∞

The End of Training

The job of a scientist is to listen carefully to nature, not to tell nature how to behave.

Freeman Dyson, explaining Richard Feynman's approach [69].

In this chapter, we'll finally finish our leading-order effective-theory analysis of finite-width networks and solve their training dynamics under gradient descent. In contrast to the infinite-width limit, for which the solution is independent of the training algorithm, the dynamics of such deep networks have a rich phenomenology that captures the different ways in which useful features may develop over the course of training. The solution to these training dynamics gives a first-principles description of the ensemble of fully-trained finite-width networks, realizing a main goal of the book.

Unfortunately, our job will be disrupted by two facts of nature: *(i)* in order to have a consistent description of training dynamics at order $1/n$, we'll need to incorporate two additional objects that arise in the Taylor expansion of the update to the network output to third order, the update to the NTK to second order, and the update to the dNTK to first order; and *(ii)* due to a lack of smoothness, we won't be able to describe the dynamics of `ReLU` networks or networks consisting of any of the other nonlinear activation functions from the scale-invariant universality class.

As for the first point, while the analysis of representation learning in the context of the quadratic model was illuminating, we've already telegraphed that it was insufficient to capture the particular details of finite-width networks. In particular, to leading order in $1/n$, there are two more NTK differentials, which we'll refer to as *ddNTKs*. Although it's straightforward, working out the stochastic forward equations, recursions, and effective theory for these ddNTKs is somewhat tedious and no longer has any pedagogical value. As such, we won't provide the details of our derivations – you've already seen these sets of manipulations three times before in §4–§5, §8–§9, and §11.2–§11.3, for the preactivations, NTK, and dNTK, respectively – instead we'll simply state the results, leaving the details for you as a kind of post-training test evaluation; after all, this is the end of your training as well.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

As for the second point, throughout the book we've had to use special methods in order to work out exceptional explanations for any nonsmooth activation function, such as the ReLU. In our minds, this extra work was justified by the ReLU's current privileged status as one of the most popular activation functions in practice. However, we have finally run out of tricks and will have to give up – for a reason that is simple to explain: our Taylor expansion in the global learning rate η will break down when applied to the dynamics of networks built with nonsmooth activation functions. Instead, we'll have to follow the direction of the community and begin thinking again about smoothed versions of the ReLU – though only the ones that permit a type of criticality – such as the GELU and the SWISH.

With both those disruptions to our work heard, in §∞.1 we'll present all the relevant results for the ddNTKs – we'll define them, we'll give their tensor decomposition, and we'll explain their scaling with width and depth – while hiding all the irrelevant details at the back of the chapter in §∞.3. If you've been paying attention, you'll not be shocked to hear that – when properly normalized – the ddNTKs scale as the effective theory cutoff: ℓ/n. This scaling indicates that we need to consider the joint statistics of the preactivation–NTK–dNTK–ddNTKs in order to understand the leading-order finite-width dynamics of deep MLPs. Importantly, these ddNTKs endow the dNTK with its own dynamics; from the parameter-space perspective of §11.4.1, this means that the *meta feature functions* of the model will now evolve.

With those results stated, in §∞.2 we'll return to our regularly scheduled pedagogy and, at long last, solve the training dynamics at finite width. After an initial false start following our infinite-width giant leap, first in §∞.2.1 we'll learn how to take a small step following an adjusted giant leap, giving us our first finite-width solution. Then in §∞.2.2, we'll analyze many many steps of vanilla gradient descent, giving us our second finite-width solution. The nonlinear dynamics at finite width ultimately lead to a dependence of the fully-trained solution on the training algorithm, and so the solutions derived in these two subsections actually exhibit meaningful differences.

In particular, the function approximation of a fully-trained finite-width network can be decomposed into a universal part, independent of the optimization details, and a set of *algorithm projectors*, whose functional form encodes the entire dependence of the solution on the training algorithm. These projectors provide a dual sample-space perspective on the learning algorithm, analogous to the relationship between the model parameters and the different kernels.

Accordingly, in §∞.2.3 we'll discuss how these projectors impact the solution, letting us understand the inductive bias of the *training dynamics* separately from the inductive bias of the *network architecture*. We'll also further analyze the predictions made by such fully-trained networks, considering the growing tradeoff between increased representation learning and increased instantiation-to-instantiation fluctuations with network depth.

While this is the final chapter of the main text, in a small epilogue following this chapter, Epilogue ε, we'll explore how to define model complexity for overparameterized networks from our effective theory's macroscopic perspective. Then in two appendices, we'll further touch on some topics that are outside the scope of our main line of inquiry. In Appendix A, we'll introduce the framework of information theory, which

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

will give us the tools we need in order to estimate the optimal aspect ratio that separates *effectively-deep* networks from *overly-deep* networks. In Appendix B, we'll apply our effective theory approach to learn about residual networks and see how they can be used to extend the range of effectively-deep networks to greater and greater depths.

∞.1 Two More Differentials

Who ordered that?

I. I. Rabi, quipping about the $O(1/n)$ ddNTKs.

One last time, let's expand the ℓ-th-layer preactivations after a parameter update, this time recording terms up to *third order*:

$$\begin{aligned} \text{đ}z_{i;\delta}^{(\ell)} &\equiv z_{i;\delta}^{(\ell)}(t=1) - z_{i;\delta}^{(\ell)}(t=0) \\ &= \sum_{\ell_1=1}^{\ell} \sum_{\mu} \frac{dz_{i;\delta}^{(\ell)}}{d\theta_{\mu}^{(\ell_1)}} \text{đ}\theta_{\mu}^{(\ell_1)} + \frac{1}{2} \sum_{\ell_1,\ell_2=1}^{\ell} \sum_{\mu_1,\mu_2} \frac{d^2 z_{i;\delta}^{(\ell)}}{d\theta_{\mu_1}^{(\ell_1)} d\theta_{\mu_2}^{(\ell_2)}} \text{đ}\theta_{\mu_1}^{(\ell_1)} \text{đ}\theta_{\mu_2}^{(\ell_2)} \\ &\quad + \frac{1}{6} \sum_{\ell_1,\ell_2,\ell_3=1}^{\ell} \sum_{\mu_1,\mu_2,\mu_3} \frac{d^3 z_{i;\delta}^{(\ell)}}{d\theta_{\mu_1}^{(\ell_1)} d\theta_{\mu_2}^{(\ell_2)} d\theta_{\mu_3}^{(\ell_3)}} \text{đ}\theta_{\mu_1}^{(\ell_1)} \text{đ}\theta_{\mu_2}^{(\ell_2)} \text{đ}\theta_{\mu_3}^{(\ell_3)} + \cdots . \end{aligned} \tag{∞.1}$$

For gradient descent, also recall that after use of the chain rule (11.3), the change in the ℓ_a-th-layer parameters of any particular network is given by (11.4),

$$\text{đ}\theta_{\mu}^{(\ell_a)} = -\eta \sum_{\nu} \lambda_{\mu\nu}^{(\ell_a)} \left(\sum_{j,k,\tilde{\alpha}} \frac{\partial \mathcal{L}_{\mathcal{A}}}{\partial z_{k;\tilde{\alpha}}^{(L)}} \frac{dz_{k;\tilde{\alpha}}^{(L)}}{dz_{j;\tilde{\alpha}}^{(\ell)}} \frac{dz_{j;\tilde{\alpha}}^{(\ell)}}{d\theta_{\nu}^{(\ell_a)}} \right) = -\eta \sum_{\nu,j,\tilde{\alpha}} \lambda_{\mu\nu}^{(\ell_a)} \epsilon_{j;\tilde{\alpha}}^{(\ell)} \frac{dz_{j;\tilde{\alpha}}^{(\ell)}}{d\theta_{\nu}^{(\ell_a)}}, \tag{∞.2}$$

where we've used our convention from §11.1 of explicitly specifying which layer each parameter comes from. Please also recall from there that the learning-rate tensor $\lambda_{\mu\nu}^{(\ell)}$ only connects the parameters within a given layer ℓ. In the above expression, ℓ is an intermediate layer such that $\ell_a \leq \ell$, and we also used our *ℓ-th-layer error factor* (11.5):

$$\epsilon_{j;\tilde{\alpha}}^{(\ell)} \equiv \sum_{k=1}^{n_L} \frac{\partial \mathcal{L}_{\mathcal{A}}}{\partial z_{k;\tilde{\alpha}}^{(L)}} \frac{dz_{k;\tilde{\alpha}}^{(L)}}{dz_{j;\tilde{\alpha}}^{(\ell)}} = \frac{d\mathcal{L}_{\mathcal{A}}}{dz_{j;\tilde{\alpha}}^{(\ell)}}. \tag{∞.3}$$

After substituting the parameter update (∞.2) back into the preactivation update (∞.1), you should be able to write it in the form

$$\begin{aligned} \text{đ}z_{i;\delta}^{(\ell)} &= -\eta \sum_{j,\tilde{\alpha}} \widehat{H}_{ij;\delta\tilde{\alpha}}^{(\ell)} \epsilon_{j;\tilde{\alpha}}^{(\ell)} + \frac{\eta^2}{2} \sum_{j_1,j_2,\tilde{\alpha}_1,\tilde{\alpha}_2} \widehat{\mathrm{d}H}_{ij_1j_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2}^{(\ell)} \epsilon_{j_1;\tilde{\alpha}_1}^{(\ell)} \epsilon_{j_2;\tilde{\alpha}_2}^{(\ell)} \\ &\quad - \frac{\eta^3}{6} \sum_{j_1,j_2,j_3,\tilde{\alpha}_1,\tilde{\alpha}_2,\tilde{\alpha}_3} \widehat{\mathrm{dd_I}H}_{ij_1j_2j_3;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}^{(\ell)} \epsilon_{j_1;\tilde{\alpha}_1}^{(\ell)} \epsilon_{j_2;\tilde{\alpha}_2}^{(\ell)} \epsilon_{j_3;\tilde{\alpha}_3}^{(\ell)} \\ &\quad + O\left(\eta^4\right), \end{aligned} \tag{∞.4}$$

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

where the first two terms we found in the last chapter (11.9), and the cubic term is new, with the *first* of the **ddNTKs** defined as

$$\widehat{\mathrm{dd_I}H}^{(\ell)}_{i_0 i_1 i_2 i_3;\delta_0\delta_1\delta_2\delta_3} \equiv \sum_{\ell_1,\ell_2,\ell_3=1}^{\ell} \sum_{\substack{\mu_1,\nu_1,\\ \mu_2,\nu_2,\\ \mu_3,\nu_3}} \lambda^{(\ell_1)}_{\mu_1\nu_1}\lambda^{(\ell_2)}_{\mu_2\nu_2}\lambda^{(\ell_3)}_{\mu_3\nu_3} \frac{d^3 z^{(\ell)}_{i_0;\delta_0}}{d\theta^{(\ell_1)}_{\mu_1} d\theta^{(\ell_2)}_{\mu_2} d\theta^{(\ell_3)}_{\mu_3}} \frac{dz^{(\ell)}_{i_1;\delta_1}}{d\theta^{(\ell_1)}_{\nu_1}} \frac{dz^{(\ell)}_{i_2;\delta_2}}{d\theta^{(\ell_2)}_{\nu_2}} \frac{dz^{(\ell)}_{i_3;\delta_3}}{d\theta^{(\ell_3)}_{\nu_3}}. \tag{∞.5}$$

As always, the hat on the ddNTK indicates that it's stochastic, depending on the particular realization of the model parameters at initialization. Also, similar to the dNTK, this ddNTK is totally symmetric in its second, third, and fourth paired sets of indices $(i_1,\delta_1) \leftrightarrow (i_2,\delta_2) \leftrightarrow (i_3,\delta_3)$, while the first neural-sample index (i_0,δ_0) is distinguished from the other three.

By expanding to order η^3, the update, (∞.4), is now *cubic* in error factors. Expanding to this order is necessary because the first ddNTK has statistics at initialization that are $O(1/n)$ and so needs to be included in our analysis. However, any higher-order terms in the update are subleading, so we may replace $O(\eta^4) = O(1/n^2)$ in this expression.

Just as we had to expand the update to the NTK to order η when we expanded the update to the preactivations to order η^2, we will now have to expand the update to the NTK to order η^2 for the dynamics with our cubic update (∞.4) to be consistent:

$$\begin{aligned} đH^{(\ell)}_{i_1 i_2;\delta_1\delta_2} &\equiv H^{(\ell)}_{i_1 i_2;\delta_1\delta_2}(t=1) - H^{(\ell)}_{i_1 i_2;\delta_1\delta_2}(t=0) \qquad (∞.6)\\ &= \sum_{\ell_1=1}^{\ell}\sum_{\mu_1} \frac{dH^{(\ell)}_{i_1 i_2;\delta_1\delta_2}}{d\theta^{(\ell_1)}_{\mu_1}} đ\theta^{(\ell_1)}_{\mu_1} + \sum_{\ell_1,\ell_2=1}^{\ell}\sum_{\mu_1,\mu_2} \frac{d^2 H^{(\ell)}_{i_1 i_2;\delta_1\delta_2}}{d\theta^{(\ell_1)}_{\mu_1} d\theta^{(\ell_2)}_{\mu_2}} đ\theta^{(\ell_1)}_{\mu_1} đ\theta^{(\ell_2)}_{\mu_2} + \cdots \\ &= -\eta \sum_{j,\tilde\alpha} \left(\widehat{\mathrm{d}H}^{(\ell)}_{i_1 i_2 j;\delta_1\delta_2\tilde\alpha} + \widehat{\mathrm{d}H}^{(\ell)}_{i_2 i_1 j;\delta_2\delta_1\tilde\alpha} \right) \epsilon^{(\ell)}_{j;\tilde\alpha} \\ &\quad + \frac{\eta^2}{2} \sum_{j_1,j_2,\tilde\alpha_1,\tilde\alpha_2} \left[\widehat{\mathrm{dd_I}H}^{(\ell)}_{i_1 i_2 j_1 j_2;\delta_1\delta_2\tilde\alpha_1\tilde\alpha_2} + \widehat{\mathrm{dd_I}H}^{(\ell)}_{i_2 i_1 j_1 j_2;\delta_2\delta_1\tilde\alpha_1\tilde\alpha_2} \right] \epsilon^{(\ell)}_{j_1;\tilde\alpha_1} \epsilon^{(\ell)}_{j_2;\tilde\alpha_2} \\ &\quad + \eta^2 \sum_{j_1,j_2,\tilde\alpha_1,\tilde\alpha_2} \widehat{\mathrm{dd_{II}}H}^{(\ell)}_{i_1 i_2 j_1 j_2;\delta_1\delta_2\tilde\alpha_1\tilde\alpha_2} \epsilon^{(\ell)}_{j_1;\tilde\alpha_1} \epsilon^{(\ell)}_{j_2;\tilde\alpha_2} + O\left(\eta^3\right). \end{aligned}$$

To go to the final equality, we substituted in our NTK definition (11.7) and our parameter update (∞.2), computed the derivatives, and then collected the terms. To do so, we identified the *second* of the **ddNTKs**, defined as

$$\widehat{\mathrm{dd_{II}}H}^{(\ell)}_{i_1 i_2 i_3 i_4;\delta_1\delta_2\delta_3\delta_4} \equiv \sum_{\ell_1,\ell_2,\ell_3=1}^{\ell} \sum_{\substack{\mu_1,\nu_1,\\ \mu_2,\nu_2,\\ \mu_3,\nu_3}} \lambda^{(\ell_1)}_{\mu_1\nu_1}\lambda^{(\ell_2)}_{\mu_2\nu_2}\lambda^{(\ell_3)}_{\mu_3\nu_3} \frac{d^2 z^{(\ell)}_{i_1;\delta_1}}{d\theta^{(\ell_1)}_{\mu_1} d\theta^{(\ell_3)}_{\mu_3}} \frac{d^2 z^{(\ell)}_{i_2;\delta_2}}{d\theta^{(\ell_2)}_{\mu_2} d\theta^{(\ell_3)}_{\nu_3}} \frac{dz^{(\ell)}_{i_3;\delta_3}}{d\theta^{(\ell_1)}_{\nu_1}} \frac{dz^{(\ell)}_{i_4;\delta_4}}{d\theta^{(\ell_2)}_{\nu_2}}. \tag{∞.7}$$

The hat on this ddNTK indicates that it's also stochastic at initialization, and we will soon detail that it also has $O(1/n)$ statistics at leading order. Finally, $\widehat{\mathrm{dd_{II}}H}_{i_1 i_2 i_3 i_4;\delta_1\delta_2\delta_3\delta_4}$

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

has a more constrained symmetry, only symmetric under a joint swap of the paired set of indices as $(i_1, \delta_1) \leftrightarrow (i_2, \delta_2)$ *and* $(i_3, \delta_4) \leftrightarrow (i_4, \delta_4)$. However, this means that we can also swap indices as

$$\sum_{j_1,j_2,\tilde{\alpha}_1,\tilde{\alpha}_2} \widehat{\mathrm{dd_{II}}H}^{(\ell)}_{i_1 i_2 j_1 j_2;\delta_1\delta_2\tilde{\alpha}_1\tilde{\alpha}_2} \epsilon^{(\ell)}_{j_1;\tilde{\alpha}_1} \epsilon^{(\ell)}_{j_2;\tilde{\alpha}_2} = \sum_{j_1,j_2,\tilde{\alpha}_1,\tilde{\alpha}_2} \widehat{\mathrm{dd_{II}}H}^{(\ell)}_{i_2 i_1 j_1 j_2;\delta_2\delta_1\tilde{\alpha}_1\tilde{\alpha}_2} \epsilon^{(\ell)}_{j_1;\tilde{\alpha}_1} \epsilon^{(\ell)}_{j_2;\tilde{\alpha}_2}, \tag{∞.8}$$

which we used to simplify the final line of the NTK update (∞.6).

Overall, this NTK update is now *quadratic* in error factors, making its dynamics coupled and nonlinear. Again, expanding to this order is necessary because both ddNTKs have statistics at initialization that are $O(1/n)$, and so both need to be included in our analysis. However, any higher-order terms in the NTK update are subleading, so in (∞.6) we may replace $O(\eta^3) = O(1/n^2)$.

Finally, consider the leading-order update to the dNTK:

$$\begin{aligned} đ\mathrm{d}H^{(\ell)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2} &\equiv \mathrm{d}H^{(\ell)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2}(t=1) - \mathrm{d}H^{(\ell)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2}(t=0) \\ &= \sum_{\ell_1=1}^{\ell} \sum_{\mu_1} \frac{d\mathrm{d}H^{(\ell)}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2}}{d\theta^{(\ell_1)}_{\mu_1}} đ\theta^{(\ell_1)}_{\mu_1} + \cdots \\ &= -\eta \sum_{j,\tilde{\alpha}} \Big(\widehat{\mathrm{dd_I}H}^{(\ell)}_{i_0 i_1 i_2 j;\delta_0\delta_1\delta_2\tilde{\alpha}} + \widehat{\mathrm{dd_{II}}H}^{(\ell)}_{i_0 i_1 i_2 j;\delta_0\delta_1\delta_2\tilde{\alpha}} + \widehat{\mathrm{dd_{II}}H}^{(\ell)}_{i_0 i_2 i_1 j;\delta_0\delta_2\delta_1\tilde{\alpha}} \Big) \epsilon^{(\ell)}_{j;\tilde{\alpha}} \\ &\quad + O\Big(\eta^2\Big) \,. \end{aligned} \tag{∞.9}$$

To go to the final equality, we substituted our dNTK definition (11.8) and parameter update (∞.2), took the derivative, and then collected all the terms using our ddNTK definitions, (∞.5) and (∞.7). The dNTK update is *linear* in error factors, and its dynamics will be the simplest. Finally, the higher-order terms in the dNTK update are subleading, so in (∞.9) we may replace $O(\eta^2) = O(1/n^2)$. We also would like to apologize for the $đ\mathrm{d}H$ notation, and we promise that we won't have to use it again.

