{2} \sum_{\tilde{\alpha}_1, \tilde{\alpha}_2 \in \mathcal{A}} y_{i;\tilde{\alpha}_1} y_{i;\tilde{\alpha}_2} \sum_{j_1, j_2, j_3, j_4 = 0}^{n_f} \phi_{j_1}(x_{\tilde{\alpha}_1}) \left(M^{-1}\right)_{j_1 j_3} \phi_{j_2}(x_{\tilde{\alpha}_2}) \left(M^{-1}\right)_{j_2 j_4} \\ & \qquad \times \left[\psi_{j_3 j_4}(x_{\dot{\beta}}) - \sum_{\tilde{\alpha}_1, \tilde{\alpha}_2 \in \mathcal{A}} k_{\dot{\beta}\tilde{\alpha}_1} \widetilde{k}^{\tilde{\alpha}_1 \tilde{\alpha}_2} \psi_{j_3 j_4}(x_{\tilde{\alpha}_2}) \right] + O\left(\epsilon^2\right), \end{aligned} \tag{11.127}$$

which is already starting to look a little better.

To simplify this expression further, we need to understand an object of the following form:

$$\sum_{j_1, j_2, j_3, j_4 = 0}^{n_f} \epsilon\, \phi_{j_1}(x_{\tilde{\alpha}_1}) \left(M^{-1}\right)_{j_1 j_3} \phi_{j_2}(x_{\tilde{\alpha}_2}) \left(M^{-1}\right)_{j_2 j_4} \psi_{j_3 j_4}(x_\delta). \tag{11.128}$$

Taking inspiration from the steps (10.129) that we took to derive our kernel-method formula (11.125), let's act on this object with two training-set kernels:

$$\begin{aligned} & \sum_{\tilde{\alpha}_1, \tilde{\alpha}_2 \in \mathcal{A}} \left[\sum_{j_1, j_2, j_3, j_4 = 0}^{n_f} \epsilon\, \phi_{j_1}(x_{\tilde{\alpha}_1}) \left(M^{-1}\right)_{j_1 j_3} \phi_{j_2}(x_{\tilde{\alpha}_2}) \left(M^{-1}\right)_{j_2 j_4} \psi_{j_3 j_4}(x_\delta) \right] \widetilde{k}_{\tilde{\alpha}_1 \tilde{\alpha}_3} \widetilde{k}_{\tilde{\alpha}_2 \tilde{\alpha}_4} \\ & \quad = \sum_{\tilde{\alpha}_1, \tilde{\alpha}_2 \in \mathcal{A}} \sum_{j_1, \ldots, j_6 = 0}^{n_f} \epsilon\, \phi_{j_1}(x_{\tilde{\alpha}_1}) \left(M^{-1}\right)_{j_1 j_3} \phi_{j_2}(x_{\tilde{\alpha}_2}) \left(M^{-1}\right)_{j_2 j_4} \\ & \qquad \times \psi_{j_3 j_4}(x_\delta) \phi_{j_5}(x_{\tilde{\alpha}_1}) \phi_{j_5}(x_{\tilde{\alpha}_3}) \phi_{j_6}(x_{\tilde{\alpha}_2}) \phi_{j_6}(x_{\tilde{\alpha}_4}) \end{aligned}$$

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

$$= \sum_{j_1,\ldots,j_6=0}^{n_f} \epsilon\, M_{j_1 j_5} \left(M^{-1}\right)_{j_1 j_3} M_{j_2 j_6} \left(M^{-1}\right)_{j_2 j_4} \psi_{j_3 j_4}(x_\delta)\phi_{j_5}(x_{\tilde{\alpha}_3})\phi_{j_6}(x_{\tilde{\alpha}_4})$$
$$= \sum_{j_1,j_2=0}^{n_f} \epsilon\, \psi_{j_1 j_2}(x_\delta)\phi_{j_1}(x_{\tilde{\alpha}_3})\phi_{j_2}(x_{\tilde{\alpha}_4}). \tag{11.129}$$

Here on the second line, we used the definition of the kernel (11.126) to swap both kernels for feature functions, on the third line we used the definition of the symmetric matrix $M_{j_1 j_2}$, (11.113), to replace two pairs of feature functions, and on the final line we simply canceled these matrices against their inverses.

This last expression suggests that an important object worth defining is

$$\mu_{\delta_0\delta_1\delta_2} \equiv \sum_{j_1,j_2=0}^{n_f} \epsilon\, \psi_{j_1 j_2}(x_{\delta_0})\phi_{j_1}(x_{\delta_1})\phi_{j_2}(x_{\delta_2}), \tag{11.130}$$

which we will call the **meta kernel**.[20] Analogous to the kernel methods' kernel (11.126), the meta kernel is a parameter-independent tensor, symmetric under an exchange of its final two sample indices $\delta_1 \leftrightarrow \delta_2$, and given entirely in terms of the fixed feature and meta feature functions that define the model. One way to think about (11.130) is that for a fixed particular input, x_{δ_0}, the meta kernel computes a different feature-space inner product between the two other inputs, x_{δ_1} and x_{δ_2}. Note also that due to the inclusion of the small parameter ϵ into the definition of the meta kernel (11.130), we should think of $\mu_{\delta_0\delta_1\delta_2}$ as being parametrically small too.

With this definition, the relation (11.129) can now be succinctly summarized as

$$\sum_{j_1,j_2,j_3,j_4=0}^{n_f} \epsilon\, \phi_{j_1}(x_{\tilde{\alpha}_1}) \left(M^{-1}\right)_{j_1 j_3} \phi_{j_2}(x_{\tilde{\alpha}_2}) \left(M^{-1}\right)_{j_2 j_4} \psi_{j_3 j_4}(x_\delta) \tag{11.131}$$
$$= \sum_{\tilde{\alpha}_3,\tilde{\alpha}_4 \in \mathcal{A}} \mu_{\delta\tilde{\alpha}_3\tilde{\alpha}_4} \widetilde{k}^{\tilde{\alpha}_3\tilde{\alpha}_1} \widetilde{k}^{\tilde{\alpha}_4\tilde{\alpha}_2}.$$

Finally, plugging this simple relation back into (11.127), we get

$$z_i(x_{\dot{\beta}};\theta^\star) = \sum_{\tilde{\alpha}_1,\tilde{\alpha}_2 \in \mathcal{A}} k_{\dot{\beta}\tilde{\alpha}_1} \widetilde{k}^{\tilde{\alpha}_1\tilde{\alpha}_2} y_{i;\tilde{\alpha}_2} \tag{11.132}$$
$$+ \frac{1}{2} \sum_{\tilde{\alpha}_1,\ldots,\tilde{\alpha}_4 \in \mathcal{A}} \left[\mu_{\dot{\beta}\tilde{\alpha}_1\tilde{\alpha}_2} - \sum_{\tilde{\alpha}_5,\tilde{\alpha}_6 \in \mathcal{A}} k_{\dot{\beta}\tilde{\alpha}_5} \widetilde{k}^{\tilde{\alpha}_5\tilde{\alpha}_6} \mu_{\tilde{\alpha}_6\tilde{\alpha}_1\tilde{\alpha}_2}\right] \left(\widetilde{k}^{\tilde{\alpha}_1\tilde{\alpha}_3} y_{i;\tilde{\alpha}_3}\right)\left(\widetilde{k}^{\tilde{\alpha}_2\tilde{\alpha}_4} y_{i;\tilde{\alpha}_4}\right).$$

[20] An alternate name for this object is the *differential of the kernel*, which we would consider symbolizing as $\mu_{\delta_0\delta_1\delta_2} \to \mathrm{d}k_{\delta_0\delta_1\delta_2}$. This name-symbol pair highlights the connection we're about to make to finite-width networks, but is perhaps less general in the context of making a broader model of representation learning.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

When the prediction of a quadratic model is computed in this way, we'll hereby make it known as a *nearly-kernel machine* or a **nearly-kernel method**.[21]

Analogous to linear models, we again have two ways of thinking about the solution of our nonlinear quadratic model's predictions: on the one hand, we can use the optimal parameters (11.115) and (11.118) to make predictions (11.119); on the other hand, we can make *nearly-kernel predictions* using the formula (11.132), in which neither the features, the meta features, nor the model parameters appear. That is, we've successfully traded our feature-space quantities $\phi_j(x)$, $\psi_{j_1 j_2}(x)$, and $W^\star_{ij}$ for sample-space quantities $k_{\delta\tilde{\alpha}}$, $\mu_{\delta_0\tilde{\alpha}_1\tilde{\alpha}_2}$, and $y_{i;\tilde{\alpha}}$. As was the case before for kernel methods, this works because all the feature indices are contracted in our prediction formula (11.119), and so only combinations of the form $k_{\delta\tilde{\alpha}}$ and $\mu_{\delta_0\tilde{\alpha}_1\tilde{\alpha}_2}$ ever show up in the result and not the value of the features or meta features themselves.[22] This duality between the microscopic