The updates (∞.4), (∞.6), and (∞.9) comprise the complete set of finite-width updates at $O(1/n)$. Thus, to proceed further in our analysis, we'll need to work out the leading-order statistics of the ddNTKs.

ddNTK Statistics

To find the distribution of fully-trained finite-width networks, we need the joint distribution of the network output, the NTK, the dNTK, and the ddNTKs:

$$p\Big(z^{(L)},\, \widehat{H}^{(L)},\, \widehat{\mathrm{d}H}^{(L)},\, \widehat{\mathrm{dd_I}H}^{(L)},\, \widehat{\mathrm{dd_{II}}H}^{(L)} \Big| \mathcal{D} \Big) \,. \tag{∞.10}$$

Rather than working through the details here, we'll do it in our own private notebooks. You already have all the tools you need: you can follow §4, §8, and §11.2 for examples of how to use RG flow to work out the recursions; and you can follow §5, §9, and §11.3

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

for examples of how to work out the details of the effective theory at initialization after tuning to criticality. The full details of this distribution (∞.10) can be found at the end of the chapter in §∞.3. Here, we'll highlight the results that you need in order to understand our dynamical computations in the following section.

After working out the stochastic forward equations for both ddNTKs – feel free to flip forward to (∞.174) and (∞.175) if you're curious about them – we'll need to find recursions for its statistics. For these tensors, the leading-order statistics come from their means; their cross correlations and their variances are all subleading. When evaluating the mean of each of the ddNTKs, it will become convenient to decompose them into the following set of tensors with sample indices only:

$$\mathbb{E}\left[\widehat{\mathrm{dd_I}H}^{(\ell)}_{i_0 i_1 i_2 i_3;\delta_0\delta_1\delta_2\delta_3}\right] \equiv \frac{1}{n_{\ell-1}}\Big[\delta_{i_0 i_1}\delta_{i_2 i_3}R^{(\ell)}_{\delta_0\delta_1\delta_2\delta_3}+\delta_{i_0 i_2}\delta_{i_3 i_1}R^{(\ell)}_{\delta_0\delta_2\delta_3\delta_1}+\delta_{i_0 i_3}\delta_{i_1 i_2}R^{(\ell)}_{\delta_0\delta_3\delta_1\delta_2}\Big], \tag{∞.11}$$

$$\mathbb{E}\left[\widehat{\mathrm{dd_{II}}H}^{(\ell)}_{i_1 i_2 i_3 i_4;\delta_1\delta_2\delta_3\delta_4}\right] \equiv \frac{1}{n_{\ell-1}}\Big[\delta_{i_1 i_2}\delta_{i_3 i_4}S^{(\ell)}_{\delta_1\delta_2\delta_3\delta_4}+\delta_{i_1 i_3}\delta_{i_4 i_2}T^{(\ell)}_{\delta_1\delta_3\delta_4\delta_2}+\delta_{i_1 i_4}\delta_{i_2 i_3}U^{(\ell)}_{\delta_1\delta_4\delta_2\delta_3}\Big]. \tag{∞.12}$$

Thus, there are four new objects whose recursions we need to determine and whose scaling with depth we'll need to compute.

The ddNTKs both vanish in the first layer, just like the dNTK. Proceeding then from the second layer to deeper layers, after a bunch of tedious algebra and plenty of flipping around in the book to make various substitutions, you can find recursions for $R^{(\ell)}$, $S^{(\ell)}$, $T^{(\ell)}$, and $U^{(\ell)}$: (∞.177), (∞.179), (∞.180), and (∞.181), respectively. You can flip forward to take a look at them, but you probably won't want to.... Importantly, these recursions in conjunction with the decompositions (∞.11) and (∞.12) altogether demonstrate that both ddNTKs have nontrivial order-$1/n$ statistics:

$$\mathbb{E}\left[\widehat{\mathrm{dd_I}H}^{(\ell)}_{i_0 i_1 i_2 i_3;\delta_0\delta_1\delta_2\delta_3}\right] = O\left(\frac{1}{n}\right), \qquad \mathbb{E}\left[\widehat{\mathrm{dd_{II}}H}^{(\ell)}_{i_1 i_2 i_3 i_4;\delta_1\delta_2\delta_3\delta_4}\right] = O\left(\frac{1}{n}\right). \tag{∞.13}$$

Thus, as we've been forecasting, we must include them in our finite-width analysis of training dynamics.

Focusing on the single-input statistics, we again make the layer-independent choices $C_b^{(\ell)} = C_b$, $C_W^{(\ell)} = C_W$, and $n_1 = \cdots = n_{L-1} \equiv n$. Ignoring 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 four recursions (∞.177), (∞.179), (∞.180), and (∞.181) together reduce to a form that at least fits on a page:

$$\begin{aligned} R^{(\ell+1)} = {} & \left(\chi_\perp^{(\ell)}\right)^2 R^{(\ell)} \\ & + \lambda_W^{(\ell+1)} C_W \left\langle \sigma''\sigma'\sigma'\sigma \right\rangle_{K^{(\ell)}} \left(\Theta^{(\ell)}\right)^2 + C_W^2 \left\langle \sigma'''\sigma'\sigma'\sigma' \right\rangle_{K^{(\ell)}} \left(\Theta^{(\ell)}\right)^3 \\ & + \chi_\perp^{(\ell)} \left(\lambda_W^{(\ell+1)} \left\langle \sigma''\sigma \right\rangle_{K^{(\ell)}} + C_W \Theta^{(\ell)} \left\langle \sigma'''\sigma' \right\rangle_{K^{(\ell)}}\right) \left(B^{(\ell)} + P^{(\ell)}\right) \\ & + \chi_\perp^{(\ell)} \left(\lambda_W^{(\ell+1)} \left\langle \sigma'\sigma' \right\rangle_{K^{(\ell)}} + C_W \Theta^{(\ell)} \left\langle \sigma''\sigma'' \right\rangle_{K^{(\ell)}}\right) P^{(\ell)}, \end{aligned} \tag{∞.14}$$

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

$$S^{(\ell+1)} = \left(\chi_\perp^{(\ell)}\right)^2 S^{(\ell)} \tag{∞.15}$$
$$+ C_W \lambda_W^{(\ell+1)} \left\langle \sigma'\sigma'\sigma'\sigma' \right\rangle_{K^{(\ell)}} \left(\Theta^{(\ell)}\right)^2 + C_W^2 \left\langle \sigma''\sigma''\sigma'\sigma' \right\rangle_{K^{(\ell)}} \left(\Theta^{(\ell)}\right)^3$$
$$+ \chi_\perp^{(\ell)} \left[\lambda_W^{(\ell+1)} \left\langle \sigma'\sigma' \right\rangle_{K^{(\ell)}} + C_W \Theta^{(\ell)} \left\langle \sigma''\sigma'' \right\rangle_{K^{(\ell)}}\right] B^{(\ell)} ,$$

$$T^{(\ell+1)} = \left(\chi_\perp^{(\ell)}\right)^2 T^{(\ell)} \tag{∞.16}$$
$$+ 2C_W \lambda_W^{(\ell+1)} \left\langle \sigma''\sigma'\sigma'\sigma \right\rangle_{K^{(\ell)}} \left(\Theta^{(\ell)}\right)^2 + C_W^2 \left\langle \sigma''\sigma''\sigma'\sigma' \right\rangle_{K^{(\ell)}} \left(\Theta^{(\ell)}\right)^3$$
$$+ \left(\lambda_W^{(\ell+1)}\right)^2 \left\langle \sigma'\sigma'\sigma\sigma \right\rangle_{K^{(\ell)}} \Theta^{(\ell)}$$
$$+ \left(\lambda_W^{(\ell+1)} \left\langle z\sigma'\sigma \right\rangle_{K^{(\ell)}} + C_W \Theta^{(\ell)} \left\langle z\sigma''\sigma' \right\rangle_{K^{(\ell)}}\right)^2 \frac{F^{(\ell)}}{\left(K^{(\ell)}\right)^2}$$
$$+ 2\chi_\perp^{(\ell)} \left[\lambda_W^{(\ell+1)} \left(\left\langle \sigma''\sigma \right\rangle_{K^{(\ell)}} + \left\langle \sigma'\sigma' \right\rangle_{K^{(\ell)}}\right) + C_W \Theta^{(\ell)} \left(\left\langle \sigma'''\sigma' \right\rangle_{K^{(\ell)}} + \left\langle \sigma''\sigma'' \right\rangle_{K^{(\ell)}}\right)\right] Q^{(\ell)} ,$$

$$U^{(\ell+1)} = \left(\chi_\perp^{(\ell)}\right)^2 U^{(\ell)} + C_W^2 \left\langle \sigma''\sigma''\sigma'\sigma' \right\rangle_{K^{(\ell)}} \left(\Theta^{(\ell)}\right)^3 . \tag{∞.17}$$

Here, we also simplified these expressions by recalling the two susceptibilities, (5.50) and (5.51):

$$\chi_\parallel^{(\ell)} \equiv \frac{C_W}{K^{(\ell)}} \left\langle \sigma'\sigma z \right\rangle_{K^{(\ell)}} , \qquad \chi_\perp^{(\ell)} \equiv C_W \left\langle \sigma'\sigma' \right\rangle_{K^{(\ell)}} . \tag{∞.18}$$

ddNTK Scalings

Now, let's tune to criticality, and use our scaling ansatz (5.93) to find the critical exponents p_R, p_S, p_T, and p_U that describe the asymptotic depth scaling of $R^{(\ell)}$, $S^{(\ell)}$, $T^{(\ell)}$, and $U^{(\ell)}$, respectively.

To understand the relative size of these tensors controlling the means of the ddNTKs, we will again need to identify appropriate dimensionless ratios. Following the dimensional analysis logic from our dNTK discussion, cf. (11.63), let's look at our third-order update for preactivations (∞.4). Remembering again that we can only add terms that have the same dimensions, we see that

$$[z] = [\eta]\,[\epsilon]\,[\widehat{H}] = [\eta]^2\,[\epsilon]^2\,[\widehat{\mathrm{d}H}] = [\eta]^3\,[\epsilon]^3\,[\widehat{\mathrm{dd_I}H}] = [\eta]^3\,[\epsilon]^3\,[\widehat{\mathrm{dd_{II}}H}]. \tag{∞.19}$$

From the first equality, we see as before that $[\eta]\,[\epsilon] = [z]\,[\widehat{H}]^{-1}$, and so R, S, T, and U each have dimensions of NTK cubed:

$$[R] \equiv [\widehat{\mathrm{dd_I}H}] = [\widehat{H}]^3\,[z]^{-2} , \qquad [S] = [T] = [U] \equiv [\widehat{\mathrm{dd_{II}}H}] = [\widehat{H}]^3\,[z]^{-2} . \tag{∞.20}$$

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

This means that the proper dimensionless combinations for the ddNTKs are

$$\frac{R^{(\ell)}K^{(\ell)}}{n\left(\Theta^{(\ell)}\right)^3} \sim \frac{1}{n}\left(\frac{1}{\ell}\right)^{p_R+p_0-3p_\Theta}, \qquad \frac{S^{(\ell)}K^{(\ell)}}{n\left(\Theta^{(\ell)}\right)^3} \sim \frac{1}{n}\left(\frac{1}{\ell}\right)^{p_S+p_0-3p_\Theta}, \tag{∞.21}$$

$$\frac{T^{(\ell)}K^{(\ell)}}{n\left(\Theta^{(\ell)}\right)^3} \sim \frac{1}{n}\left(\frac{1}{\ell}\right)^{p_T+p_0-3p_\Theta}, \qquad \frac{U^{(\ell)}K^{(\ell)}}{n\left(\Theta^{(\ell)}\right)^3} \sim \frac{1}{n}\left(\frac{1}{\ell}\right)^{p_U+p_0-3p_\Theta}, \tag{∞.22}$$

where to get a normalized ratio, we multiplied by the kernel $K^{(\ell)}$ to account for the $[z]^{-2}$ and then divided by the three factors of the frozen NTK $\Theta^{(\ell)}$. Here, p_0 is the critical exponent for the kernel, and p_Θ is the critical exponent for the NTK.

Finally, as a brief aside, let us comment on one aspect of dimensionality that we've been ignoring but will soon become important. In particular, since the network output z is set to the true output y, they should really have the same dimensions:

$$[z] = [y]. \tag{∞.23}$$

However, while the leading depth scaling of the preactivations is given by the kernel,

$$\mathbb{E}\left[z^{(\ell)}z^{(\ell)}\right] = K^{(\ell)} \sim \left(\frac{1}{\ell}\right)^{p_0}, \tag{∞.24}$$

the true output y is fixed and doesn't scale with depth. When comparing the performance of networks of different depths, this suggests that it might be helpful to rescale the final network outputs as

$$z_{i;\delta} \to z_{i;\delta}\left(\frac{1}{L}\right)^{-p_0/2}, \tag{∞.25}$$

effectively fixing the scaling of the overall network output as $p_0 = 0$, or equivalently rescale the training outputs as

$$y_{i;\tilde{\alpha}} \to y_{i;\tilde{\alpha}}\left(\frac{1}{L}\right)^{p_0/2}. \tag{∞.26}$$

We will further see how this can affect the predictions of a fully-trained network on its test set in §∞.2.3.[1]

$K^\star = 0$ Universality Class

For the $K^\star = 0$ universality class, remember that we Taylor-expanded the activation function as

$$\sigma(z) = \sum_{p=0}^{\infty} \frac{\sigma_p}{p!} z^p, \tag{∞.27}$$

[1] For regression tasks, where we want to learn a vector of real numbers, this rescaling is appropriate. For classification tasks, where we want to learn a discrete probability distribution, we should rescale either the network outputs (before any softmax layer) or the raw output targets $y_{i;\delta}$ (again before any softmax layer) as in (10.37).

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

defined the following Taylor coefficient for convenience,

$$a_1 \equiv \left(\frac{\sigma_3}{\sigma_1}\right) + \frac{3}{4}\left(\frac{\sigma_2}{\sigma_1}\right)^2 , \tag{∞.28}$$

and required that all activation functions in this class satisfy $\sigma_0 = 0$ and $\sigma_1 \neq 0$.

To solve our single input ddNTK recursions, (∞.14)–(∞.17), you'll have to evaluate a few new Gaussian expectations, taking particular note of the fact that some of them now depend on the third derivative of the activation function. Finally, to tune to $K^\star = 0$ criticality (5.90), we need to set the initialization hyperparameters as $C_b = 0$ and $C_W = 1/\sigma_1^2$; to implement the learning rate equivalence principle, we need to set our training hyperparameters as (9.95),

$$\lambda_b^{(\ell)} = \widetilde{\lambda}_b \left(\frac{1}{\ell}\right)^{p_\perp} L^{p_\perp - 1} , \qquad \lambda_W^{(\ell)} = \widetilde{\lambda}_W \left(\frac{L}{\ell}\right)^{p_\perp - 1} . \tag{∞.29}$$

For simplicity, let us also focus on odd activation functions, such as `tanh`, for which importantly $\sigma_2 = 0$ and $p_\perp = 1$.

Inspecting the single-input ddNTK recursions (∞.14)–(∞.17), we see that they depend on Gaussian expectations of preactivations as well as our previous solutions for all the other objects that we've considered: the NTK variance, the NTK–preactivation cross correlation, and the dNTK–preactivation cross correlation. Substituting in the solutions for all these quantities as needed – you'll have to flip around to find them, though most were reprinted in §11.3.2 for the dNTK analysis – we can find solutions to all the single-input ddNTK recursions:

$$R^{(\ell)} = -\frac{\ell^2}{48}\left[3\widetilde{\lambda}_b + 4\frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)}\right]\left[\widetilde{\lambda}_b + \frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)}\right]^2 (-a_1) + \cdots , \tag{∞.30}$$

$$S^{(\ell)} = \frac{\ell^2}{12}\left[\widetilde{\lambda}_b + \frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)}\right]^2 \widetilde{\lambda}_W \sigma_1^2 + \cdots , \tag{∞.31}$$

$$T^{(\ell)} = \frac{\ell^2}{32}\left[\widetilde{\lambda}_b + \frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)}\right](-a_1)\widetilde{\lambda}_b^2 + \cdots , \tag{∞.32}$$

$$U^{(\ell)} = \frac{1}{2}\left[\widetilde{\lambda}_b + \frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)}\right]^3 (-a_1) + \cdots . \tag{∞.33}$$

From these, we can read off the critical exponents as $p_R = p_S = p_T = -2$, and $p_U = 0$, and we see that the dimensionless ratios are given by

$$\frac{R^{(\ell)} K^{(\ell)}}{n\left(\Theta^{(\ell)}\right)^3} = -\frac{1}{48}\left[\frac{3\widetilde{\lambda}_b + 4\widetilde{\lambda}_W \sigma_1^2/(-a_1)}{\widetilde{\lambda}_b + \widetilde{\lambda}_W \sigma_1^2/(-a_1)}\right]\frac{\ell}{n} + \cdots , \tag{∞.34}$$

$$\frac{S^{(\ell)} K^{(\ell)}}{n\left(\Theta^{(\ell)}\right)^3} = \frac{1}{12}\left[\frac{\widetilde{\lambda}_W \sigma_1^2/(-a_1)}{\widetilde{\lambda}_b + \widetilde{\lambda}_W \sigma_1^2/(-a_1)}\right]\frac{\ell}{n} + \cdots , \tag{∞.35}$$

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

$$\frac{T^{(\ell)}K^{(\ell)}}{n\left(\Theta^{(\ell)}\right)^3} = \frac{1}{32}\left[\frac{\widetilde{\lambda}_b}{\widetilde{\lambda}_b + \widetilde{\lambda}_W\sigma_1^2/(-a_1)}\right]^2 \frac{\ell}{n} + \cdots , \tag{∞.36}$$

$$\frac{U^{(\ell)}K^{(\ell)}}{n\left(\Theta^{(\ell)}\right)^3} = \frac{1}{2}\frac{1}{\ell n} + \cdots . \tag{∞.37}$$

This means that $R^{(\ell)}$, $S^{(\ell)}$, and $T^{(\ell)}$ all scale according to our leading effective theory cutoff as

$$p_R + p_0 - 3p_\Theta = -1 \qquad p_S + p_0 - 3p_\Theta = -1\,, \qquad p_T + p_0 - 3p_\Theta = -1. \tag{∞.38}$$

Thus, we see that the leading-order finite-width dynamics of $K^\star = 0$ activation functions have contributions from the first ddNTK, $\widehat{\mathrm{dd_I}H}$, via $R^{(\ell)}$, and from the second ddNTK, $\widehat{\mathrm{dd_{II}}H}$, via $S^{(\ell)}$ and $T^{(\ell)}$.[2]

Scale-Invariant Universality Class

Perhaps the most important fact to remember about nonlinear scale-invariant activation functions (2.13),

$$\sigma(z) = \begin{cases} a_+ z\,, & z \geq 0, \\ a_- z\,, & z < 0, \end{cases} \tag{∞.39}$$

with $a_+ \neq -a_-$, is that they are not smooth: their first derivative is a step function centered at the origin; their second derivative is a Dirac delta function, (2.32),

$$\sigma'(z) = \begin{cases} a_+\,, & z \geq 0, \\ a_-\,, & z < 0, \end{cases}\,, \qquad \sigma''(z) = (a_+ - a_-)\delta(z); \tag{∞.40}$$

and higher derivatives will involve derivatives of the Dirac delta function. Inspecting again the single-input ddNTK recursions (∞.14)–(∞.17), the kink in these activation functions and the presence of Gaussian expectations with up to three derivatives of $\sigma(z)$ should scare you, especially if you heeded our warning at the end of §5.5. In fact, if you formally try to evaluate some of these expectations, particularly $\langle \sigma''\sigma'' \rangle_K$ and others related to it via integration by parts, you'll find that they want to blow up, even if you use all the *magic tricks* from physics that you might have at your disposal for trying to make sense of divergent integrals.[3]

The divergence of these Gaussian correlators is actually telling us that there is something very wrong with our expansions for the updates to the preactivations (∞.4),

[2] If we relaxed the restriction for the activation function to be odd, we'd find the same scalings for $R^{(\ell)}$, $S^{(\ell)}$, and $T^{(\ell)}$ – though with different coefficients – and we'd find that $U^{(\ell)}$ was up by a factor of ℓ, but still subleading overall.

[3] We were somewhat lucky in §11.3.1 when analyzing the dNTK: all the higher-derivative Gaussian integrals that we needed simplified via integration by parts and gave finite – in fact, vanishing – answers, cf. (11.67) and (11.69).

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

the NTK (∞.6), and the dNTK (∞.9). In particular, the Taylor expansion in the global learning rate η breaks down for these nonsmooth activation functions and doesn't accurately describe how a network is updated. As a result, our approach for solving the finite-width dynamics will not work for nonlinear scale-invariant activation functions.

To understand why, let's consider an extremely simple model of a network, a single neuron with a bias:

$$z(x) = \sigma(x + b). \tag{∞.41}$$

Here, the input x and the output z are both scalars, and for the activation function $\sigma(z)$ we'll pick the `ReLU`, with $a_+ = 1$ and $a_- = 0$. Accordingly, for a particular input x such that $x + b > 0$, the activation fires, and the output is $z(x) = x + b$; for a particular input x' such that $x' + b < 0$, the activation doesn't fire, and the output is $z(x') = 0$.

Now, let's consider a gradient-descent update to the parameters with a training example $x > -b$ such that the activation fires. Then, the bias updates as

$$đb = -\eta \frac{dz}{db}\epsilon = -\eta\epsilon = O(\eta)\,, \tag{∞.42}$$

where ϵ is the error factor of the loss, depending on the true output y. Now, if $x+b+đb > 0$, then the change in the output is

$$đz = -\eta\epsilon = O(\eta) \tag{∞.43}$$

and would be perfectly described by our Taylor expansion (∞.4). However, if the training error is large enough such that $x + b + đb = x + b - \eta\epsilon < 0$, then the activation turns off and

$$đz = -x - b = O(1)\,. \tag{∞.44}$$

Importantly, this update is independent of our expansion parameter η, and a Taylor expansion in η cannot detect this discontinuity at $\eta = (x + b)/\epsilon$.

Thus, for the `ReLU`, any time an activation crosses its firing threshold, it can contribute to the gradient-descent update in an η-independent way. Empirically, if you try to measure the NTK update for a deep MLP consisting of `ReLU` activation functions, you don't find anything consistent with the expansion (∞.6). Instead, for the appropriately normalized quantity, you'll find an $\sim 1/\sqrt{n}$ scaling with width and a linear scaling with depth ℓ, in contrast to the ℓ/n scaling expected from our perturbative formalism.[4] From this we reach the unfortunate conclusion that we'll have to give up on describing the finite-width dynamics of nonlinear scale-invariant activation functions using these methods.

Note that everything we have discussed for nonlinear scale-invariant activation functions with respect to criticality and the infinite-width training dynamics is perfectly

[4] It's temping to think that this $\sim 1/\sqrt{n}$ scaling arises from accumulating the probabilities that one of the Ln total hidden-layer $n \gg 1$ activations experiences an η-independent $O(1)$ change after the gradient-descent step.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

fine and experimentally validated, and the presence of a nonzero dNTK at finite width is still indicative of a dynamical NTK and representation learning at finite width. The problem we just described only affects the analysis we're going to perform in the following section for the training dynamics of finite-width networks, and the breakdown of the Taylor expansion just means that we will be unable to give a quantitative picture of representation learning for these nonsmooth activation functions.[5] So, if you do want to understand this, you'll probably need an entirely new approach.

This leaves us one activation function in the entire scale-invariant universality class: the `linear` activation function used for deep linear networks. Tuning to criticality, $C_b = 0$ and $C_W = 1$, which fixes $\chi = 1$, and choosing layer-independent learning rates (9.94) as

$$\lambda_b^{(\ell)} = \frac{\widetilde{\lambda}_b}{L}\,, \qquad \lambda_W^{(\ell)} = \frac{\widetilde{\lambda}_W}{L}, \tag{∞.45}$$

we can solve the single-input ddNTK recursions (∞.14)–(∞.17). Note that for every term in the U-recursion, (∞.17), and for every term but one in the R-recursion, (∞.14), the Gaussian expectations of activations involve second derivatives or third derivatives, which vanish for a `linear` function. For the one term in the R-recursion that does not vanish, it is also multiplied by $P^{(\ell)}$, which vanishes for all scale-invariant activations, cf. (11.74). This means that

$$R^{(\ell)} = 0\,, \qquad U^{(\ell)} = 0 \tag{∞.46}$$

and in particular that the first ddNTK, $\widehat{\mathrm{dd}_1 H}$, does not contribute to the deep linear network's dynamics at $O(1/n)$ since it's entirely determined by $R^{(\ell)}$. However, for the S-recursion, (∞.15), and the T-recursion, (∞.16), we find something nontrivial: these recursions can be exactly solved as

$$S^{(\ell)} = \frac{\ell^2(\ell^2-1)}{12L^3}(\widetilde{\lambda}_b + \widetilde{\lambda}_W K^\star)^2 \widetilde{\lambda}_W, \tag{∞.47}$$

$$T^{(\ell)} = \frac{\ell^2(\ell^2-1)}{12L^3}(\widetilde{\lambda}_b + \widetilde{\lambda}_W K^\star)\widetilde{\lambda}_W^2 K^\star, \tag{∞.48}$$

from which we see that the critical exponents are $p_S = p_T = -4$. Finally, the dimensionless ratios are

$$\frac{S^{(\ell)} K^{(\ell)}}{n\left(\Theta^{(\ell)}\right)^3} = \frac{1}{12}\left[\frac{\widetilde{\lambda}_W K^\star}{\widetilde{\lambda}_b + \widetilde{\lambda}_W K^\star}\right]\frac{\ell}{n} + \cdots\,, \tag{∞.49}$$

$$\frac{T^{(\ell)} K^{(\ell)}}{n\left(\Theta^{(\ell)}\right)^3} = \frac{1}{12}\left[\frac{\widetilde{\lambda}_W K^\star}{\widetilde{\lambda}_b + \widetilde{\lambda}_W K^\star}\right]^2\frac{\ell}{n} + \cdots\,, \tag{∞.50}$$

[5] However, our analysis applies to any of the smoothed versions of the `ReLU` that permit a type of criticality, cf. our criticality discussion of the `SWISH` and `GELU` in §5.3.4. In particular, the training dynamics we'll work out in the next section describe these networks. These dynamical solutions, in conjunction with the output-layer solutions to all their associated recursions, will accurately characterize such fully-trained `ReLU`-like networks in practice.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

and we see that these scale according to our leading effective theory cutoff as

$$p_S + p_0 - 3p_\Theta = -1\,, \qquad p_T + p_0 - 3p_\Theta = -1. \tag{∞.51}$$

In conclusion, we see that the second ddNTK, $\widehat{\mathrm{dd}_{\mathrm{II}}H}$, contributes via $S^{(\ell)}$ and $T^{(\ell)}$ to the dynamics of deep linear networks at leading order.

∞.2 Training at Finite Width

Now that we understand the joint statistics of the preactivations, the NTK, the dNTK, and the ddNTKs, we have nearly all the tools we need in order to evaluate the distribution of *fully-trained* networks at finite width and nonzero depth. To see why, recall our finite-width expansion of the network output evolution (∞.4):

$$\begin{aligned} z^{(L)}_{i;\delta}(t=1) = z^{(L)}_{i;\delta} - \eta \sum_{j,\tilde{\alpha}} \widehat{H}^{(L)}_{ij;\delta\tilde{\alpha}} \epsilon^{(L)}_{j;\tilde{\alpha}} + \frac{\eta^2}{2} \sum_{j_1,j_2,\tilde{\alpha}_1,\tilde{\alpha}_2} \widehat{\mathrm{d}H}^{(L)}_{ij_1j_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2} \epsilon^{(L)}_{j_1;\tilde{\alpha}_1} \epsilon^{(L)}_{j_2;\tilde{\alpha}_2} \\ - \frac{\eta^3}{6} \sum_{j_1,j_2,j_3,\tilde{\alpha}_1,\tilde{\alpha}_2,\tilde{\alpha}_3} \widehat{\mathrm{dd}_{\mathrm{I}}H}^{(L)}_{ij_1j_2j_3;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3} \epsilon^{(\ell)}_{j_1;\tilde{\alpha}_1} \epsilon^{(\ell)}_{j_2;\tilde{\alpha}_2} \epsilon^{(\ell)}_{j_3;\tilde{\alpha}_3} + O\Big(\frac{1}{n^2}\Big)\,. \end{aligned} \tag{∞.52}$$

Importantly, all the quantities on the right-hand side of the network update (∞.52) – $z^{(L)}_{i;\delta}$, $\widehat{H}^{(L)}_{ij;\delta\tilde{\alpha}}$, $\epsilon^{(L)}_{j;\tilde{\alpha}}$, $\widehat{\mathrm{d}H}^{(L)}_{ij_1j_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2}$, and $\widehat{\mathrm{dd}_{\mathrm{I}}H}^{(\ell)}_{ij_1j_2j_3;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}$ – are evaluated at initialization and thus are determined completely by the statistics of the joint preactivation–NTK–dNTK–ddNTKs distribution,

$$p\Big(z^{(L)},\, \widehat{H}^{(L)},\, \widehat{\mathrm{d}H}^{(L)},\, \widehat{\mathrm{dd}_{\mathrm{I}}H}^{(L)},\, \widehat{\mathrm{dd}_{\mathrm{II}}H}^{(L)} \Big| \mathcal{D}\Big)\,, \tag{∞.53}$$

that we spent the majority of this book evaluating in detail. Accordingly – just as we did before at infinite width (§10) – we could use the joint distribution (∞.53) to compute the statistics of the fully-trained network outputs after the update (∞.52), *if* we tuned a single step of gradient descent for each realization of a network so that we landed on the minimum of the training loss $z^{(L)}_{i;\tilde{\alpha}}(t=1) = y_{i;\tilde{\alpha}}$.

More generally, *if* we trained the network with T steps of gradient descent such that

$$z^{(L)}_{i;\tilde{\alpha}}(t=T) = y_{i;\tilde{\alpha}}\,, \qquad \text{for all } \tilde{\alpha} \in \mathcal{A}, \tag{∞.54}$$

and if we determined how to express the output of such a fully-trained network for general inputs $\delta \in \mathcal{D}$ as a functional of our statistical variables at initialization,

$$z^{(L)}_{i;\delta}(T) \equiv \Big[z^{(L)}_{i;\delta}(t=T)\Big]\Big(z^{(L)},\, \widehat{H}^{(L)},\, \widehat{\mathrm{d}H}^{(L)},\, \widehat{\mathrm{dd}_{\mathrm{I}}H}^{(L)},\, \widehat{\mathrm{dd}_{\mathrm{II}}H}^{(L)}\Big)\,, \tag{∞.55}$$

then we could give an analytical expression for the distribution of *fully-trained network outputs*:

$$p\Big(z^{(L)}(T)\Big)\,. \tag{∞.56}$$

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

The equation (∞.54) is our *fully-trained condition*, and the distribution (∞.56) completely describes the ensemble of finite-width networks at the end of training. The theoretical understanding of this distribution is exactly the goal we set for ourselves at the beginning of this book in §0. The only thing left for us to work out is the functional (∞.55); to do so, we first need to figure out what kind of steps to take in order to fully train these finite-width networks.

Now, recall from §10.2.2 that in the infinite-width limit, the fully-trained network solution (∞.55) had an *algorithm independence*: the distribution at the end of training didn't depend on the details of the optimization algorithm, and thus we could perform our theoretical analysis with any algorithm we wanted. In contrast, at finite width, the fully-trained solution (∞.55) will have an **algorithm dependence**: different fully-trained solutions will make different test-set predictions depending on the details of the particular optimization algorithm used to train the network, even when holding fixed the statistical variables at initialization and their associated initialization and training hyperparameters. Encouragingly, the solutions will nonetheless take a universal form, with the nonuniversal details of the particular training algorithm captured by six projective tensors: cf. (∞.154). Thus, we will be able to very generally study the distribution of fully-trained networks at finite width by working out such solutions.

With that in mind, in this section we'll present fully-trained solutions for two different optimization algorithms. First, in §∞.2.1 we'll take *two* Newton-like steps in order to satisfy our fully-trained condition (∞.54). While practically infeasible for large training sets, this training algorithm is rich in pedagogical value, emphasizing the way in which a finite-width network needs to adapt its representation to the training data in order to minimize its training error. Then, in §∞.2.2 we'll analytically solve the dynamics of the vanilla gradient descent at order $1/n$ and obtain a slightly different ensemble of fully-trained finite-width networks. This algorithm is not only practically implementable but also quite often used to optimize real neural networks, and our corresponding solution is an actual theoretical description of such fully-trained networks. Together, these solutions will help us understand the ways in which the details of the optimization algorithm can affect the corresponding fully-trained solution. Finally, in §∞.2.3 we'll be able to generally analyze the predictions of these different fully-trained networks on novel examples from the test set.

Throughout this section, we will declutter the notation a bit by dropping the layer indices, since to understand training we only need to focus on the network output at layer $\ell = L$.

An Infinite-Width Giant Leap at Finite Width

Before we begin, let's first review the giant leap that we took in §10.2 at infinite width. From the finer-grained perspective of finite width, we'll see that our leap actually missed the minimum, exhibiting training errors of order $1/n$. However, our new eyes on this leap will be instructive, as they will help us see how we can correct for these errors and reduce the finite-width training error even further.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

Recall from §10.2 that, in order to fully train an infinite-width network in a single step, we needed to make a *second-order update* of the form

$$\text{đ}\theta_\mu = -\sum_{\nu,\tilde{\alpha}_1,\tilde{\alpha}_2,i} \eta\lambda_{\mu\nu}\kappa^{\tilde{\alpha}_1\tilde{\alpha}_2}\frac{dz_{i;\tilde{\alpha}_1}}{d\theta_\nu}(z_{i;\tilde{\alpha}_2}-y_{i;\tilde{\alpha}_2})\,, \tag{∞.57}$$

which we interpreted either *(i)* as a *generalized training algorithm* (10.19) optimizing the standard MSE loss (10.5), or *(ii)* as a standard (tensorial) gradient-descent step (7.11) optimizing a *generalized MSE loss* (10.22). Here, $\kappa^{\tilde{\alpha}_1\tilde{\alpha}_2}$ was called the *Newton tensor*, and – in the first understanding of (∞.57) – could be interpreted as allowing us to take anisotropic steps in training sample space.