[21] Unlike kernel methods, this solution actually depends on the details of the learning algorithm. For instance, if we had optimized the quadratic-regression loss (11.109) by *gradient descent* rather than by *direct optimization* (11.110), then we would have found instead (for zero initialization $W_{ij} = 0$)

$$\begin{aligned} z_i(x_{\dot\beta};\theta^\star) = &\sum_{\tilde\alpha_1,\tilde\alpha_2\in\mathcal{A}} k_{\dot\beta\tilde\alpha_1}\widetilde{k}^{\tilde\alpha_1\tilde\alpha_2} y_{i;\tilde\alpha_2} \\ &+ \sum_{\tilde\alpha_1,\ldots,\tilde\alpha_4\in\mathcal{A}} \left[\mu_{\tilde\alpha_1\dot\beta\tilde\alpha_2} - \sum_{\tilde\alpha_5,\tilde\alpha_6\in\mathcal{A}} k_{\dot\beta\tilde\alpha_5}\widetilde{k}^{\tilde\alpha_5\tilde\alpha_6}\mu_{\tilde\alpha_1\tilde\alpha_6\tilde\alpha_2}\right] Z_{\text{A}}^{\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3\tilde\alpha_4} y_{i;\tilde\alpha_3} y_{i;\tilde\alpha_4} \\ &+ \sum_{\tilde\alpha_1,\ldots,\tilde\alpha_4\in\mathcal{A}} \left[\mu_{\dot\beta\tilde\alpha_1\tilde\alpha_2} - \sum_{\tilde\alpha_5,\tilde\alpha_6\in\mathcal{A}} k_{\dot\beta\tilde\alpha_5}\widetilde{k}^{\tilde\alpha_5\tilde\alpha_6}\mu_{\tilde\alpha_6\tilde\alpha_1\tilde\alpha_2}\right] Z_{\text{B}}^{\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3\tilde\alpha_4} y_{i;\tilde\alpha_3} y_{i;\tilde\alpha_4} \end{aligned} \tag{11.133}$$

for our nearly-kernel methods prediction formula, where the *algorithm projectors* are given by

$$Z_{\text{A}}^{\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3\tilde\alpha_4} \equiv \widetilde{k}^{\tilde\alpha_1\tilde\alpha_3}\widetilde{k}^{\tilde\alpha_2\tilde\alpha_4} - \sum_{\tilde\alpha_5} \widetilde{k}^{\tilde\alpha_2\tilde\alpha_5} X_{\text{II}}^{\tilde\alpha_1\tilde\alpha_5\tilde\alpha_3\tilde\alpha_4}, \tag{11.134}$$

$$Z_{\text{B}}^{\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3\tilde\alpha_4} \equiv \widetilde{k}^{\tilde\alpha_1\tilde\alpha_3}\widetilde{k}^{\tilde\alpha_2\tilde\alpha_4} - \sum_{\tilde\alpha_5} \widetilde{k}^{\tilde\alpha_2\tilde\alpha_5} X_{\text{II}}^{\tilde\alpha_1\tilde\alpha_5\tilde\alpha_3\tilde\alpha_4} + \frac{\eta}{2} X_{\text{II}}^{\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3\tilde\alpha_4}, \tag{11.135}$$

with the tensor $X_{\text{II}}^{\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3\tilde\alpha_4}$ implicitly satisfying

$$\sum_{\tilde\alpha_3,\tilde\alpha_4\in\mathcal{A}} X_{\text{II}}^{\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3\tilde\alpha_4} \left(\widetilde{k}_{\tilde\alpha_3\tilde\alpha_5}\delta_{\tilde\alpha_4\tilde\alpha_6} + \delta_{\tilde\alpha_3\tilde\alpha_5}\widetilde{k}_{\tilde\alpha_4\tilde\alpha_6} - \eta\widetilde{k}_{\tilde\alpha_3\tilde\alpha_5}\widetilde{k}_{\tilde\alpha_4\tilde\alpha_6}\right) = \delta^{\tilde\alpha_1}_{\tilde\alpha_5}\delta^{\tilde\alpha_2}_{\tilde\alpha_6}, \tag{11.136}$$

and global learning rate η. The origin of this gradient-descent solution should be clear after you traverse through §∞.2.2. Such *algorithm dependence* is to be expected for a nonlinear overparameterized model and is an important characteristic of finite-width networks as well. However, for the rest of the section we will continue to analyze the direct optimization formula, (11.132), with $Z_{\text{A}}^{\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3\tilde\alpha_4} = 0$ and $Z_{\text{B}}^{\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3\tilde\alpha_4} = \widetilde{k}^{\tilde\alpha_1\tilde\alpha_3}\widetilde{k}^{\tilde\alpha_2\tilde\alpha_4}/2$.

[22] Just as we discussed for kernel methods in footnote 46 of §10, in some situations we expect that specifying and evaluating the meta kernel $\mu_{\delta_0\delta_1\delta_2}$ is much simpler than specifying and evaluating meta feature function $\psi_{j_1 j_2}(x)$. Although picking these out of thin air seems difficult, perhaps there are other inspired ways of determining these functions that don't require an underlying description in terms of neural networks. It would be interesting to determine the necessary and sufficient conditions for a general three-input function, $\mu(x_{\delta_0}, x_{\delta_1}, x_{\delta_2}) \equiv \mu_{\delta_0\delta_1\delta_2}$, to be a meta kernel.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

feature-space description of the model and a macroscopic sample-space description is another realization of the effective theory approach discussed in §0.1, and we will return to comment more broadly on this duality in Epilogue ε after we discuss the dynamics of finite-width networks in §∞.

Finally, as we saw before for kernel methods, the nearly-kernel prediction is computed by direct comparison with previously-seen examples. In this case, it has the same piece linear in the true outputs, proportional to $y_{i;\tilde{\alpha}_2}$, and also has a new piece that's quadratic in the true output across different training examples, proportional to $y_{i;\tilde{\alpha}_1} y_{i;\tilde{\alpha}_2}$. In this way, nearly-kernel methods are also *memory-based* methods that involve memorizing the entire training set.

Trained-Kernel Prediction

Even though these nearly-kernel methods are *very-nearly* kernel methods, there's a real qualitative difference between them due to the presence of interactions between the parameters. In the feature-space picture described in §11.4.1, this difference manifested itself in terms of the nontrivial feature learning for the effective features $\phi^{\mathrm{E}}_{ij}(x,\theta)$, as expressed in the last line of the quadratic model prediction formula (11.119). To better understand this from the dual sample-space picture, let's analogously define an **effective kernel**

$$k^{\mathrm{E}}_{ii;\delta_1\delta_2}(\theta) \equiv \sum_{j=0}^{n_f} \phi^{\mathrm{E}}_{ij}(x_{\delta_1};\theta)\, \phi^{\mathrm{E}}_{ij}(x_{\delta_2};\theta), \tag{11.137}$$

which measures a parameter-dependent similarity between two inputs x_{δ_1} and x_{δ_2} using our effective features (11.107). Interestingly, we see that the model actually gives a different effective kernel for each output component i.[23] Let's try to understand this a little better by evaluating the effective kernel at the end of training:

$$\begin{aligned} k^{\mathrm{E}}_{ii;\delta_1\delta_2}(\theta^\star) &\equiv \sum_{j=0}^{n_f} \phi^{\mathrm{E}}_{ij}(x_{\delta_1};\theta^\star)\, \phi^{\mathrm{E}}_{ij}(x_{\delta_2};\theta^\star) \\ &= \sum_{j=0}^{n_f} \phi_j(x_{\delta_1})\, \phi_j(x_{\delta_2}) + \sum_{j_1,j_2=0}^{n_f} W^{\mathrm{F}}_{ij_1} \left[\psi_{j_1 j_2}(x_{\delta_1})\phi_{j_2}(x_{\delta_2}) + \psi_{j_1 j_2}(x_{\delta_2})\phi_{j_2}(x_{\delta_1})\right] \\ &\quad + O\left(\epsilon^2\right) \\ &= k_{\delta_1\delta_2} + \sum_{\tilde{\alpha}_1,\tilde{\alpha}_2 \in \mathcal{A}} (\mu_{\delta_1\delta_2\tilde{\alpha}_1} + \mu_{\delta_2\delta_1\tilde{\alpha}_1}) \widetilde{k}^{\tilde{\alpha}_1\tilde{\alpha}_2} y_{i;\tilde{\alpha}_2} + O\left(\epsilon^2\right). \end{aligned} \tag{11.138}$$

[23] Here, the use of two i's in the subscript of the effective kernel to represent the output component is just our convention; we'll later require a version with off-diagonal components in the slightly-less minimal model (11.146).

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

To get this last result on the final line, we plugged in the free solution (11.115) and then secretly used the following relation:

$$\phi_{j_0}(x_{\tilde{\alpha}}) \left(M^{-1}\right)_{j_0 j_1} \psi_{j_1 j_2}(x_{\delta_1}) \phi_{j_2}(x_{\delta_2}) = \sum_{\tilde{\alpha}_1 \in \mathcal{A}} \mu_{\delta_1 \delta_2 \tilde{\alpha}_1} \widetilde{k}^{\tilde{\alpha}_1 \tilde{\alpha}}, \tag{11.139}$$

which can be derived with manipulations analogous to those that we used in (10.129) and (11.129).[24] Here, in (11.138) we see that the effective kernel is shifted from the kernel and includes a contribution proportional to the meta kernel as well as the true training outputs $y_{i;\tilde{\alpha}}$; this is what gives the effective kernel its output-component dependence.