With this type of parameter update, a finite-width network output will evolve as

$$\begin{aligned} &z_{i;\delta}(t=1) \\ &\quad= z_{i;\delta} - \eta\sum_{\tilde{\alpha}_1,\tilde{\alpha}_2} H_{\delta\tilde{\alpha}_1}\kappa^{\tilde{\alpha}_1\tilde{\alpha}_2}\left(z_{i;\tilde{\alpha}_2}-y_{i;\tilde{\alpha}_2}\right) - \eta\sum_{j,\tilde{\alpha}_1,\tilde{\alpha}_2}\widehat{\Delta H}_{ij;\delta\tilde{\alpha}_1}\kappa^{\tilde{\alpha}_1\tilde{\alpha}_2}\left(z_{j;\tilde{\alpha}_2}-y_{j;\tilde{\alpha}_2}\right) \\ &\qquad+\frac{\eta^2}{2}\sum_{j_1,j_2,\tilde{\alpha}_1,\tilde{\alpha}_2,\tilde{\alpha}_3,\tilde{\alpha}_4}\widehat{\mathrm{d}H}_{ij_1j_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2}\kappa^{\tilde{\alpha}_1\tilde{\alpha}_3}\kappa^{\tilde{\alpha}_2\tilde{\alpha}_4}(z_{j_1;\tilde{\alpha}_3}-y_{j_1;\tilde{\alpha}_3})(z_{j_2;\tilde{\alpha}_4}-y_{j_2;\tilde{\alpha}_4}) \\ &\qquad-\frac{\eta^3}{6}\sum_{j_1,j_2,j_3,\tilde{\alpha}_1,\ldots,\tilde{\alpha}_6}\widehat{\mathrm{dd_I}H}_{ij_1j_2j_3;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}\kappa^{\tilde{\alpha}_1\tilde{\alpha}_4}\kappa^{\tilde{\alpha}_2\tilde{\alpha}_5}\kappa^{\tilde{\alpha}_3\tilde{\alpha}_6} \\ &\qquad\qquad\times\left(z_{j_1;\tilde{\alpha}_4}-y_{j_1;\tilde{\alpha}_4}\right)\left(z_{j_2;\tilde{\alpha}_5}-y_{j_2;\tilde{\alpha}_5}\right)\left(z_{j_3;\tilde{\alpha}_6}-y_{j_3;\tilde{\alpha}_6}\right)+O\!\left(\frac{1}{n^2}\right)\,, \end{aligned} \tag{∞.58}$$

where here we've made our usual decomposition of the NTK into a mean and fluctuation:

$$\widehat{H}_{ij;\delta\tilde{\alpha}} \equiv \delta_{ij}H_{\delta\tilde{\alpha}} + \widehat{\Delta H}_{ij;\delta\tilde{\alpha}}. \tag{∞.59}$$

In terms of such an update, our fully-trained condition (∞.54) after a single step $T=1$,

$$z_{i;\tilde{\alpha}}(t=1) = y_{i;\tilde{\alpha}}\,, \qquad \text{for all } \tilde{\alpha}\in\mathcal{A}, \tag{∞.60}$$

can be written as

$$\begin{aligned} &z_{i;\tilde{\alpha}} - y_{i;\tilde{\alpha}} \\ &\quad= \eta\sum_{\tilde{\alpha}_1,\tilde{\alpha}_2} H_{\tilde{\alpha}\tilde{\alpha}_1}\kappa^{\tilde{\alpha}_1\tilde{\alpha}_2}\left(z_{i;\tilde{\alpha}_2}-y_{i;\tilde{\alpha}_2}\right) + \eta\sum_{j,\tilde{\alpha}_1,\tilde{\alpha}_2}\widehat{\Delta H}_{ij;\tilde{\alpha}\tilde{\alpha}_1}\kappa^{\tilde{\alpha}_1\tilde{\alpha}_2}\left(z_{j;\tilde{\alpha}_2}-y_{j;\tilde{\alpha}_2}\right) \\ &\qquad-\frac{\eta^2}{2}\sum_{j_1,j_2,\tilde{\alpha}_1,\tilde{\alpha}_2,\tilde{\alpha}_3,\tilde{\alpha}_4}\widehat{\mathrm{d}H}_{ij_1j_2;\tilde{\alpha}\tilde{\alpha}_1\tilde{\alpha}_2}\kappa^{\tilde{\alpha}_1\tilde{\alpha}_3}\kappa^{\tilde{\alpha}_2\tilde{\alpha}_4}\left(z_{j_1;\tilde{\alpha}_3}-y_{j_1;\tilde{\alpha}_3}\right)\left(z_{j_2;\tilde{\alpha}_4}-y_{j_2;\tilde{\alpha}_4}\right) \\ &\qquad+\frac{\eta^3}{6}\sum_{j_1,j_2,j_3,\tilde{\alpha}_1,\ldots,\tilde{\alpha}_6}\widehat{\mathrm{dd_I}H}_{ij_1j_2j_3;\tilde{\alpha}\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}\kappa^{\tilde{\alpha}_1\tilde{\alpha}_4}\kappa^{\tilde{\alpha}_2\tilde{\alpha}_5}\kappa^{\tilde{\alpha}_3\tilde{\alpha}_6} \\ &\qquad\qquad\times\left(z_{j_1;\tilde{\alpha}_4}-y_{j_1;\tilde{\alpha}_4}\right)\left(z_{j_2;\tilde{\alpha}_5}-y_{j_2;\tilde{\alpha}_5}\right)\left(z_{j_3;\tilde{\alpha}_6}-y_{j_3;\tilde{\alpha}_6}\right)+O\!\left(\frac{1}{n^2}\right). \end{aligned} \tag{∞.61}$$

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

This new giant-leap condition (∞.61) perhaps seems a little daunting, and further it's not obvious that there's any particular choice of Newton tensor $\kappa^{\tilde{\alpha}_1\tilde{\alpha}_2}$ that can land us on the minimum. Nonetheless, we do expect that our infinite-width solution should be near the true finite-width solution, up to errors of order $O(1/n)$.[6]

With that in mind, as a first step let's try our infinite-width giant leap (10.27) and see where we land. This infinite-width giant leap had an interpretation as *Newton's method* and was given by the second-order update (∞.57), with a particular choice of the product of the global learning rate and the Newton tensor,

$$\eta\kappa^{\tilde{\alpha}_1\tilde{\alpha}_2} = \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_2}, \tag{∞.62}$$

where the *inverse* NTK mean submatrix $\widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_2}$ was defined implicitly via

$$\sum_{\tilde{\alpha}_2\in\mathcal{A}} \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_2}\,\widetilde{H}_{\tilde{\alpha}_2\tilde{\alpha}_3} = \delta^{\tilde{\alpha}_1}_{\tilde{\alpha}_3}. \tag{∞.63}$$

As always, the tilde on $\widetilde{H}_{\tilde{\alpha}_1\tilde{\alpha}_2}$ emphasizes that it's an $N_\mathcal{A}\times N_\mathcal{A}$-dimensional submatrix of the NTK mean evaluated on pairs of training inputs *only*. By now this distinction should be familiar enough that we will stop belaboring it.

More importantly, here we've used the inverse of the full NTK mean $\widetilde{H}_{\tilde{\alpha}_1\tilde{\alpha}_2}$ rather than the infinite-width frozen NTK $\widetilde{\Theta}_{\tilde{\alpha}_1\tilde{\alpha}_2}$. To explain why, let us recall from (9.3) that the NTK mean receives a series of corrections at each order in the $1/n$ expansion, of which the leading-order piece is the frozen NTK (9.4). Since we're now working at order $1/n$, we should in particular take into account the next-to-leading-order (NLO) $1/n$ correction to the NTK mean $H^{\{1\}}_{\tilde{\alpha}_1\tilde{\alpha}_2}$ by working with $\widetilde{H}_{\tilde{\alpha}_1\tilde{\alpha}_2}$ instead of $\widetilde{\Theta}_{\tilde{\alpha}_1\tilde{\alpha}_2}$.[7]

Substituting our infinite-width giant-leap Newton tensor (∞.62) into our giant-leap condition (∞.61) at finite width and rearranging, we get

$$\begin{aligned} 0 = &\sum_{j,\tilde{\alpha}_1,\tilde{\alpha}_2} \widehat{\Delta H}_{ij;\tilde{\alpha}\tilde{\alpha}_1}\widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_2}(z_{j;\tilde{\alpha}_1} - y_{j;\tilde{\alpha}_1}) \\ &-\frac{1}{2}\sum_{\substack{j_1,j_2,\\ \tilde{\alpha}_1,\ldots,\tilde{\alpha}_4}} \widehat{\mathrm{d}H}_{ij_1j_2;\tilde{\alpha}\tilde{\alpha}_1\tilde{\alpha}_2}\widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_3}\widetilde{H}^{\tilde{\alpha}_2\tilde{\alpha}_4}(z_{j_1;\tilde{\alpha}_3} - y_{j_1;\tilde{\alpha}_3})(z_{j_2;\tilde{\alpha}_4} - y_{j_2;\tilde{\alpha}_4}) \end{aligned} \tag{∞.64}$$

[6] For deep networks, we know that technically the corrections will be of order $O(L/n)$, corresponding to the cutoff scale of our effective theory description.

[7] While we won't show it explicitly, we expect that this *NLO NTK mean*, $H^{\{1\}(\ell)}_{\tilde{\alpha}_1\tilde{\alpha}_2}$, will have a solution that scales like $O(1/n)$ as compared to the frozen NTK; this would be analogous to what we found for the NLO metric $G^{\{1\}(\ell)}_{\tilde{\alpha}_1\tilde{\alpha}_2}$, cf. the discussion in §5.4 after (5.143). In particular, if we make a $1/n$ expansion for our training hyperparameters $\lambda_b^{(\ell)}$ and $\lambda_W^{(\ell)}$ as we did for our initialization hyperparameters in (5.138) and (5.139), then we will have extra freedom in the subleading hyperparameters $\lambda_b^{(\ell)\{1\}}$ and $\lambda_W^{(\ell)\{1\}}$ to eliminate the growing-in-ℓ contribution to $H^{\{1\}(\ell)}_{\tilde{\alpha}_1\tilde{\alpha}_2}$. Overall, this will make the NLO correction to the NTK mean scale as $O(1/n)$, subleading to the leading finite-width effects which scale as $O(L/n)$: in the language of RG flow, the NLO NTK mean is *marginal*. In practice, such a contribution is negligible and can thus be neglected for networks of any real depth.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

$$+\frac{1}{6}\sum_{\substack{j_1,j_2,j_3,\\ \tilde{\alpha}_1,\ldots,\tilde{\alpha}_6}} \widehat{\mathrm{dd_I} H}_{ij_1j_2j_3;\tilde{\alpha}\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3} \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_4} \widetilde{H}^{\tilde{\alpha}_2\tilde{\alpha}_5} \widetilde{H}^{\tilde{\alpha}_3\tilde{\alpha}_6} \times \left(z_{j_1;\tilde{\alpha}_4} - y_{j_1;\tilde{\alpha}_4}\right)\left(z_{j_2;\tilde{\alpha}_5} - y_{j_2;\tilde{\alpha}_5}\right)\left(z_{j_3;\tilde{\alpha}_6} - y_{j_3;\tilde{\alpha}_6}\right) + O\left(\frac{1}{n^2}\right).$$

Thus, we actually missed the minimum: for the network to be fully trained, the right-hand side of (∞.64) should have vanished, while here it is clearly nonzero in general. Taking a step back, it's clear now that what we actually found at infinite width in §10.2 was

$$z_{i;\tilde{\alpha}}(t=1) - y_{i;\tilde{\alpha}} = O\left(\frac{1}{n}\right). \tag{∞.65}$$

In other words, our networks were fully trained only at leading order, and (∞.64) gives an explicit expression for the $1/n$ correction.

Disentangling a little further, there are two such corrections at order $1/n$: the first correction – the first term on the right-hand side of (∞.64) – arises from the instantiation-to-instantiation fluctuations of the NTK across different realizations of the biases and weights at initialization; the second correction – comprised of the second and third terms on the right-hand side of (∞.64) – arises from nonlinear changes in the output as we take our step. In particular, this second correction is a *bona fide* manifestation of representation learning at finite width, accounting for the fact that the network's *effective features* are evolving. If we properly account for these two types of $1/n$ corrections, we should be able to attain a training error of order $1/n^2$:

$$z_{i;\tilde{\alpha}}(T) - y_{i;\tilde{\alpha}} = O\left(\frac{1}{n^2}\right). \tag{∞.66}$$

That is, we should be able to improve our effective theory of fully-trained networks, quantitatively by another multiplicative factor of $1/n$ and qualitatively by properly including representation learning into such an effective description.

Our first approach (§∞.2.1) to attain such effectively-zero training error, (∞.66), is to continue to engineer theoretical giant leaps so as to account for both the instantiation-to-instantiation fluctuations of the NTK and the effect of the dNTK and the first ddNTK.

Another approach (§∞.2.2) is to simply use the vanilla tensorial gradient descent algorithm as we do in practice; in that case, we will have to not only account for the dynamics of the network output but also account for the dynamics of the NTK and dNTK. After doing so, we will see that we can iteratively decrease the training loss to zero after many many such steps.

∞.2.1 A Small Step Following a Giant Leap

Here we'll train our networks with *two* second-order updates. For the first update, a giant leap, we'll need to further generalize our theoretical optimization algorithm in order to properly account for the instantiation-to-instantiation fluctuations of the NTK. In the

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

second update, a small step, we'll be able to account for the $1/n$ change in representation due to the nonzero dNTK and ddNTK, ultimately landing on the minimum of the loss as (∞.66). In particular, we can think of these updates as loosely corresponding to distinct phases of training that arise when implementing gradient-based training of neural networks in practice.

First Update: One Final Generalization of Gradient Decent

Please flip back to take a closer look at our unsatisfied condition for fully training our networks (∞.64). Right away, you should notice a serious problem in satisfying this constraint: the NTK fluctuation, $\widehat{\Delta H}_{ij;\tilde{\alpha}\tilde{\alpha}_1}$, the dNTK, $\widehat{\mathrm{d}H}_{ij_1j_2;\tilde{\alpha}\tilde{\alpha}_1\tilde{\alpha}_2}$, and the ddNTK, $\widehat{\mathrm{dd}_{\mathrm{I}}H}_{ij_1j_2j_3;\tilde{\alpha}\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}$, all mix different output components together in an update; i.e., the j-th component of the prediction error $z_{j;\tilde{\alpha}} - y_{j;\tilde{\alpha}}$ at initialization affects the i-th component of the output $z_{i;\delta}$ after the update, even for $i \neq j$. This stands in contrast to what we found for an infinite-width update in §10.1.1, where there was no such mixing or *wiring* of output components. Meanwhile, we did see a similar wiring effect for Bayesian inference at finite width, as we discussed in depth in §6.4.2.

While such wiring at finite width is fantastic from a practitioner's standpoint, it makes our theoretical work slightly more complicated. In particular, in order to satisfy the training constraint (∞.64), we will need to further generalize our second-order update (∞.57). Following in the footsteps of our previous two generalizations, (7.11) and (10.18), let's make one final generalization,

$$\eta \to \eta\lambda_{\mu\nu} \to \eta\lambda_{\mu\nu}\kappa^{\tilde{\alpha}_1\tilde{\alpha}_2} \to \eta\lambda_{\mu\nu}\kappa_{ij}^{\tilde{\alpha}_1\tilde{\alpha}_2}, \tag{∞.67}$$

with the final form of our theoretical update given by

$$\text{đ}\theta_\mu = -\sum_{\nu,\tilde{\alpha}_1,\tilde{\alpha}_2,i,j} \eta\lambda_{\mu\nu}\kappa_{ij}^{\tilde{\alpha}_1\tilde{\alpha}_2} \frac{dz_{i;\tilde{\alpha}_1}}{d\theta_\nu} \epsilon_{j;\tilde{\alpha}_2}. \tag{∞.68}$$

Here, we also introduced a further *generalized* Newton tensor $\kappa_{ij}^{\tilde{\alpha}_1\tilde{\alpha}_2}$ with output-component indices. This flexibility will allow us to resolve the mixing of the output components in the residual training error from our infinite-width leap (∞.64).[8]

[8] Similarly to the generalized MSE loss (10.22) discussed in §10.2.1, we can alternatively think of this further-generalized second-order update (∞.68) optimizing the standard MSE loss as instead arising from a standard first-order (tensorial) gradient descent update (7.11) on a further-generalized MSE loss,

$$\mathcal{L}_{\mathcal{A},\,\kappa}(\theta) \equiv \frac{1}{2}\sum_{i_1,i_2=1}^{n_L}\sum_{\tilde{\alpha}_1,\tilde{\alpha}_2\in\mathcal{A}} \kappa_{i_1i_2}^{\tilde{\alpha}_1\tilde{\alpha}_2}\left(z_{i_1;\tilde{\alpha}_1} - y_{i_1;\tilde{\alpha}_1}\right)\left(z_{i_2;\tilde{\alpha}_2} - y_{i_2;\tilde{\alpha}_2}\right), \tag{∞.69}$$

as long as the Newton tensor $\kappa_{i_1i_2}^{\tilde{\alpha}_1\tilde{\alpha}_2}$ is *also* assumed to be symmetric under the exchange of paired indices $(i_1,\tilde{\alpha}_1) \leftrightarrow (i_2,\tilde{\alpha}_2)$. This symmetry will in fact be present for both our first update, the giant leap, and our second update, the small step. With this interpretation (∞.69), our generalized Newton tensor $\kappa_{ij}^{\tilde{\alpha}_1\tilde{\alpha}_2}$ acts as a metric on sample space, through its sample indices $\tilde{\alpha}_1,\tilde{\alpha}_2$, and on output-component space, through its L-th-layer neural indices i,j.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

Note importantly that the μ, ν indices of $\lambda_{\mu\nu}$ are very different from the i, j indices of $\kappa_{ij}^{\tilde{\alpha}_1\tilde{\alpha}_2}$: the former each runs over all P parameters, while the latter each only runs over the n_L components of the network output. In particular, while learning-rate tensor $\lambda_{\mu\nu}$ lets us control how the gradient of the ν-th parameter affects the update to the μ-th parameter, the i, j indices of the generalized Newton tensor $\kappa_{ij}^{\tilde{\alpha}_1\tilde{\alpha}_2}$ instead control how the network's *features* (10.137) $dz_{i;\tilde{\alpha}_1}/d\theta_\nu$ are combined with the error factor $\epsilon_{j;\tilde{\alpha}_2}$ in order to make the update.[9] Allowing for a $\kappa_{ij}^{\tilde{\alpha}_1\tilde{\alpha}_2}$ with nonzero off-diagonal components in the i, j indices, we can precisely tune the *wiring* or mixing of output components that occurs in a particular update.

Finally, plugging our final-form second-order update (∞.68) back into the $1/n$ expansion of the network update (∞.4) and using the NTK, dNTK, and first ddNTK definitions, we can see how the network output changes after making an update with this new optimization algorithm:

$$
\begin{aligned}
&z_{i;\delta}(t=1) \qquad\qquad (\infty.70)\\
&\quad = z_{i;\delta} - \eta \sum_{j,\tilde{\alpha}_1,\tilde{\alpha}_2} H_{\delta\tilde{\alpha}_1} \kappa_{ij}^{\tilde{\alpha}_1\tilde{\alpha}_2} \left(z_{j;\tilde{\alpha}_2} - y_{j;\tilde{\alpha}_2}\right) - \eta \sum_{j,k,\tilde{\alpha}_1,\tilde{\alpha}_2} \widehat{\Delta H}_{ij;\delta\tilde{\alpha}_1} \kappa_{jk}^{\tilde{\alpha}_1\tilde{\alpha}_2} \left(z_{k;\tilde{\alpha}_2} - y_{k;\tilde{\alpha}_2}\right)\\
&\qquad + \frac{\eta^2}{2} \sum_{\substack{j_1,j_2,k_1,k_2,\\ \tilde{\alpha}_1,\tilde{\alpha}_2,\tilde{\alpha}_3,\tilde{\alpha}_4}} \widehat{\mathrm{d}H}_{ij_1j_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2} \kappa_{j_1k_1}^{\tilde{\alpha}_1\tilde{\alpha}_3} \kappa_{j_2k_2}^{\tilde{\alpha}_2\tilde{\alpha}_4} \left(z_{k_1;\tilde{\alpha}_3} - y_{k_1;\tilde{\alpha}_3}\right)\left(z_{k_2;\tilde{\alpha}_4} - y_{k_2;\tilde{\alpha}_4}\right)\\
&\qquad - \frac{\eta^3}{6} \sum_{\substack{j_1,j_2,j_3,k_1,k_2,k_3\\ \tilde{\alpha}_1,\tilde{\alpha}_2,\tilde{\alpha}_3,\tilde{\alpha}_4,\tilde{\alpha}_5,\tilde{\alpha}_6}} \widehat{\mathrm{dd_I}H}_{ij_1j_2j_3;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3} \kappa_{j_1k_1}^{\tilde{\alpha}_1\tilde{\alpha}_4} \kappa_{j_2k_2}^{\tilde{\alpha}_2\tilde{\alpha}_5} \kappa_{j_3k_3}^{\tilde{\alpha}_3\tilde{\alpha}_6}\\
&\qquad\qquad \times \left(z_{k_1;\tilde{\alpha}_4} - y_{k_1;\tilde{\alpha}_4}\right)\left(z_{k_2;\tilde{\alpha}_5} - y_{k_2;\tilde{\alpha}_5}\right)\left(z_{k_3;\tilde{\alpha}_6} - y_{k_3;\tilde{\alpha}_6}\right) + O\left(\frac{1}{n^2}\right).
\end{aligned}
$$

Here, we've again used the standard MSE loss, for which the error factor is given by the residual training error $\epsilon_{j;\tilde{\alpha}} = z_{j;\tilde{\alpha}} - y_{j;\tilde{\alpha}}$. Now, we'll need to pick our generalized Newton tensor $\kappa_{ij}^{\tilde{\alpha}_1\tilde{\alpha}_2}$ judiciously in order to make a first update that fully accounts for instantiation-to-instantiation fluctuations of particular networks in our ensemble.

Taking inspiration from our $1/n$-failed infinite-width Newton's step (∞.62), let's take a similar-looking first step according to

$$
\begin{aligned}
\eta\kappa_{ij}^{\tilde{\alpha}_1\tilde{\alpha}_2} &= \left(\widehat{H}^{-1}\right)_{ij}^{\tilde{\alpha}_1\tilde{\alpha}_2} \qquad\qquad (\infty.71)\\
&= \delta_{ij}\widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_2} - \sum_{\tilde{\alpha}_3,\tilde{\alpha}_4\in\mathcal{A}} \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_3} \widehat{\Delta H}_{ij;\tilde{\alpha}_3\tilde{\alpha}_4} \widetilde{H}^{\tilde{\alpha}_4\tilde{\alpha}_2}\\
&\qquad + \sum_{k=1}^{n_L} \sum_{\tilde{\alpha}_3,\ldots,\tilde{\alpha}_6\in\mathcal{A}} \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_3} \widehat{\Delta H}_{ik;\tilde{\alpha}_3\tilde{\alpha}_4} \widetilde{H}^{\tilde{\alpha}_4\tilde{\alpha}_5} \widehat{\Delta H}_{kj;\tilde{\alpha}_5\tilde{\alpha}_6} \widetilde{H}^{\tilde{\alpha}_6\tilde{\alpha}_2} + O\left(\Delta^3\right).
\end{aligned}
$$

[9] Note that when we developed our interpretation of an infinite-width network as a linear model in §10.4, we made a similar distinction between parameter indices μ and output-component indices i in (10.137) when defining the random feature functions $\widehat{\phi}_{i,\mu}(x)$. We also discussed how *wiring* can be incorporated into a nonminimal model of representation learning in §11.4.3.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

Here, we've introduced the complete inverse of the stochastic NTK sub-tensor evaluated on the training set, satisfying

$$\sum_{j,\tilde{\alpha}_2} \left(\widehat{H}^{-1}\right)_{ij}^{\tilde{\alpha}_1\tilde{\alpha}_2} \widehat{H}_{jk;\tilde{\alpha}_2\tilde{\alpha}_3} = \delta_{ik}\delta^{\tilde{\alpha}_1}_{\tilde{\alpha}_3}, \tag{∞.72}$$

and in the last equality of (∞.71), we've used the Schwinger–Dyson equations (4.55), which is a physicist's way of saying that we expanded the inverse of the stochastic NTK around the NTK mean.[10] The main difference between this new giant leap (∞.71) and our previous infinite-width giant leap (∞.62) is that we're now taking into account the instantiation-to-instantiation fluctuations of the NTK across different realizations of the model parameters; in other words, we're implementing a *different* Newton step for each *particular* network with its associated NTK $\widehat{H}_{i_1 i_2;\tilde{\alpha}_1\tilde{\alpha}_2}$. Accordingly, this step can be thought of as loosely corresponding to the first phase of training for such a particular network.

Taking this step, i.e., plugging this generalized Newton tensor (∞.71) into our update to the network output (∞.70), we find the training error decreases to

$$\begin{aligned} z_{i;\tilde{\alpha}}(t=1) - y_{i;\tilde{\alpha}} & \qquad (\infty.73) \\ &= \frac{1}{2}\sum_{\substack{j_1,j_2,\\ \tilde{\alpha}_1,\ldots,\tilde{\alpha}_4}} \widehat{\mathrm{d}H}_{ij_1j_2;\tilde{\alpha}\tilde{\alpha}_1\tilde{\alpha}_2}\widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_3}\widetilde{H}^{\tilde{\alpha}_2\tilde{\alpha}_4}(z_{j_1;\tilde{\alpha}_3} - y_{j_1;\tilde{\alpha}_3})(z_{j_2;\tilde{\alpha}_4} - y_{j_2;\tilde{\alpha}_4}) \\ &\quad - \frac{1}{6}\sum_{\substack{j_1,j_2,j_3,\\ \tilde{\alpha}_1,\ldots\tilde{\alpha}_6}} \widehat{\mathrm{dd_I}H}_{ij_1j_2j_3;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}\widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_4}\widetilde{H}^{\tilde{\alpha}_2\tilde{\alpha}_5}\widetilde{H}^{\tilde{\alpha}_3\tilde{\alpha}_6} \\ &\qquad\qquad \times \left(z_{j_1;\tilde{\alpha}_4} - y_{j_1;\tilde{\alpha}_4}\right)\left(z_{j_2;\tilde{\alpha}_5} - y_{j_2;\tilde{\alpha}_5}\right)\left(z_{j_3;\tilde{\alpha}_6} - y_{j_3;\tilde{\alpha}_6}\right) + O\left(\frac{1}{n^2}\right). \end{aligned}$$

Thus, we've correctly taken care of the first type of the $1/n$ corrections, and with this step we've reduced the residual prediction error on the training set from the order-one error of our initial prediction,

$$z_{i;\tilde{\alpha}}(t=0) - y_{i;\tilde{\alpha}} = O(1)\,, \tag{∞.74}$$

to a much smaller error of

$$z_{i;\tilde{\alpha}}(t=1) - y_{i;\tilde{\alpha}} = O\left(\frac{1}{n}\right), \tag{∞.75}$$

loosely corresponding to the empirically-large initial decrease of error when training networks in practice. Moreover, the rest of the error in (∞.73) is now entirely due to the

[10] Despite saying that we wouldn't belabor this any further, this is one of those unfortunate situations where we had to decide between decorating the inverse $(\widehat{H}^{-1})_{ij}^{\tilde{\alpha}_1\tilde{\alpha}_2}$ with either a hat or a tilde, and we went with the hat. Hopefully the tildes on the sample indices, e.g., $\tilde{\alpha}_1, \tilde{\alpha}_2$, will remind you that the inverse of this stochastic object is taken only with respect to the training set. Note that at no point will we ever need the inverse of the stochastic NTK evaluated on a general dataset $\mathcal{D}$.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

additional finite-width corrections encoded by the dNTK and first ddNTK, a consequence of representation learning.[11]

Second Update: Representation Learning Strikes Back

To reduce the training error further, we'll need to update our network again to further account for the fact that its representation evolved with the first update. In other words, we'll need to make a second gradient-descent update. If you'd like, you can imagine that this update corresponds to a second phase of training – following a first phase where the network significantly decreased its training error with the NTK evaluated at initialization – and so now the model must refine its features in order to further improve its performance.

Accordingly, since our training error is already down to $\sim 1/n$, our second update is going to be a lot smaller than our first. In particular, the update itself will necessarily only be of order $1/n$ so as to precisely cancel the remaining $1/n$ training error; that is, it's actually more of a *small step* than another *giant leap*.[12]

To determine which step we need to take, let's write the network output after a second update as

$$
\begin{aligned}
&z_{i;\delta}(t=2) \qquad\qquad (\infty.76)\\
&\quad = z_{i;\delta}(t=1) - \sum_{j,k,\tilde{\alpha}_1,\tilde{\alpha}_2} H_{ij;\delta\tilde{\alpha}_1}(t=1)\,\eta\kappa_{jk}^{\tilde{\alpha}_1\tilde{\alpha}_2}(t=1)\,[z_{k;\tilde{\alpha}_2}(t=1) - y_{k;\tilde{\alpha}_2}] + O\left(\frac{1}{n^2}\right)\\
&\quad = z_{i;\delta}(t=1) - \sum_{j,\tilde{\alpha}_1,\tilde{\alpha}_2} H_{\delta\tilde{\alpha}_1}\,\eta\kappa_{ij}^{\tilde{\alpha}_1\tilde{\alpha}_2}(t=1)\,[z_{j;\tilde{\alpha}_2}(t=1) - y_{j;\tilde{\alpha}_2}] + O\left(\frac{1}{n^2}\right)\,,
\end{aligned}
$$

where we left the second update's product of global learning rate and generalized Newton tensor $\eta\kappa_{ij}^{\tilde{\alpha}_1\tilde{\alpha}_2}(t=1)$ unspecified for now. Note here that in the first equality we dropped the d(d)NTK terms from the update: since after our first update (∞.73) the training error has already decreased to $O(1/n)$, the would-be dNTK term is very subleading $\sim \widehat{\mathrm{d}H} \times [\epsilon(t=1)]^2 = O(1/n^3)$, and the would-be ddNTK term is ridiculously subleading $\sim \widehat{\mathrm{dd_I}H} \times [\epsilon(t=1)]^3 = O(1/n^4)$. Similarly, on the third line we replaced the NTK after the first step $H_{ij;\delta\tilde{\alpha}_1}(t=1)$ by the NTK mean at initialization $\delta_{ij}H_{\delta\tilde{\alpha}_1}$. Given the $1/n$-suppressed training error after the first step, this substitution can be justified for these two reasons in conjunction: *(i)* the update to the NTK $\widehat{H}_{ij;\delta\tilde{\alpha}}(t=1) - \widehat{H}_{ij;\delta\tilde{\alpha}}(t=0)$ is itself suppressed by $1/n$, cf. (∞.6), so we may use the version from before the update,

[11] In particular, this first update (∞.71) would satisfy the training condition (∞.66) only if the NTK were constant under gradient descent. There's actually a name given to this type of phenomenon, *lazy training*, referring to situations when the network function behaves as if it is equal to a linearzation around the initial value of its parameters [70]. As we know from our discussion in §10.4, if the NTK is constant, then the network is a linear model.

[12] As we will see, the overall learning rate is essentially the same for both updates; the second update is only smaller because the gradient of the loss after the first update is itself much smaller when near a minimum.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

and *(ii)* the NTK fluctuation is also suppressed compared to its mean, so we may then swap the stochastic NTK at initialization for its mean. This means that if we make the following choice for our *small-step* second update,

$$\eta\kappa_{ij}^{\tilde{\alpha}_1\tilde{\alpha}_2}(t=1) = \delta_{ij}\widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_2} + O\left(\frac{1}{n}\right), \tag{∞.77}$$

then it's easy to see from (∞.76) that our network will now be fully trained as (∞.66):

$$z_{i;\tilde{\alpha}}(t=2) - y_{i;\tilde{\alpha}} = O\left(\frac{1}{n^2}\right). \tag{∞.78}$$

Thus, with our second update we were able to reduce the residual training error by an additional factor of $1/n$ as compared to the error after the first update (∞.73).[13]

For general inputs $\delta \in \mathcal{D}$, plugging our choice of learning rate and Newton tensor (∞.77) back into our second update (∞.76) and further re-expressing the network output after the first step $z_{i;\delta}(t=1)$ by (∞.70), with (∞.71) as our choice for the first step, we get

$$\begin{aligned}
&z_{i;\delta}(t=2) \qquad\qquad (\infty.79)\\
&= z_{i;\delta} - \sum_{\tilde{\alpha}_1,\tilde{\alpha}_2\in\mathcal{A}} H_{\delta\tilde{\alpha}_1}\widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_2}(z_{i;\tilde{\alpha}_2} - y_{i;\tilde{\alpha}_2})\\
&\quad + \sum_{j=1}^{n_L}\sum_{\tilde{\alpha}_1,\tilde{\alpha}_2\in\mathcal{A}}\left[\widehat{\Delta H}_{ij;\delta\tilde{\alpha}_1} - \sum_{\tilde{\alpha}_3,\tilde{\alpha}_4\in\mathcal{A}} H_{\delta\tilde{\alpha}_3}\widetilde{H}^{\tilde{\alpha}_3\tilde{\alpha}_4}\widehat{\Delta H}_{ij;\tilde{\alpha}_4\tilde{\alpha}_1}\right]\widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_2}(z_{j;\tilde{\alpha}_2} - y_{j;\tilde{\alpha}_2})\\
&\quad - \sum_{j,k=1}^{n_L}\sum_{\tilde{\alpha}_1,\ldots,\tilde{\alpha}_4\in\mathcal{A}}\left[\widehat{\Delta H}_{ij;\delta\tilde{\alpha}_1} - \sum_{\tilde{\alpha}_5,\tilde{\alpha}_6\in\mathcal{A}} H_{\delta\tilde{\alpha}_5}\widetilde{H}^{\tilde{\alpha}_5\tilde{\alpha}_6}\widehat{\Delta H}_{ij;\tilde{\alpha}_6\tilde{\alpha}_1}\right]\\
&\qquad\qquad \times \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_2}\widehat{\Delta H}_{jk;\tilde{\alpha}_2\tilde{\alpha}_3}\widetilde{H}^{\tilde{\alpha}_3\tilde{\alpha}_4}(z_{k;\tilde{\alpha}_4} - y_{k;\tilde{\alpha}_4})\\
&\quad + \frac{1}{2}\sum_{j_1,j_2=1}^{n_L}\sum_{\tilde{\alpha}_1,\ldots,\tilde{\alpha}_4\in\mathcal{A}}\left[\widehat{\mathrm{d}H}_{ij_1j_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2} - \sum_{\tilde{\alpha}_5,\tilde{\alpha}_6\in\mathcal{A}} H_{\delta\tilde{\alpha}_5}\widetilde{H}^{\tilde{\alpha}_5\tilde{\alpha}_6}\widehat{\mathrm{d}H}_{ij_1j_2;\tilde{\alpha}_6\tilde{\alpha}_1\tilde{\alpha}_2}\right]\\
&\qquad\qquad \times \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_3}\widetilde{H}^{\tilde{\alpha}_2\tilde{\alpha}_4}(z_{j_1;\tilde{\alpha}_3} - y_{j_1;\tilde{\alpha}_3})(z_{j_2;\tilde{\alpha}_4} - y_{j_2;\tilde{\alpha}_4})\\
&\quad - \frac{1}{6}\sum_{j_1,j_2,j_3=1}^{}\sum_{\tilde{\alpha}_1,\ldots,\tilde{\alpha}_6\in\mathcal{A}}\left[\widehat{\mathrm{dd_I}H}_{ij_1j_2j_3;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3} - \sum_{\tilde{\alpha}_7,\tilde{\alpha}_8\in\mathcal{A}} H_{\delta\tilde{\alpha}_7}\widetilde{H}^{\tilde{\alpha}_7\tilde{\alpha}_8}\widehat{\mathrm{dd_I}H}_{ij_1j_2j_3;\tilde{\alpha}_8\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}\right]\\
&\qquad\qquad \times \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_4}\widetilde{H}^{\tilde{\alpha}_2\tilde{\alpha}_5}\widetilde{H}^{\tilde{\alpha}_3\tilde{\alpha}_6}\left(z_{j_1;\tilde{\alpha}_4} - y_{j_1;\tilde{\alpha}_4}\right)\left(z_{j_2;\tilde{\alpha}_5} - y_{j_2;\tilde{\alpha}_5}\right)\left(z_{j_3;\tilde{\alpha}_6} - y_{j_3;\tilde{\alpha}_6}\right)\\
&\quad + O\left(\frac{1}{n^2}\right).
\end{aligned}$$

[13] This suggests that additional updates could continue to reduce the training error by additional factors of $1/n$. However, these further refinements – determined in terms of the higher-order corrections to our effective theory description – will be qualitatively the same as our leading finite-width description at $O(L/n)$. In other words, improving an infinite-width description to finite-width description incorporates representation learning, while more precise finite-width descriptions just allow the model to make further refinements to its features. Practically speaking, we expect our leading finite-width description to be very accurate for networks with reasonable values of the aspect ratio L/n.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

Here, we see that the expressions in the square brackets vanish identically for all the training inputs $\delta = \tilde{\alpha} \in \mathcal{A}$: our network is thus fully trained. In particular, since all of the variables on the right-hand side of (∞.79) are at initialization, this solution realizes our goal (∞.55) of expressing the output of a fully-trained network as a functional of such variables at initialization. Accordingly, the statistics of the fully-trained distribution (∞.56) can now be worked out from the joint preactivation–NTK–dNTK–ddNTKs distribution (∞.53) at initialization.[14]

While this is all very exciting, let us also caution you that these generalized second-order updates are probably best thought of as giving a simple theoretical model of a training algorithm – designed to let us understand the training process analytically – and by no means are we suggesting that they provide a good or useful option for practical optimization. Instead, the most practical algorithm for optimization is vanilla first-order gradient descent. As we've already pointed out that the output of a fully-trained network *does* depend on the details of the algorithm used for optimization, now we really have no choice left other than to explicitly analyze many many steps of gradient descent.

[14] Alternatively, we could have reached this same solution in a *single update* if we had instead made a *finely-tuned* giant leap, picking the generalized Newton tensor as

$$
\begin{aligned}
\eta\kappa_{ij}^{\tilde{\alpha}_1\tilde{\alpha}_2} = \delta_{ij}\widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_2} &- \sum_{\tilde{\alpha}_3,\tilde{\alpha}_4\in\mathcal{A}} \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_3}\widehat{\Delta H}_{ij;\tilde{\alpha}_3\tilde{\alpha}_4}\widetilde{H}^{\tilde{\alpha}_4\tilde{\alpha}_2} \\
&+ \sum_{k=1}^{n_L}\sum_{\tilde{\alpha}_3,\ldots,\tilde{\alpha}_6\in\mathcal{A}} \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_3}\widehat{\Delta H}_{ik;\tilde{\alpha}_3\tilde{\alpha}_4}\widetilde{H}^{\tilde{\alpha}_4\tilde{\alpha}_5}\widehat{\Delta H}_{kj;\tilde{\alpha}_5\tilde{\alpha}_6}\widetilde{H}^{\tilde{\alpha}_6\tilde{\alpha}_2} \\
&+ \frac{1}{2}\sum_{k=1}^{n_L}\sum_{\tilde{\alpha}_3,\ldots,\tilde{\alpha}_6\in\mathcal{A}} \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_3}\widetilde{H}^{\tilde{\alpha}_2\tilde{\alpha}_4}\widetilde{H}^{\tilde{\alpha}_5\tilde{\alpha}_6}\widehat{\mathrm{d}H}_{ijk;\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5}\left(z_{k;\tilde{\alpha}_6} - y_{k;\tilde{\alpha}_6}\right) \\
&- \frac{1}{6}\sum_{k_1,k_2=1}^{n_L}\sum_{\tilde{\alpha}_3,\ldots,\tilde{\alpha}_8\in\mathcal{A}} \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_3}\widetilde{H}^{\tilde{\alpha}_2\tilde{\alpha}_4}\widetilde{H}^{\tilde{\alpha}_5\tilde{\alpha}_7}\widetilde{H}^{\tilde{\alpha}_6\tilde{\alpha}_8}\widehat{\mathrm{dd_I}H}_{ijk_1k_2;\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6} \\
&\qquad\qquad \times \left(z_{k_1;\tilde{\alpha}_7} - y_{k_1;\tilde{\alpha}_7}\right)\left(z_{k_2;\tilde{\alpha}_8} - y_{k_2;\tilde{\alpha}_8}\right).
\end{aligned}
\tag{∞.80}
$$

In essence, the first three terms of (∞.80) come from inverting the stochastic NTK, corresponding to our first-update giant leap (∞.71) and account for the instantiation-to-instantiation fluctuations in the NTK. In contrast, the final two terms correspond to our second-update small step (∞.78) and account for the dNTK–ddNTK-induced representation learning. Note that this generalized Newton tensor (∞.80) is *asymmetric* under the exchange of paired indices $(i,\tilde{\alpha}_1) \leftrightarrow (j,\tilde{\alpha}_2)$ due to the last term, and thus this finely-tuned update doesn't admit an alternative interpretation of optimization with a further generalized loss (∞.69).