Finally, let's define one more kernel:

$$k^{\sharp}_{ii;\delta_1\delta_2} \equiv \frac{1}{2}\left[k_{\delta_1\delta_2} + k^{\mathrm{E}}_{ii;\delta_1\delta_2}(\theta^\star)\right]. \tag{11.140}$$

This **trained kernel** averages between the simple kernel methods' kernel from the linear model and the learned nearly-kernel methods' effective kernel. Defining the inverse of the trained-kernel submatrix evaluated on the training set in the usual way,

$$\sum_{\tilde{\alpha}_2 \in \mathcal{A}} \widetilde{k}^{\sharp\, \tilde{\alpha}_1 \tilde{\alpha}_2}_{ii} \widetilde{k}^{\sharp}_{ii;\tilde{\alpha}_2 \tilde{\alpha}_3} = \delta^{\tilde{\alpha}_1}_{\tilde{\alpha}_3}, \tag{11.141}$$

the utility of this final formulation is that the nearly-kernel prediction formula (11.132) can now be compressed as

$$z_i(x_{\dot{\beta}}; \theta^\star) = \sum_{\tilde{\alpha}_1, \tilde{\alpha}_2 \in \mathcal{A}} k^{\sharp}_{ii;\dot{\beta}\tilde{\alpha}_1} \widetilde{k}^{\sharp\, \tilde{\alpha}_1 \tilde{\alpha}_2}_{ii} y_{i;\tilde{\alpha}_2} + O\left(\epsilon^2\right), \tag{11.142}$$

taking the form of a *kernel prediction*, but with the benefit of nontrivial feature evolution incorporated into the trained kernel.[25] This is how representation learning manifests itself in nearly-kernel methods.

Finally, note that in our minimal model of representation learning, there's no *wiring* or mixing among the n_{out} different output components: while the prediction $z_i(x_{\dot{\beta}}; \theta^\star)$ is quadratic in the true output $y_{i;\tilde{\alpha}}$ – most easily seen in (11.132) – it still only involves the i-th component. From the perspective of the **trained-kernel prediction**, (11.142), each output component i has a *different* trained kernel associated with its prediction, but the i-th prediction never depends on other true output components $y_{i';\tilde{\alpha}}$ with $i' \neq i$.

However, this lack of wiring is by our design; this representation-learning model is really intended to be *minimal*. To enable mixing of the output components, we'll have to slightly generalize the quadratic model. This we'll do next when we explain how finite-width networks can be described in this nearly-kernel methods framework.

[24] Note that if we had instead optimized the quadratic-regression loss, (11.109), using gradient descent, then the effective kernel at the end of training, $k^{\mathrm{E}}_{ii;\delta_1\delta_2}(\theta^\star)$, would have a different expression than the one above, (11.138), for direct optimization, cf. our discussion in footnote 21.

[25] To verify the formula, use the definition of the trained kernel, (11.140), then expand in the effective kernel using the Schwinger–Dyson equations (4.55) to evaluate the matrix inverse. The result should agree with the nearly-kernel prediction formula (11.132).

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

11.4.3 Finite-Width Networks as Nonlinear Models

While the discussion so far in this section has been somewhat disconnected from the deep learning framework, much of it should still feel pretty familiar to you. For instance, the formula for the effective kernel at the end of training, (11.138), seems like it could be related to the update to the NTK, (11.10), if we identify the meta kernel $\mu_{\delta_0\delta_1\delta_2}$ with the dNTK $\widehat{\mathrm{d}H}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2}$ and also make the previous identifications that we made in §10.4.3 between the kernel methods' kernel and the NTK. Let's now make these connections between finite-width networks and nonlinear models more precise.

To start, for neural networks, let us define an analog of the effective feature function $\phi^{\mathrm{E}}_{ij}(x_\delta;\theta)$ (11.107) by

$$\phi^{\mathrm{E}}_{i,\mu}(x_\delta;\theta) \equiv \frac{dz^{(L)}_{i;\delta}}{d\theta_\mu}. \tag{11.143}$$

Note that for the linear model description of infinite-width networks, the derivative of the model output is a constant, and these features are completely *fixed* throughout training. In contrast, for quadratic models and finite-width networks, the derivative (11.143) is not constant, and so these effective features evolve throughout training as the model parameters move. As for the function approximator itself, after a small change in the parameters $\theta \to \theta + d\theta$, the network output evolves as

$$\begin{aligned} z^{(L)}_{i;\delta}(\theta + d\theta) &= z^{(L)}_{i;\delta}(\theta) + \sum_{\mu=1}^{P} \frac{dz^{(L)}_{i;\delta}}{d\theta_\mu} d\theta_\mu + \frac{1}{2}\sum_{\mu,\nu=1}^{P} \frac{d^2 z^{(L)}_{i;\delta}}{d\theta_\mu d\theta_\nu} d\theta_\mu d\theta_\nu + \cdots, \\ &= z^{(L)}_{i;\delta}(\theta) + \sum_{\mu=1}^{P} \phi^{\mathrm{E}}_{i,\mu}(x_\delta;\theta)\, d\theta_\mu + \frac{\epsilon}{2}\sum_{\mu,\nu=1}^{P} \widehat{\psi}_{i,\mu\nu}(x_\delta)\, d\theta_\mu d\theta_\nu + \cdots, \end{aligned} \tag{11.144}$$