This single update is analogous to the *direct optimization* solution (11.132) for quadratic regression in §11.4. The very finely-tuned nature of this single-update algorithm (∞.80) suggests that it's easier to make the fine adjustments required to reach a solution by taking many simpler steps rather than fewer complicated steps. We will see the other side of this next in §∞.2.2 when we study training by many many steps of vanilla gradient descent.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

∞.2.2 Many Many Steps of Gradient Descent

In this extended subsection, we're going to study tensorial gradient descent (7.11) and optimize finite-width neural networks according to the MSE loss with a constant global learning rate η.[15] With this progenitor-of-all-other-gradient-based-learning-algorithms algorithm, the network output will evolve as (∞.4),

$$\begin{aligned} z_{i;\delta}(t+1) &= z_{i;\delta}(t) - \eta \sum_{j,\tilde{\alpha}} H_{ij;\delta\tilde{\alpha}}(t)\left[z_{j;\tilde{\alpha}}(t) - y_{j;\tilde{\alpha}}\right] \\ &\quad + \frac{\eta^2}{2} \sum_{j_1,j_2,\tilde{\alpha}_1,\tilde{\alpha}_2} \mathrm{d}H_{ij_1j_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2}(t)\left[z_{j_1;\tilde{\alpha}_1}(t) - y_{j_1;\tilde{\alpha}_1}\right]\left[z_{j_2;\tilde{\alpha}_2}(t) - y_{j_2;\tilde{\alpha}_2}\right] \\ &\quad - \frac{\eta^3}{6} \sum_{j_1,j_2,j_3,\tilde{\alpha}_1,\tilde{\alpha}_2,\tilde{\alpha}_3} \widehat{\mathrm{dd_I}H}_{ij_1j_2j_3;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3} \\ &\qquad\qquad \times \left[z_{j_1;\tilde{\alpha}_1}(t) - y_{j_1;\tilde{\alpha}_1}\right]\left[z_{j_2;\tilde{\alpha}_2}(t) - y_{j_2;\tilde{\alpha}_2}\right]\left[z_{j_3;\tilde{\alpha}_3}(t) - y_{j_3;\tilde{\alpha}_3}\right] ; \end{aligned} \tag{∞.81}$$

in conjunction the NTK will evolve as (∞.6),

$$\begin{aligned} &H_{i_1i_2;\delta_1\delta_2}(t+1) \\ &\quad = H_{i_1i_2;\delta_1\delta_2}(t) - \eta \sum_{j,\tilde{\alpha}} \left(\mathrm{d}H_{i_1i_2j;\delta_1\delta_2\tilde{\alpha}}(t) + \mathrm{d}H_{i_2i_1j;\delta_2\delta_1\tilde{\alpha}}(t)\right)\left[z_{j;\tilde{\alpha}}(t) - y_{j;\tilde{\alpha}}\right] \\ &\qquad + \frac{\eta^2}{2} \sum_{j_1,j_2,\tilde{\alpha}_1,\tilde{\alpha}_2} \left(\widehat{\mathrm{dd_I}H}_{i_1i_2j_1j_2;\delta_1\delta_2\tilde{\alpha}_1\tilde{\alpha}_2} + \widehat{\mathrm{dd_I}H}_{i_2i_1j_1j_2;\delta_2\delta_1\tilde{\alpha}_1\tilde{\alpha}_2} + 2\widehat{\mathrm{dd_{II}}H}_{i_1i_2j_1j_2;\delta_1\delta_2\tilde{\alpha}_1\tilde{\alpha}_2}\right) \\ &\qquad\qquad \times \left[z_{j_1;\tilde{\alpha}_1}(t) - y_{j_1;\tilde{\alpha}_1}\right]\left[z_{j_2;\tilde{\alpha}_2}(t) - y_{j_2;\tilde{\alpha}_2}\right] ; \end{aligned} \tag{∞.82}$$

and in further conjunction the dNTK will evolve as (∞.9),

$$\begin{aligned} &\mathrm{d}H_{i_0i_1i_2;\delta_0\delta_1\delta_2}(t+1) \\ &= \mathrm{d}H_{i_0i_1i_2;\delta_0\delta_1\delta_2}(t) \\ &\quad - \eta \sum_{j,\tilde{\alpha}} \left(\widehat{\mathrm{dd_I}H}_{i_0i_1i_2j;\delta_0\delta_1\delta_2\tilde{\alpha}} + \widehat{\mathrm{dd_{II}}H}_{i_0i_1i_2j;\delta_0\delta_1\delta_2\tilde{\alpha}} + \widehat{\mathrm{dd_{II}}H}_{i_0i_2i_1j;\delta_0\delta_2\delta_1\tilde{\alpha}}\right)\left[z_{j;\tilde{\alpha}}(t) - y_{j;\tilde{\alpha}}\right] . \end{aligned} \tag{∞.83}$$

In writing these dynamical equations, we've stopped explicitly denoting that our equations have $O(1/n^2)$ errors, and we've also expressed the fact that the ddNTKs are

[15] Don't let the word *tensorial* scare you here; this just means that we will allow for our training hyperparameters $\lambda_b^{(\ell)}$ and $\lambda_W^{(\ell)}$ as part of the definition of the NTK. We hope it's already clear why including these hyperparameters is a good idea – if not, please flip back to §9.4 and reread the paragraphs on the learning rate equivalence principle – and they are actually simple to include practically as part of any optimization algorithm.

Moreover, as they are just part of the definition of the NTK, they have absolutely no consequence on the dynamics presented here; i.e., our solution also covers *nontensorial* gradient descent, though in such a case you'd have different asymptotic solutions for the statistics of the NTK, dNTK, and ddNTKs. This is also why we think of the training hyperparameters $\lambda_b^{(\ell)}$ and $\lambda_W^{(\ell)}$ as being independent from the details of the optimization algorithm itself.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

t-independent at order $1/n$, using their values at initialization, $\widehat{\mathrm{dd_I}H}_{i_1 i_2 i_3 i_4;\delta_1\delta_2\delta_3\delta_4}$ and $\widehat{\mathrm{dd_{II}}H}_{i_1 i_2 i_3 i_4;\delta_1\delta_2\delta_3\delta_4}$, with hats to remind us of their stochasticity. These joint updates (∞.81), (∞.82), and (∞.83) are coupled *difference equations*, and the equations for the network output and the dynamical NTK are both *nonlinear* in the output $z_{i;\delta}$.

Although this seems daunting, we are now going to solve these equations in a closed form. First, we'll analyze the dynamics while neglecting the finite-width effects of the dNTK and ddNTKs, in which the problem will reduce to a single *linear* difference equation. Then, we'll use perturbation theory to incorporate the effect of all the NTK differentials and allow for a dynamically evolving NTK and dNTK.

Free Theory: Step-Independent NTK

Let's begin by setting the dNTK and ddNTKs to zero. Since their leading statistics are $1/n$-suppressed, we'll be able to use perturbation theory to reincorporate all their effects later. In this limit the NTK update equation (∞.82) is trivial, solved by the *free* or *step-independent* NTK:

$$H_{i_1 i_1;\delta_1\delta_2}(t) = H_{i_1 i_2;\delta_1\delta_2}(t=0) \equiv \widehat{H}_{i_1 i_2;\delta_1\delta_2}. \tag{∞.84}$$

Unsurprisingly, this just means that when the dNTK and ddNTKs vanish, the NTK doesn't update from its initialization.

Plugging this solution into the preactivation update equation (∞.81) and turning off the dNTK and ddNTKs, the remaining dynamical equation simplifies to

$$z_{i;\delta}(t+1) = z_{i;\delta}(t) - \eta \sum_{j,\tilde{\alpha}} \widehat{H}_{ij;\delta\tilde{\alpha}} [z_{j;\tilde{\alpha}}(t) - y_{j;\tilde{\alpha}}]. \tag{∞.85}$$

Thus, the residual training error, $z_{\tilde{\alpha}}(t) - y_{\tilde{\alpha}}$, sources the updates to the network output $z_\delta(t)$ for general inputs $\delta \in \mathcal{D}$. Moreover, when restricted to the inputs from the training set $\tilde{\alpha} \in \mathcal{A}$, we can rewrite this difference equation (∞.85) as

$$z_{i;\tilde{\alpha}}(t+1) - y_{i;\tilde{\alpha}} = \sum_{j,\tilde{\alpha}_1} \left(I_{ij;\tilde{\alpha}\tilde{\alpha}_1} - \eta \widehat{H}_{ij;\tilde{\alpha}\tilde{\alpha}_1} \right) [z_{j;\tilde{\alpha}_1}(t) - y_{j;\tilde{\alpha}_1}], \tag{∞.86}$$

where we've defined the *identity operator*

$$I_{i_1 i_2;\tilde{\alpha}_1\tilde{\alpha}_2} \equiv \delta_{i_1 i_2}\delta_{\tilde{\alpha}_1\tilde{\alpha}_2}. \tag{∞.87}$$

In this form, (∞.86) is a first-order homogeneous linear difference equation for the residual training error, $z_{\tilde{\alpha}}(t) - y_{\tilde{\alpha}}$, which is just a fancy way of saying that this is going to be a piece of cake.

In particular, the update to the prediction error is just a simple multiplication by a constant matrix, and the solution is given by an exponential:

$$z^{\mathrm{F}}_{i;\tilde{\alpha}}(t) - y_{i;\tilde{\alpha}} = \sum_{j,\tilde{\alpha}_1} U_{ij;\tilde{\alpha}\tilde{\alpha}_1}(t) \left(z_{j;\tilde{\alpha}_1} - y_{j;\tilde{\alpha}_1} \right). \tag{∞.88}$$

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

Here, on the left-hand side, we've labeled the solution with an "F" to indicate it's the *free solution*, with the nonlinear effects from the dNTK and ddNTKs turned off; on the right-hand side, we have the residual training error at initialization, and the **step-evolution operator** is defined as an iterative product of t steps:

$$U_{i_t i_0;\tilde{\alpha}_t\tilde{\alpha}_0}(t) \equiv \left[\left(I - \eta\widehat{H}\right)^t\right]_{i_t i_0;\tilde{\alpha}_t\tilde{\alpha}_0} \tag{∞.89}$$
$$= \sum_{\substack{i_1,\ldots i_{t-1} \\ \tilde{\alpha}_1,\ldots,\tilde{\alpha}_{t-1}}} \left(I_{i_t i_{t-1};\tilde{\alpha}_t\tilde{\alpha}_{t-1}} - \eta\widehat{H}_{i_t i_{t-1};\tilde{\alpha}_t\tilde{\alpha}_{t-1}}\right)\cdots\left(I_{i_1 i_0;\tilde{\alpha}_1,\tilde{\alpha}_0} - \eta\widehat{H}_{i_1 i_0;\tilde{\alpha}_1\tilde{\alpha}_0}\right).$$

For any positive-definite NTK and sufficiently small global learning rate η, this operator will exponentially decay to zero, $U(t) \to 0$, as the number of steps becomes large, $t \to \infty$.[16] Thus, the residual training error (∞.88) will vanish exponentially quickly,

$$\lim_{t\to\infty} z^{\mathrm{F}}_{i;\tilde{\alpha}}(t) = y_{i;\tilde{\alpha}}, \tag{∞.90}$$

with the step scale for the decay of the individual components set by the step-independent NTK.[17]

Having now solved the free dynamics on the training set, we can plug this solution (∞.88) back into the difference equation (∞.85) for general inputs $\delta \in \mathcal{D}$. With the source known explicitly, we can easily write down a solution that satisfies the initial condition $z^{\mathrm{F}}_{i;\delta}(t=0) = z_{i;\delta}$:

$$z^{\mathrm{F}}_{i;\delta}(t) = z_{i;\delta} - \sum_{j,\tilde{\alpha}} \widehat{H}_{ij;\delta\tilde{\alpha}} \left\{\eta \sum_{s=0}^{t-1}\left[z^{\mathrm{F}}_{j;\tilde{\alpha}}(s) - y_{j;\tilde{\alpha}}\right]\right\} \tag{∞.91}$$
$$= z_{i;\delta} - \sum_{j,\tilde{\alpha}_1,\tilde{\alpha}_2} \widehat{H}_{ij;\delta\tilde{\alpha}_1} a_{j;\tilde{\alpha}}(t).$$

[16] In particular, for this limit to converge we just need $||I - \eta\widehat{H}||_\infty < 1$, i.e., the largest eigenvalue of the operator $I - \eta\widehat{H}$ must be less than one. With our attention to the principles of criticality and equivalence, our choices of initialization and training hyperparameters were made so that the NTK is always of order one, and thus it's very easy for our networks to satisfy this constraint.

[17] If we wanted to study the ODE or *continuum limit* of the dynamics, we could take the global learning rate to zero, $\eta \to 0$, while holding the product, $\tau \equiv \eta t$, fixed. In such a limit, the step-evolution operator becomes simply $U(t) \to \exp(-\widehat{H}\tau)$. While such a limit is mostly unnecessary for any theoretical purpose – it's just as easy to study the discrete dynamics that actually describe the practical optimization algorithm – it does provide more substance to the objection in footnote 5 of §7.2 of the name *neural tangent kernel* for the stochastic operator $\widehat{H}$. In particular, this continuum limit makes clear that the NTK is really best thought of as a *Hamiltonian* as it generates the evolution of observables, and the step-evolution operator $U(t)$ is like a unitary time-evolution operator, albeit in *i*maginary time. More precisely, in the limit where the dNTK and ddNTKs are set to zero, the NTK is akin to a *free* Hamiltonian, with exactly solvable dynamics; with a nonzero dNTK or ddNTKs, the Hamiltonian includes nontrivial *interactions* and can be analyzed via time-dependent perturbation theory.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

Here we've defined a dynamical helper function, with an explicit representation given by

$$a_{j;\tilde{\alpha}}(t) \equiv \eta \sum_{s=0}^{t-1} \left[z^{\mathrm{F}}_{j;\tilde{\alpha}}(s) - y_{j;\tilde{\alpha}} \right] = \eta \sum_{s=0}^{t-1} \left[\sum_{k,\tilde{\alpha}_1} U_{jk;\tilde{\alpha}\tilde{\alpha}_1}(t) \left(z_{k;\tilde{\alpha}_1} - y_{k;\tilde{\alpha}_1} \right) \right] \tag{∞.92}$$

$$= \eta \sum_{k,\tilde{\alpha}_1} \left\{ \sum_{s=0}^{t-1} \left[\left(I - \eta \widehat{H} \right)^s \right]_{jk;\tilde{\alpha}\tilde{\alpha}_1} \right\} \left(z_{k;\tilde{\alpha}_1} - y_{k;\tilde{\alpha}_1} \right)$$

$$= \eta \sum_{k,m,\tilde{\alpha}_1,\tilde{\alpha}_2} \left\{ \left[I - \left(I - \eta \widehat{H} \right) \right]^{-1} \right\}^{\tilde{\alpha}\tilde{\alpha}_2}_{jm} \left[I - \left(I - \eta \widehat{H} \right)^t \right]_{mk;\tilde{\alpha}_2\tilde{\alpha}_1} \left(z_{k;\tilde{\alpha}_1} - y_{k;\tilde{\alpha}_1} \right)$$

$$= \sum_{m,\tilde{\alpha}_2} \left(\widehat{H}^{-1} \right)^{\tilde{\alpha}\tilde{\alpha}_2}_{jm} \left\{ z_{m;\tilde{\alpha}_2} - y_{m;\tilde{\alpha}_2} - \left[z^{\mathrm{F}}_{m;\tilde{\alpha}_2}(t) - y_{m;\tilde{\alpha}_2} \right] \right\} .$$

In this expression, to get to the third line we used the standard formula for evaluating geometric sums, $1 + x + x^2 + \cdots + x^{t-1} = (1 - x^t)/(1 - x)$, and to get to the final line we evaluated the inverse and substituted in for the step-evolution operator (∞.89). Recall also that $\widehat{H}^{-1}$, defined by (∞.72), is the inverse of the stochastic NTK submatrix evaluated on the training set for a particular realization of the parameters.[18]

In particular, for sufficiently small η, as the number of steps becomes large, $t \to \infty$, the residual error $z^{\mathrm{F}}_{m;\tilde{\alpha}_2}(t) - y_{m;\tilde{\alpha}_2}$ exponentially vanishes (∞.90). Thus, the prediction for a general input $\delta \in \mathcal{D}$ (∞.91) will exponentially converge to

$$z^{\mathrm{F}}_{i;\delta}(t = \infty) = z_{i;\delta} - \sum_{j,k,\tilde{\alpha}_1,\tilde{\alpha}_2} \widehat{H}_{ij;\delta\tilde{\alpha}_1} \left(\widehat{H}^{-1} \right)^{\tilde{\alpha}_1\tilde{\alpha}_2}_{jk} \left(z_{k;\tilde{\alpha}_2} - y_{k;\tilde{\alpha}_2} \right) . \tag{∞.93}$$

Although we took the limit of a large number of steps here, the exponential convergence means that for any number of steps T such that $T \gtrsim (\eta \widehat{H})^{-1}$, the prediction error will be exponentially close to its final $T \to \infty$ value.

In fact, this solution (∞.93) precisely matches the solution we would have gotten in §∞.2.1 after the first update (∞.68) with the Newton tensor (∞.71), so long as the dNTK and ddNTKs are turned off. This means that the *algorithm dependence* that emerges at finite width is solely due to the presence of the NTK differentials and the resulting nonlinear dynamics.