where we've additionally defined an analog of the meta feature function $\psi_{j_1 j_2}(x_\delta)$ for neural networks by

$$\epsilon\widehat{\psi}_{i,\mu\nu}(x_\delta) \equiv \frac{d^2 z^{(L)}_{i;\delta}}{d\theta_\mu d\theta_\nu}. \tag{11.145}$$

For this discussion, we truncated the "..." in (11.144) so that the update to the output is exactly quadratic in the small change in the parameters. With this truncation, the update (11.144) for a finite-width neural network is identical to the update equation (11.106) that we found for our quadratic model after taking a small step.[26]

Let us further note that for the linear model description of infinite-width networks, the meta feature functions (11.145) vanish identically – as any linear function has a zero

[26] Considering the definition of our quadratic model, (11.105), we have included the small parameter ϵ as part of our identification. For MLPs, this parameter will be set automatically by the architecture and is given by the effective theory cutoff, the depth-to-width ratio of the network: $\epsilon \equiv L/n$. However, for such finite-width networks there are additional terms of order $\epsilon \equiv L/n$ that need to be incorporated in order to have a consistent description, as we will explain soon.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

second derivative – and thus have no effect on the dynamics. For finite-width networks with a quadratic truncation, these meta features (11.145) are parametrically small but no longer zero; they are stochastically sampled at initialization and then fixed over the course of training, hence decorated with a hat. Therefore, at quadratic order we will call these meta feature functions, $\widehat{\psi}_{i,\mu\nu}(x)$, **random meta features**, just as we called the feature functions *random features* for infinite-width networks.

Having established a connection in the feature space, let us now establish a similar connection in the sample-space dual description. First, associated with the effective feature functions (11.143) is the analog of the effective kernel $k^{\mathrm{E}}_{ii;\delta_1\delta_2}(\theta)$ (11.137), defined by

$$k^{\mathrm{E}}_{i_1 i_2;\delta_1\delta_2}(\theta) = \sum_{\mu,\nu} \lambda_{\mu\nu}\, \phi^{\mathrm{E}}_{i_1,\mu}(x_{\delta_1};\theta)\phi^{\mathrm{E}}_{i_2,\nu}(x_{\delta_2};\theta) = \sum_{\mu,\nu} \lambda_{\mu\nu} \frac{dz^{(L)}_{i_1;\delta_1}}{d\theta_\mu}\frac{dz^{(L)}_{i_2;\delta_2}}{d\theta_\nu} \equiv H^{(L)}_{i_1 i_2;\delta_1\delta_2}(\theta). \tag{11.146}$$

Here, we used our more general definition of the kernel (10.139) to include the learning-rate tensor, and since the effective features (11.143) have a parameter dependence, in the final equality we used most general definition of the NTK, (7.17), and gave it a θ argument, $H^{(L)}_{i_1 i_2;\delta_1\delta_2}(\theta)$, to indicate its parameter dependence. In particular, if we evaluated the effective kernel at initialization, $\theta = \theta(t=0)$, in terms of the random features

$$\widehat{\phi}_{i,\mu}(x_\delta) \equiv \phi^{\mathrm{E}}_{i,\mu}\Big(x_\delta;\theta(t=0)\Big) = \left.\frac{dz^{(L)}_{i;\delta}}{d\theta_\mu}\right|_{\theta=\theta(t=0)}, \tag{11.147}$$

we'd just have the usual L-th-layer stochastic NTK at initialization (8.4):

$$\begin{aligned}\widehat{k}_{i_1 i_2;\delta_1\delta_2} \equiv k^{\mathrm{E}}_{i_1 i_2;\delta_1\delta_2}\Big(\theta(t=0)\Big) &= \sum_{\mu,\nu} \lambda_{\mu\nu}\, \widehat{\phi}_{i_1,\mu}(x_{\delta_1})\, \widehat{\phi}_{i_2,\nu}(x_{\delta_2}) \\ &= \sum_{\mu,\nu} \lambda_{\mu\nu} \left(\frac{dz^{(L)}_{i_1;\delta_1}}{d\theta_\mu}\frac{dz^{(L)}_{i_2;\delta_2}}{d\theta_\nu} \right)\Bigg|_{\theta=\theta(t=0)} \equiv \widehat{H}^{(L)}_{i_1 i_2;\delta_1\delta_2}.\end{aligned} \tag{11.148}$$

For infinite-width networks, this NTK doesn't evolve during training and is composed of *random features* at initialization (10.137). In contrast, as we saw in §11.1, for finite-width networks, the effective kernel (11.146) *does* evolve during training, just as the analogous effective kernel (11.137) did for the quadratic model.

Finally, analogously to the meta kernel for the quadratic model (11.130), we can form a meta kernel for finite-width networks from the random features (11.147) and the random meta features (11.145) as

$$\begin{aligned}\widehat{\mu}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2} &\equiv \sum_{\substack{\mu_1,\nu_1,\\ \mu_2,\nu_2}} \epsilon \lambda_{\mu_1\nu_1}\lambda_{\mu_2\nu_2} \widehat{\psi}_{i_0,\mu_1\mu_2}(x_{\delta_0})\, \widehat{\phi}_{i_1,\nu_1}(x_{\delta_1})\, \widehat{\phi}_{i_2,\nu_2}(x_{\delta_2}) \\ &= \sum_{\substack{\mu_1,\nu_1,\\ \mu_2,\nu_2}} \lambda_{\mu_1\nu_1}\lambda_{\mu_2\nu_2} \left(\frac{d^2 z^{(L)}_{i_0;\delta_0}}{d\theta_{\mu_1} d\theta_{\mu_2}} \frac{dz^{(L)}_{i_1;\delta_1}}{d\theta_{\nu_1}} \frac{dz^{(L)}_{i_2;\delta_2}}{d\theta_{\nu_2}} \right)\Bigg|_{\theta=\theta(t=0)} \equiv \widehat{\mathrm{d}H}^{(L)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2},\end{aligned} \tag{11.149}$$

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

where we slightly generalized our earlier definition of the meta kernel (11.130) with the inclusion of the learning-rate tensors.[27] Thus, we've now identified the random meta kernel (11.149) with the L-th-layer stochastic dNTK (11.8).

With all these connections established, there are three notable differences between our minimal quadratic model and finite-width neural networks.

First, as should be clear from the definitions of the random features and random meta features, (11.147) and (11.145), these functions are stochastic rather than designed: they are determined by the details of the neural network architecture and depend on the values of the randomly-sampled parameters at initialization. We might more generally call such a quadratic model of the form (11.105) with random functions $\widehat{\phi}_j(x)$ and $\widehat{\psi}_{j_1 j_2}(x)$ a **random meta feature model**, generalizing the notion of a *random feature model* that we discussed in conjunction with infinite-width networks and linear models in §10.4.3.

Second, as we discussed at the end of §11.4.2, the quadratic model (11.105) does not wire together different components of the true outputs from the training set when making nearly-kernel predictions (11.132) on test-set inputs. In contrast, we will show soon in §∞.2.3 that the finite-width network predictions do have this wiring property. This deficiency of the quadratic model was actually by design on our part in an effort to eliminate extra complications when working through our minimal model of representation learning. To include wiring in the quadratic model, we can generalize it slightly as

$$z_i(x_\delta;\theta) = \sum_{\mu=1}^{P} \theta_\mu \widehat{\phi}_{i,\mu}(x_\delta) + \frac{\epsilon}{2} \sum_{\mu,\nu=1}^{P} \theta_\mu \theta_\nu \widehat{\psi}_{i,\mu\nu}(x_\delta). \tag{11.150}$$

This slightly-less-minimal model will now allow a parameter θ_μ to connect to various different output components, as the feature functions and meta feature functions now also carry vectorial indices specifying an output component.[28]

Third, as we've mentioned throughout this chapter, the leading finite-width contributions to the update to the network output include $O(\eta^3)$ terms. To capture these effects, we need to deform our quadratic model (11.105) into a **cubic model**:

$$z_i(x_\delta;\theta) = \sum_{\mu=1}^{P} \theta_\mu \widehat{\phi}_{i,\mu}(x_\delta) + \frac{1}{2} \sum_{\mu,\nu=1}^{P} \theta_\mu \theta_\nu \widehat{\psi}_{i,\mu\nu}(x_\delta) + \frac{1}{6} \sum_{\mu,\nu,\rho=1}^{P} \theta_\mu \theta_\nu \theta_\rho \widehat{\Psi}_{i,\mu\nu\rho}(x_\delta). \tag{11.151}$$

Here, the random **meta-meta feature functions** are given by the third derivative of the network output,

$$\widehat{\Psi}_{i,\mu\nu\rho}(x_\delta) \equiv \frac{d^3 z^{(L)}_{i;\delta}}{d\theta_\mu d\theta_\nu d\theta_\rho}\,; \tag{11.152}$$

[27] Our slightly more general definition of the meta kernel here should be understood as analogous to the slightly more general definition of the kernel (10.139).

[28] Note that these feature functions may have constraints, cf. the explicit form of the random feature (10.138). These constraints end up causing the infinite-width model not to wire, while allowing to wire the predictions of any particular network at finite width. These constraints can be thought of as a type of *weight tying*.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

the addition of this cubic term will enable the meta features to effectively evolve as if they were described by a linear model, while in turn the features will effectively evolve as if they were described by a quadratic model.[29] In summary, for finite-width networks of depth $L > 1$, this less-minimal model, (11.151), is a consistent description, with random features (11.147), random meta features (11.145), and random meta-meta features (11.152).

Deep Learning: A Nonminimal Model of Representation Learning

Representation learning is a big part of what makes deep learning exciting. What our minimal model of representation learning has shown us is that we can actually decouple the analysis of the *learning* from the analysis of the *deep*: the simple quadratic model (11.105) exhibits nontrivial representation learning for general choices of feature functions $\phi_j(x)$ and meta feature functions $\psi_{jk}(x)$, or dually, of a kernel $k_{\delta_1\delta_2}$ and a meta kernel $\mu_{\delta_0\delta_1\delta_2}$. In particular, the meta kernel is what made learning features from the training data possible, and we hope that this broader class of representation-learning models will be of both theoretical and practical interest in their own right.

Of course, *deep* learning is a nonminimal model of representation learning, and the structure of these kernels and meta kernels *does* matter. Specifically, for deep neural networks, the statistics of these functions encoded in the joint preactivation–NTK–dNTK distribution $p\Big(z^{(L)}, \widehat{H}^{(L)}, \widehat{\mathrm{d}H}^{(L)} \Big| \mathcal{D}\Big)$ are controlled by the representation group flow recursions – cf. §4, §8, and §11.2 – the details of which are implicitly determined by the underlying architecture and hyperparameters. In particular, we can understand the importance of this RG flow by remembering there can be a vast improvement from selecting other architectures beyond MLPs when applying function approximation to specific domains or datasets: RG flow *is* the inductive bias of the deep learning architecture.[30]

[29] To make this connection precise, we must give the small parameter ϵ not in the cubic model definition (11.151), but instead in the statistics of the joint distribution, $p(\widehat{\phi}_{i,\mu}, \widehat{\psi}_{i,\mu\nu}, \widehat{\Psi}_{i,\mu\nu\rho})$, that controls the random meta-meta feature model. Schematically, the nontrivial combinations are the following:

$$\mathbb{E}\left[\widehat{\phi}^2\right] = O(1)\,, \qquad \mathbb{E}\left[\widehat{\psi}\,\widehat{\phi}^2 z\right] = O(\epsilon)\,, \qquad \mathbb{E}\left[\widehat{\Psi}\,\widehat{\phi}^3\right] = O(\epsilon)\,, \qquad \mathbb{E}\left[\widehat{\psi}^2\,\widehat{\phi}^2\right] = O(\epsilon)\,. \tag{11.153}$$

In the next chapter, we'll identify these combinations with the NTK, the dNTK, and (soon-to-be-revealed) two ddNTKs, respectively. Importantly, since all of these combinations are the same order in $\epsilon = L/n$, to describe finite-width networks self-consistently, we need to think of them as cubic models.

[30] Note that the formalism of nonlinear models and nearly-kernel methods that we outlined in this section should also describe these other deep learning architectures so long as they admit an expansion around an infinite-width (or infinite-channel or infinite-head) limit. In particular, everything we learned here about representation learning and the training dynamics can be carried over; the only difference is that we will have different functions $\phi_{i,\mu}(x)$ and $\psi_{i,\mu\nu}(x)$, leading to different kernels and meta kernels, $k_{i_1 i_2;\delta_1\delta_2}$ and $\mu_{i_0 i_1 i_2;\delta_0\delta_1\delta_2}$, that can be built up from a different set of recursions than the ones that we studied in this book.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press

Thus, even in the set of models that exhibit nontrivial representation learning, these choices – the initial features, meta features, and so on – are still really important.[31]

The full power of deep learning is likely due to the deep – i.e., the *representation group flow* induced by interactions between neurons in deep models of many iterated layers – working in conjunction with the learning – i.e., the *representation learning* induced by the nonlinear dynamical interactions present at finite width. The principles of deep learning theory presented in this book are precisely those that will let you analyze both of these irreducible basic elements in full generality.

[31] In Appendix B, we'll explore an aspect of this question directly by studying *residual networks*: these networks let us introduce a parameter that in a single network has an interpretation of trading off more layers of representation group flow against more effective realizations from the ensemble.

https://doi.org/10.1017/9781009023405.013 Published online by Cambridge University Press