Interacting Theory: Dynamical NTK and dNTK

Now, let's incorporate the nonzero dNTK and ddNTKs into our analysis. To do so, we need to decompose both the dynamical network output and the dynamical NTK as

$$z_{i;\delta}(t) \equiv z^{\mathrm{F}}_{i;\delta}(t) + z^{\mathrm{I}}_{i;\delta}(t), \tag{∞.94}$$

$$H_{ij;\delta\tilde{\alpha}}(t) \equiv \widehat{H}_{ij;\delta\tilde{\alpha}} + H^{\mathrm{I}}_{ij;\delta\tilde{\alpha}}(t). \tag{∞.95}$$

[18] Unfortunately, in the absence of a Newton tensor, the raised/lowered sample indices in (∞.92) do not align well. (In fact, the problem really began in the update equation (∞.85) with the doubled-lowered $\tilde{\alpha}$ index.) If you'd like, you can fix this by judicious use of identity matrices such as $\delta_{\tilde{\alpha}_1\tilde{\alpha}_2}$ and $\delta^{\tilde{\alpha}_3\tilde{\alpha}_4}$. However, such usage over-clutters the presentation and so is probably not worth it, despite the additional clarity and *type safety* that such index alignment provides.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

Here, for the network output, $z^{\mathrm{F}}_{i;\delta}(t)$ is the *free* part, satisfying the linear difference equation (∞.85) with a solution given by (∞.91), and $z^{\mathrm{I}}_{i;\delta}(t)$ is the *interacting* part, encapsulating the corrections due to all the nonzero NTK differentials. Similarly, for the NTK, $\widehat{H}_{ij;\delta\tilde{\alpha}}$ is the step-independent or *free* part, fixed at initialization, and $H^{\mathrm{I}}_{ij;\delta\tilde{\alpha}}(t)$ is the dynamical step-dependent or *interaction* NTK. Since both $z^{\mathrm{I}}_{i;\delta}(t)$ and $H^{\mathrm{I}}_{ij;\delta\tilde{\alpha}}(t)$ are absent in the free limit, $\widehat{\mathrm{d}H}$, $\widehat{\mathrm{dd}_{\mathrm{I}}H}$, $\widehat{\mathrm{dd}_{\mathrm{II}}H} \to 0$, at leading order we expect them to be a linear combination of these objects, schematically

$$z^{\mathrm{I}}(t) = [\text{thing } 0](t)\ \widehat{\mathrm{d}H} + [\text{thing } 1](t)\ \widehat{\mathrm{dd}_{\mathrm{I}}H} + [\text{thing } 2](t)\ \widehat{\mathrm{dd}_{\mathrm{II}}H}, \tag{∞.96}$$

$$H^{\mathrm{I}}(t) = [\widetilde{\text{thing}}\ 0](t)\ \widehat{\mathrm{d}H} + [\widetilde{\text{thing}}\ 1](t)\ \widehat{\mathrm{dd}_{\mathrm{I}}H} + [\widetilde{\text{thing}}\ 2](t)\ \widehat{\mathrm{dd}_{\mathrm{II}}H}, \tag{∞.97}$$

where various tensorial things will have various time dependencies. This means in turn that any product of $z^{\mathrm{I}}_{i;\delta}(t)$ or $H^{\mathrm{I}}_{ij;\delta\tilde{\alpha}}(t)$ with one of $\widehat{\mathrm{d}H}, \widehat{\mathrm{dd}_{\mathrm{I}}H}, \widehat{\mathrm{dd}_{\mathrm{II}}H}$ can be neglected to leading order, which is the main reason why we'll be able to systematically solve these nonlinear dynamics by perturbation theory. Finally, the initial condition for the interacting parts of the network output and the NTK must satisfy

$$z^{\mathrm{I}}_{i;\delta}(t=0) = 0, \tag{∞.98}$$

$$H^{\mathrm{I}}_{ij;\delta\tilde{\alpha}}(t=0) = 0, \tag{∞.99}$$

since the free part of the network output already satisfies $z^{\mathrm{F}}_{i;\delta}(t=0) = z_{i;\delta}$ at initialization, and the free part of the NTK is step-independent and thus *is* the NTK at initialization. With all those in mind, our goal now is to find a solution for $z^{\mathrm{I}}_{i;\delta}(t)$ such that the full network output, $z_{i;\delta}(t)$, the interaction NTK, $H^{\mathrm{I}}_{ij;\delta\tilde{\alpha}}(t)$, and the dynamical dNTK, $\mathrm{d}H_{i_0 i_1 i_2;\delta_0\delta_1\delta_2}(t)$, all together satisfy the coupled nonlinear dynamics (∞.81), (∞.82), and (∞.83) to leading order in $1/n$.

First, let's work out the dynamics of the dNTK. From (∞.83) we see that the updates to the dNTK are sourced by the residual training error $z_{i;\tilde{\alpha}}(t) - y_{i;\tilde{\alpha}}$. Using our decomposition for the network output, (∞.94), and the initial condition,

$$\mathrm{d}H_{i_0 i_1 i_2;\delta_0\delta_1\delta_2}(t=0) = \widehat{\mathrm{d}H}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2}, \tag{∞.100}$$

after iterating the dynamics, we have

$$\begin{aligned}
&\mathrm{d}H_{i_0 i_1 i_2;\delta_0\delta_1\delta_2}(t) \qquad\qquad (∞.101)\\
&= \widehat{\mathrm{d}H}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2} - \sum_{j,\tilde{\alpha}} \Big(\widehat{\mathrm{dd}_{\mathrm{I}}H}_{i_0 i_1 i_2 j;\delta_0\delta_1\delta_2\tilde{\alpha}} + \widehat{\mathrm{dd}_{\mathrm{II}}H}_{i_0 i_1 i_2 j;\delta_0\delta_1\delta_2\tilde{\alpha}} + \widehat{\mathrm{dd}_{\mathrm{II}}H}_{i_0 i_2 i_1 j;\delta_0\delta_2\delta_1\tilde{\alpha}} \Big)\\
&\qquad\qquad \times \left\{ \eta \sum_{s=0}^{t-1} \Big[z^{\mathrm{F}}_{j;\tilde{\alpha}}(s) + z^{\mathrm{I}}_{j;\tilde{\alpha}}(s) - y_{j;\tilde{\alpha}} \Big] \right\}\\
&= \widehat{\mathrm{d}H}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2} - \sum_{j,\tilde{\alpha}} \Big(\widehat{\mathrm{dd}_{\mathrm{I}}H}_{i_0 i_1 i_2 j;\delta_0\delta_1\delta_2\tilde{\alpha}} + \widehat{\mathrm{dd}_{\mathrm{II}}H}_{i_0 i_1 i_2 j;\delta_0\delta_1\delta_2\tilde{\alpha}} + \widehat{\mathrm{dd}_{\mathrm{II}}H}_{i_0 i_2 i_1 j;\delta_0\delta_2\delta_1\tilde{\alpha}} \Big) a_{j;\tilde{\alpha}}(t).
\end{aligned}$$

Here in the last step, we first dropped the product of $z^{\mathrm{I}}_{i;\delta}(t)$ with $\widehat{\mathrm{dd}_{\mathrm{I}}H}$, as explained earlier, and then substituted in our dynamical helper function (∞.92). Now, plugging in

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

our final expression from (∞.92) and neglecting the fluctuation part of the NTK inverse as subleading, we find an expression for the dynamical dNTK:

$$\begin{aligned}\widehat{\mathrm{d}H}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2}(t) &= \widehat{\mathrm{d}H}_{i_0 i_1 i_2;\delta_0\delta_1\delta_2} - \sum_{k,\tilde{\alpha}_1,\tilde{\alpha}_2} \Big(\widehat{\mathrm{dd_I}H}_{i_0 i_1 i_2 j;\delta_0\delta_1\delta_2\tilde{\alpha}_1} \\ &\quad + \widehat{\mathrm{dd_{II}}H}_{i_0 i_1 i_2 j;\delta_0\delta_1\delta_2\tilde{\alpha}_1} + \widehat{\mathrm{dd_{II}}H}_{i_0 i_2 i_1 j;\delta_0\delta_2\delta_1\tilde{\alpha}_1} \Big) \\ &\quad \times \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_2} \Big\{ z_{j;\tilde{\alpha}_2} - y_{j;\tilde{\alpha}_2} - \Big[z^{\mathrm{F}}_{j;\tilde{\alpha}_2}(t) - y_{j;\tilde{\alpha}_2} \Big] \Big\} . \end{aligned} \tag{∞.102}$$

In particular, the quantity in the curly brackets represents the difference in training errors between initialization and step t: the larger this difference – i.e., the more the residual training error decreases – the greater the evolution of the dNTK and the more the meta feature functions evolve, undergoing their own form of *meta representation learning* over the course of training.

Next, using our decompositions for the network output and NTK, (∞.94) and (∞.95), we can rewrite the NTK dynamics (∞.82) as a difference equation for the interaction NTK:

$$\begin{aligned} &H^{\mathrm{I}}_{i_1 i_2;\delta_1\delta_2}(t+1) \\ &\quad = H^{\mathrm{I}}_{i_1 i_2;\delta_1\delta_2}(t) - \eta \sum_{j,\tilde{\alpha}} \Big[\mathrm{d}H_{i_1 i_2 j;\delta_1\delta_2\tilde{\alpha}}(t) + \mathrm{d}H_{i_2 i_1 j;\delta_2\delta_1\tilde{\alpha}}(t) \Big] \Big[z^{\mathrm{F}}_{j;\tilde{\alpha}}(t) - y_{j;\tilde{\alpha}} \Big] \\ &\qquad + \frac{\eta^2}{2} \sum_{j_1,j_2,\tilde{\alpha}_1,\tilde{\alpha}_2} \Big(\widehat{\mathrm{dd_I}H}_{i_1 i_2 j_1 j_2;\delta_1\delta_2\tilde{\alpha}_1\tilde{\alpha}_2} + \widehat{\mathrm{dd_I}H}_{i_2 i_1 j_1 j_2;\delta_2\delta_1\tilde{\alpha}_1\tilde{\alpha}_2} + 2\widehat{\mathrm{dd_{II}}H}_{i_1 i_2 j_1 j_2;\delta_1\delta_2\tilde{\alpha}_1\tilde{\alpha}_2} \Big) \\ &\qquad\qquad \times \Big[z^{\mathrm{F}}_{j_1;\tilde{\alpha}_1}(t) - y_{j_1;\tilde{\alpha}_1} \Big] \Big[z^{\mathrm{F}}_{j_2;\tilde{\alpha}_2}(t) - y_{j_2;\tilde{\alpha}_2} \Big] . \end{aligned} \tag{∞.103}$$

Here, once again, we've dropped the interacting part of the network output on the right-hand side, as it is always multiplied by either the dNTK or the ddNTKs and thus will be subleading in $1/n$. Denoting the free part of the residual training error (∞.88) as

$$\epsilon^{\mathrm{F}}_{j;\tilde{\alpha}}(t) \equiv z^{\mathrm{F}}_{j;\tilde{\alpha}}(t) - y_{j;\tilde{\alpha}} \tag{∞.104}$$

for notational convenience and substituting in our solution for the dynamical dNTK (∞.102), we get

$$\begin{aligned} &H^{\mathrm{I}}_{i_1 i_2;\delta_1\delta_2}(t+1) - H^{\mathrm{I}}_{i_1 i_2;\delta_1\delta_2}(t) \\ &\quad = -\eta \sum_{j,\tilde{\alpha}} \Big(\widehat{\mathrm{d}H}_{i_1 i_2 j;\delta_1\delta_2\tilde{\alpha}} + \widehat{\mathrm{d}H}_{i_2 i_1 j;\delta_2\delta_1\tilde{\alpha}} \Big) \epsilon^{\mathrm{F}}_{j;\tilde{\alpha}}(t) \\ &\qquad + \eta \sum_{j,k,\tilde{\alpha},\tilde{\alpha}_1,\tilde{\alpha}_2} \Big(\widehat{\mathrm{dd_I}H}_{i_1 i_2 j k;\delta_1\delta_2\tilde{\alpha}\tilde{\alpha}_1} + \widehat{\mathrm{dd_{II}}H}_{i_1 i_2 j k;\delta_1\delta_2\tilde{\alpha}\tilde{\alpha}_1} + \widehat{\mathrm{dd_{II}}H}_{i_1 j i_2 k;\delta_1\tilde{\alpha}\delta_2\tilde{\alpha}_1} \\ &\qquad\qquad + \widehat{\mathrm{dd_I}H}_{i_2 i_1 j k;\delta_2\delta_1\tilde{\alpha}\tilde{\alpha}_1} + \widehat{\mathrm{dd_{II}}H}_{i_2 i_1 j k;\delta_2\delta_1\tilde{\alpha}\tilde{\alpha}_1} + \widehat{\mathrm{dd_{II}}H}_{i_2 j i_1 k;\delta_2\tilde{\alpha}\delta_1\tilde{\alpha}_1} \Big) \\ &\qquad\qquad \times \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_2} \epsilon^{\mathrm{F}}_{j;\tilde{\alpha}}(t) \Big[z_{k;\tilde{\alpha}_2} - y_{k;\tilde{\alpha}_2} - \epsilon^{\mathrm{F}}_{k;\tilde{\alpha}_2}(t) \Big] \\ &\qquad + \frac{\eta^2}{2} \sum_{j_1,j_2,\tilde{\alpha}_1,\tilde{\alpha}_2} \Big(\widehat{\mathrm{dd_I}H}_{i_1 i_2 j_1 j_2;\delta_1\delta_2\tilde{\alpha}_1\tilde{\alpha}_2} + \widehat{\mathrm{dd_I}H}_{i_2 i_1 j_1 j_2;\delta_2\delta_1\tilde{\alpha}_1\tilde{\alpha}_2} + 2\widehat{\mathrm{dd_{II}}H}_{i_1 i_2 j_1 j_2;\delta_1\delta_2\tilde{\alpha}_1\tilde{\alpha}_2} \Big) \\ &\qquad\qquad \times \epsilon^{\mathrm{F}}_{j_1;\tilde{\alpha}_1}(t)\, \epsilon^{\mathrm{F}}_{j_2;\tilde{\alpha}_2}(t). \end{aligned} \tag{∞.105}$$

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

In particular, the step dependence on the right-hand side is expressed entirely in terms of the free residual training error $\epsilon^{\mathrm{F}}_{j;\tilde{\alpha}}(t)$, and each term is either linear or quadratic in $\epsilon^{\mathrm{F}}_{j;\tilde{\alpha}}(t)$. Thus, in order to solve this difference equation and get $H^{\mathrm{I}}_{i_1 i_2;\delta_1\delta_2}(t)$, we'll just have to compute sums over these terms.

One of those sums – the one that's linear in the free residual training error – is the dynamical helper function $a_{j;\tilde{\alpha}}(t) \equiv \eta \sum_{s=0}^{t-1} \epsilon^{\mathrm{F}}_{j;\tilde{\alpha}}(t)$ that we evaluated in (∞.92). The other type of sum is quadratic in the free residual training error, which will define a second dynamical helper function:

$$
\begin{aligned}
b_{j_1 j_2;\tilde{\alpha}_1\tilde{\alpha}_2}(t) &\equiv \eta \sum_{s=0}^{t-1} \epsilon^{\mathrm{F}}_{j_1;\tilde{\alpha}_1}(t)\epsilon^{\mathrm{F}}_{j_2;\tilde{\alpha}_2}(t) \qquad (\infty.106)\\
&= \eta \sum_{k_1,k_2,\tilde{\alpha}_3,\tilde{\alpha}_4} \sum_{s=0}^{t-1} \Big[\Big(I - \eta\widehat{H}\Big)^s\Big]_{j_1 k_1;\tilde{\alpha}_1\tilde{\alpha}_3} \\
&\qquad\qquad \Big[\Big(I - \eta\widehat{H}\Big)^s\Big]_{j_2 k_2;\tilde{\alpha}_2\tilde{\alpha}_4} (z_{k_1;\tilde{\alpha}_3} - y_{k_1;\tilde{\alpha}_3})\,(z_{k_2;\tilde{\alpha}_4} - y_{k_2;\tilde{\alpha}_4}) \\
&= \sum_{\tilde{\alpha}_3,\tilde{\alpha}_4} X^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}_{\mathrm{II}} \Big[(z_{j_1;\tilde{\alpha}_3} - y_{j_1;\tilde{\alpha}_3})\,(z_{j_2;\tilde{\alpha}_4} - y_{j_2;\tilde{\alpha}_4}) - \epsilon^{\mathrm{F}}_{j_1;\tilde{\alpha}_3}(t)\epsilon^{\mathrm{F}}_{j_2;\tilde{\alpha}_4}(t)\Big] \\
&\quad + O\Big(\frac{1}{n}\Big)\,.
\end{aligned}
$$

To evaluate this sum, we again used our expression for the free residual training error, (∞.88), in conjunction with the definition of the step-evolution operator, (∞.89). Then, we replaced the stochastic NTK $\widehat{H}_{ij;\tilde{\alpha}_1\tilde{\alpha}_2}$ of the training set by its mean $\delta_{ij}H_{\tilde{\alpha}_1\tilde{\alpha}_2}$, at the cost of subleading corrections, and formally performed the geometric sum as in (∞.92). This last operation yielded an *inverting tensor* $X^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}_{\mathrm{II}}$ implicitly defined by

$$
\begin{aligned}
\delta^{\tilde{\alpha}_1}_{\tilde{\alpha}_5}\delta^{\tilde{\alpha}_2}_{\tilde{\alpha}_6} &= \sum_{\tilde{\alpha}_3,\tilde{\alpha}_4\in\mathcal{A}} X^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}_{\mathrm{II}} \frac{1}{\eta}\Big[\delta_{\tilde{\alpha}_3\tilde{\alpha}_5}\delta_{\tilde{\alpha}_4\tilde{\alpha}_6} - (\delta_{\tilde{\alpha}_3\tilde{\alpha}_5} - \eta\widetilde{H}_{\tilde{\alpha}_3\tilde{\alpha}_5})(\delta_{\tilde{\alpha}_4\tilde{\alpha}_6} - \eta\widetilde{H}_{\tilde{\alpha}_4\tilde{\alpha}_6})\Big] \\
&= \sum_{\tilde{\alpha}_3,\tilde{\alpha}_4\in\mathcal{A}} X^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}_{\mathrm{II}} \Big(\widetilde{H}_{\tilde{\alpha}_3\tilde{\alpha}_5}\delta_{\tilde{\alpha}_4\tilde{\alpha}_6} + \delta_{\tilde{\alpha}_3\tilde{\alpha}_5}\widetilde{H}_{\tilde{\alpha}_4\tilde{\alpha}_6} - \eta\widetilde{H}_{\tilde{\alpha}_3\tilde{\alpha}_5}\widetilde{H}_{\tilde{\alpha}_4\tilde{\alpha}_6}\Big)\,. \qquad (\infty.107)
\end{aligned}
$$

This tensor is a generalization of the familiar inverting matrix $X^{\tilde{\alpha}_1\tilde{\alpha}_2}_{\mathrm{I}} \equiv \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_2}$ for geometric sums over matrices that satisfies

$$
\delta^{\tilde{\alpha}_1}_{\tilde{\alpha}_3} = \sum_{\tilde{\alpha}_2\in\mathcal{A}} X^{\tilde{\alpha}_1\tilde{\alpha}_2}_{\mathrm{I}} \frac{1}{\eta}\Big[\delta_{\tilde{\alpha}_2\tilde{\alpha}_3} - (\delta_{\tilde{\alpha}_2\tilde{\alpha}_3} - \eta\widetilde{H}_{\tilde{\alpha}_2\tilde{\alpha}_3})\Big] = \sum_{\tilde{\alpha}_2\in\mathcal{A}} X^{\tilde{\alpha}_1\tilde{\alpha}_2}_{\mathrm{I}}\widetilde{H}_{\tilde{\alpha}_2\tilde{\alpha}_3} \qquad (\infty.108)
$$

and appeared in the last expression of the dynamical helper function $a_{j;\tilde{\alpha}}(t)$ (∞.92). While we are on fire like an activated neuron, let's also define a final dynamical helper function for sums that are cubic in the free residual training error:

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

$$c_{j_1 j_2 j_3;\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}(t) \equiv \eta \sum_{s=0}^{t} \epsilon^{\mathrm{F}}_{j_1;\tilde{\alpha}_1}(t)\epsilon^{\mathrm{F}}_{j_2;\tilde{\alpha}_2}(t)\epsilon^{\mathrm{F}}_{j_3;\tilde{\alpha}_3}(t) \tag{∞.109}$$

$$= \sum_{\tilde{\alpha}_4,\tilde{\alpha}_5,\tilde{\alpha}_6} X_{\mathrm{III}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6} \Big[\left(z_{k_1;\tilde{\alpha}_4} - y_{k_1;\tilde{\alpha}_4}\right)\left(z_{k_2;\tilde{\alpha}_5} - y_{k_2;\tilde{\alpha}_5}\right)\left(z_{k_3;\tilde{\alpha}_6} - y_{k_3;\tilde{\alpha}_6}\right) - \epsilon^{\mathrm{F}}_{j_1;\tilde{\alpha}_4}(t)\,\epsilon^{\mathrm{F}}_{j_2;\tilde{\alpha}_5}(t)\,\epsilon^{\mathrm{F}}_{j_3;\tilde{\alpha}_6}(t) \Big] + O\left(\frac{1}{n}\right).$$

In this case, the inverting tensor $X_{\mathrm{III}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}$ is implicitly defined by

$$\delta^{\tilde{\alpha}_1}_{\tilde{\alpha}_7}\delta^{\tilde{\alpha}_2}_{\tilde{\alpha}_8}\delta^{\tilde{\alpha}_3}_{\tilde{\alpha}_9} \tag{∞.110}$$

$$= \sum_{\tilde{\alpha}_4,\tilde{\alpha}_5,\tilde{\alpha}_6} X_{\mathrm{III}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6} \Big[\Big(\widetilde{H}_{\tilde{\alpha}_4\tilde{\alpha}_7}\delta_{\tilde{\alpha}_5\tilde{\alpha}_8}\delta_{\tilde{\alpha}_6\tilde{\alpha}_9} + \delta_{\tilde{\alpha}_4\tilde{\alpha}_7}\widetilde{H}_{\tilde{\alpha}_5\tilde{\alpha}_8}\delta_{\tilde{\alpha}_6\tilde{\alpha}_9} + \delta_{\tilde{\alpha}_4\tilde{\alpha}_7}\delta_{\tilde{\alpha}_5\tilde{\alpha}_8}\widetilde{H}_{\tilde{\alpha}_6\tilde{\alpha}_9}\Big)$$
$$- \eta\Big(\widetilde{H}_{\tilde{\alpha}_4\tilde{\alpha}_7}\widetilde{H}_{\tilde{\alpha}_5\tilde{\alpha}_8}\delta_{\tilde{\alpha}_6\tilde{\alpha}_9} + \widetilde{H}_{\tilde{\alpha}_4\tilde{\alpha}_7}\delta_{\tilde{\alpha}_5\tilde{\alpha}_8}\widetilde{H}_{\tilde{\alpha}_6\tilde{\alpha}_9} + \delta_{\tilde{\alpha}_4\tilde{\alpha}_7}\widetilde{H}_{\tilde{\alpha}_5\tilde{\alpha}_8}\widetilde{H}_{\tilde{\alpha}_6\tilde{\alpha}_9}\Big)$$
$$+ \eta^2\Big(\widetilde{H}_{\tilde{\alpha}_4\tilde{\alpha}_7}\widetilde{H}_{\tilde{\alpha}_5\tilde{\alpha}_8}\widetilde{H}_{\tilde{\alpha}_6\tilde{\alpha}_9}\Big)\Big].$$

Essentially, these three dynamical helper functions encode the step dependence of various geometric sums of the free residual training error $\epsilon^{\mathrm{F}}_{j;\tilde{\alpha}}(t)$. Note that we have

$$a_{j_1;\tilde{\alpha}_1}(t=0) = 0\,, \tag{∞.111}$$

$$b_{j_1 j_2;\tilde{\alpha}_1\tilde{\alpha}_2}(t=0) = 0\,, \tag{∞.112}$$

$$c_{j_1 j_2 j_3;\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}(t=0) = 0\,, \tag{∞.113}$$

and so they all vanish at initialization, while at the end of training we have

$$a_{j_1;\tilde{\alpha}_1}(\infty) = \sum_{\tilde{\alpha}_2} X_{\mathrm{I}}^{\tilde{\alpha}_1\tilde{\alpha}_2}\left(z_{j_1;\tilde{\alpha}_2} - y_{j_1;\tilde{\alpha}_2}\right) + O\left(\frac{1}{n}\right)\,, \tag{∞.114}$$

$$b_{j_1 j_2;\tilde{\alpha}_1\tilde{\alpha}_2}(\infty) = \sum_{\tilde{\alpha}_3,\tilde{\alpha}_4} X_{\mathrm{II}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}\left(z_{j_1;\tilde{\alpha}_3} - y_{j_1;\tilde{\alpha}_3}\right)\left(z_{j_2;\tilde{\alpha}_4} - y_{j_2;\tilde{\alpha}_4}\right) + O\left(\frac{1}{n}\right)\,, \tag{∞.115}$$

$$c_{j_1 j_2 j_3;\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}(\infty) = \sum_{\tilde{\alpha}_4,\tilde{\alpha}_5,\tilde{\alpha}_6} X_{\mathrm{III}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}\left(z_{j_1;\tilde{\alpha}_4} - y_{j_1;\tilde{\alpha}_4}\right)\left(z_{j_2;\tilde{\alpha}_5} - y_{j_2;\tilde{\alpha}_5}\right)\left(z_{j_3;\tilde{\alpha}_6} - y_{j_3;\tilde{\alpha}_6}\right)$$
$$+ O\left(\frac{1}{n}\right)\,, \tag{∞.116}$$

since the free residual training error $\epsilon^{\mathrm{F}}_{j;\tilde{\alpha}}(t)$ vanishes exponentially quickly, cf. (∞.90).

With the help of the first two of these dynamical helper functions, we can semi-compactly write the solution to the difference equation for the interaction NTK (∞.105) as

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

$$
\begin{aligned}
&H^{\mathrm{I}}_{i_1 i_2;\delta_1\delta_2}(t) \qquad\qquad (\infty.117)\\
&\quad = -\sum_{j,\tilde{\alpha}} \Big(\widehat{\mathrm{d}H}_{i_1 i_2 j;\delta_1\delta_2\tilde{\alpha}} + \widehat{\mathrm{d}H}_{i_2 i_1 j;\delta_2\delta_1\tilde{\alpha}} \Big) a_{j;\tilde{\alpha}}(t)\\
&\qquad + \sum_{j,k,\tilde{\alpha}_1,\tilde{\alpha}_2,\tilde{\alpha}_3} \Big(\widehat{\mathrm{dd}_{\mathrm{I}}H}_{i_1 i_2 j k;\delta_1\delta_2\tilde{\alpha}_1\tilde{\alpha}_2} + \widehat{\mathrm{dd}_{\mathrm{II}}H}_{i_1 i_2 j k;\delta_1\delta_2\tilde{\alpha}_1\tilde{\alpha}_2} + \widehat{\mathrm{dd}_{\mathrm{II}}H}_{i_1 j i_2 k;\delta_1\tilde{\alpha}_1\delta_2\tilde{\alpha}_2}\\
&\qquad\qquad + \widehat{\mathrm{dd}_{\mathrm{I}}H}_{i_2 i_1 j k;\delta_2\delta_1\tilde{\alpha}_1\tilde{\alpha}_2} + \widehat{\mathrm{dd}_{\mathrm{II}}H}_{i_2 i_1 j k;\delta_2\delta_1\tilde{\alpha}_1\tilde{\alpha}_2} + \widehat{\mathrm{dd}_{\mathrm{II}}H}_{i_2 j i_1 k;\delta_2\tilde{\alpha}_1\delta_1\tilde{\alpha}_2} \Big)\\
&\qquad\qquad\qquad \times \widetilde{H}^{\tilde{\alpha}_2\tilde{\alpha}_3} \Big[\left(z_{k;\tilde{\alpha}_3} - y_{k;\tilde{\alpha}_3}\right) a_{j;\tilde{\alpha}_1}(t) - b_{jk;\tilde{\alpha}_1\tilde{\alpha}_3}(t) \Big]\\
&\qquad + \frac{\eta}{2} \sum_{j_1,j_2,\tilde{\alpha}_1,\tilde{\alpha}_2} \Big(\widehat{\mathrm{dd}_{\mathrm{I}}H}_{i_1 i_2 j_1 j_2;\delta_1\delta_2\tilde{\alpha}_1\tilde{\alpha}_2} + \widehat{\mathrm{dd}_{\mathrm{I}}H}_{i_2 i_1 j_1 j_2;\delta_2\delta_1\tilde{\alpha}_1\tilde{\alpha}_2}\\
&\qquad\qquad\qquad + 2\widehat{\mathrm{dd}_{\mathrm{II}}H}_{i_1 i_2 j_1 j_2;\delta_1\delta_2\tilde{\alpha}_1\tilde{\alpha}_2} \Big) b_{j_1 j_2;\tilde{\alpha}_1\tilde{\alpha}_2}(t)\,.
\end{aligned}
$$

As a quick sanity check, the vanishing initial condition for the interaction NTK, (∞.99), is satisfied due to the vanishing of the helper functions at initialization, (∞.111) and (∞.112). We can also plug in (∞.114) and (∞.115) to evaluate the change in NTK at the end of training. In particular, we see that the larger the change in the NTK, the more the feature functions evolve. Thus, more initial error entails more representation learning over the course of training.

Lastly, we need to determine the step dependence of the interacting part of the network output, $z^{\mathrm{I}}_{i;\delta}(t)$. Inserting our free-interacting decomposition, (∞.94), into our dynamics for the network output, (∞.81), and using the fact that the free part satisfies the step-independent evolution equation (∞.86), we can reorganize terms to find a dynamical equation for the interacting part only:

$$
z^{\mathrm{I}}_{i;\delta}(t+1) = z^{\mathrm{I}}_{i;\delta}(t) - \sum_{j,\tilde{\alpha}} \eta\, \widehat{H}_{ij;\delta\tilde{\alpha}}\, z^{\mathrm{I}}_{j;\tilde{\alpha}}(t) + \eta\, \mathbb{F}_{i;\delta}(t). \qquad (\infty.118)
$$

Here, we've defined a *damping force*:

$$
\begin{aligned}
\mathbb{F}_{i;\delta}(t) \equiv &-\sum_{j,\tilde{\alpha}} H^{\mathrm{I}}_{ij;\delta\tilde{\alpha}}(t)\, \epsilon^{\mathrm{F}}_{j;\tilde{\alpha}}(t) + \frac{\eta}{2} \sum_{j_1,j_2,\tilde{\alpha}_1,\tilde{\alpha}_2} \mathrm{d}H_{ij_1j_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2}(t)\, \epsilon^{\mathrm{F}}_{j_1;\tilde{\alpha}_1}(t)\, \epsilon^{\mathrm{F}}_{j_2;\tilde{\alpha}_2}(t) \qquad (\infty.119)\\
&- \frac{\eta^2}{6} \sum_{j_1,j_2,j_3,\tilde{\alpha}_1,\tilde{\alpha}_2,\tilde{\alpha}_3} \widehat{\mathrm{dd}_{\mathrm{I}}H}_{ij_1j_2j_3;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}\, \epsilon^{\mathrm{F}}_{j_1;\tilde{\alpha}_1}(t)\, \epsilon^{\mathrm{F}}_{j_2;\tilde{\alpha}_2}(t)\, \epsilon^{\mathrm{F}}_{j_3;\tilde{\alpha}_3}(t).
\end{aligned}
$$

Since we have solutions for the interaction NTK and the dNTK dynamics in terms of the free residual training error solution $\epsilon^{\mathrm{F}}_{j;\tilde{\alpha}}(t)$, (∞.117) and (∞.102), this damping force is an explicitly known function of the step t.

Let us now try to implicitly express the solution to this difference equation (∞.118) as a sum over steps. First, for inputs in the training set $\tilde{\alpha} \in \mathcal{A}$, the dynamics of the interacting part of the output, (∞.118), reduce to a first-order inhomogeneous linear difference equation, with the damping force ruining the homogeneity:

$$
z^{\mathrm{I}}_{i;\tilde{\alpha}}(t+1) = \sum_{j,\tilde{\alpha}_1} \Big(I_{ij;\tilde{\alpha}\tilde{\alpha}_1} - \eta \widehat{H}_{ij;\tilde{\alpha}\tilde{\alpha}_1} \Big)\, z^{\mathrm{I}}_{j;\tilde{\alpha}_1}(t) + \eta\, \mathbb{F}_{i;\tilde{\alpha}}(t). \qquad (\infty.120)
$$

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

We can formally solve this equation as a convolution of the damping force with the free step-evolution operator (∞.89),

$$z^{\mathrm{I}}_{i;\tilde{\alpha}}(t) = \eta \sum_{s=0}^{t-1} \sum_{j,\tilde{\alpha}_1} U_{ij;\tilde{\alpha}\tilde{\alpha}_1}(t-1-s)\, \mathbb{F}_{j;\tilde{\alpha}_1}(s), \tag{∞.121}$$

which satisfies the initial condition $z^{\mathrm{I}}_{i;\tilde{\alpha}}(t=0) = 0$. Plugging this result back into the dynamical equation for general inputs $\delta \in \mathcal{D}$, (∞.118), we can then find a solution for the interacting part of the network output for such general inputs:

$$z^{\mathrm{I}}_{i;\delta}(t) = \eta \sum_{s=0}^{t-1} \left[\mathbb{F}_{i;\delta}(s) - \sum_{j,\tilde{\alpha}} \widehat{H}_{ij;\delta\tilde{\alpha}}\, z^{\mathrm{I}}_{j;\tilde{\alpha}}(s) \right]. \tag{∞.122}$$

In order to further simplify these expressions, we need to work out one convoluted sum:

$$\begin{aligned}
\eta \sum_{s=0}^{t-1} z^{\mathrm{I}}_{j;\tilde{\alpha}}(s) &= \eta^2 \sum_{s=0}^{t-1} \sum_{\tilde{s}=0}^{s-1} \sum_{k,\tilde{\alpha}_1} U_{jk;\tilde{\alpha}\tilde{\alpha}_1}(s-1-\tilde{s})\, \mathbb{F}_{k;\tilde{\alpha}_1}(\tilde{s}) \\
&= \eta \sum_{u=0}^{t-2} \sum_{k,\tilde{\alpha}_1} \left[\eta \sum_{\tilde{u}=0}^{t-u-2} U_{jk;\tilde{\alpha}\tilde{\alpha}_1}(\tilde{u}) \right] \mathbb{F}_{k;\tilde{\alpha}_1}(u) \\
&= \eta \sum_{u=0}^{t-2} \sum_{k,m,\tilde{\alpha}_1,\tilde{\alpha}_2} \left(\widehat{H}^{-1}\right)^{\tilde{\alpha}\tilde{\alpha}_2}_{jm} \left[I - U(t-u-1) \right]_{mk;\tilde{\alpha}_2\tilde{\alpha}_1} \mathbb{F}_{k;\tilde{\alpha}_1}(u) \\
&= \sum_{m,\tilde{\alpha}_2} \left(\widehat{H}^{-1}\right)^{\tilde{\alpha}\tilde{\alpha}_2}_{jm} \left\{ \left[\eta \sum_{u=0}^{t-2} \mathbb{F}_{m;\tilde{\alpha}_2}(u) \right] + \left[\eta \mathbb{F}_{m;\tilde{\alpha}_2}(t-1) - z^{\mathrm{I}}_{m;\tilde{\alpha}_2}(t) \right] \right\} \\
&= \sum_{m,\tilde{\alpha}_2} \left(\widehat{H}^{-1}\right)^{\tilde{\alpha}\tilde{\alpha}_2}_{jm} \left\{ \left[\eta \sum_{u=0}^{t-1} \mathbb{F}_{m;\tilde{\alpha}_2}(u) \right] - z^{\mathrm{I}}_{m;\tilde{\alpha}_2}(t) \right\}.
\end{aligned} \tag{∞.123}$$

Step by step, on the first line we used the expression for the formal solution (∞.121); on the second line we first reversed the order of the sums as $\sum_{s=0}^{t-1}\sum_{\tilde{s}=0}^{s-1} = \sum_{\tilde{s}=0}^{t-2}\sum_{s=\tilde{s}+1}^{t-1}$, and then we rewrote these sums in terms of new variables, $u \equiv \tilde{s}$ and $\tilde{u} \equiv s-1-\tilde{s}$; on the third line, we performed the geometric sum exactly as in (∞.92); on the fourth line, we used our formal solution (∞.121) at step t; and on the final line, we combined terms to extend the limits of the sum over the damping force. Plugging this evaluation back into our formal solution for $z^{\mathrm{I}}_{i;\delta}(t)$, (∞.122), we get a slightly less formal solution:

$$z^{\mathrm{I}}_{i;\delta}(t) = \eta \sum_{s=0}^{t-1} \left[\mathbb{F}_{i;\delta}(s) - \sum_{\tilde{\alpha}_1,\tilde{\alpha}_2} H_{\delta\tilde{\alpha}_1} \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_2} \mathbb{F}_{i;\tilde{\alpha}_2}(s) \right] + \sum_{\tilde{\alpha}_1,\tilde{\alpha}_2} H_{\delta\tilde{\alpha}_1} \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_2} z^{\mathrm{I}}_{i;\tilde{\alpha}_2}(t) + O\!\left(\frac{1}{n^2}\right). \tag{∞.124}$$

In this result, we've replaced the stochastic NTK by its mean, $\widehat{H}_{ij;\delta_1\delta_2} \to \delta_{ij} H_{\delta_1,\delta_2}$, as every term is otherwise already proportional to the dNTK or ddNTKs. As a quick sanity

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

check, note that for inputs in the training set, $\delta \to \tilde{\alpha} \in \mathcal{A}$, this general expression reduces to an identity on $z^{\mathrm{I}}_{i;\tilde{\alpha}}(t)$.

Ultimately, what we care about is the interacting solution at the end of training, $t \to \infty$. As before, let's assume that the product of η with the stochastic NTK is sufficiently small such that the free step-evolution operator,

$$\lim_{t\to\infty} U(t) \propto \exp\Big(-\eta \widehat{H} t\Big)\,, \tag{∞.125}$$

exponentially decays to zero.[19] Then, for training inputs, the interacting part of the network outputs, $z^{\mathrm{I}}_{i;\tilde{\alpha}}(t)$, will exponentially converge to zero:

$$\lim_{t\to\infty} z^{\mathrm{I}}_{i;\tilde{\alpha}}(t) = 0. \tag{∞.126}$$

To see why this holds, note that the dampening force (∞.119) decays exponentially as

$$\lim_{s\to\infty} \mathbb{F}(s) \propto \exp\Big(-\eta \widehat{H} s\Big)\,, \tag{∞.127}$$

since its leading behavior is linearly proportional to the free residual training error $\epsilon^{\mathrm{F}}_{j;\tilde{\alpha}}(t)$. Combined with (∞.125), this means that the interacting solution (∞.121) converges as

$$\begin{aligned}\lim_{t\to\infty} z^{\mathrm{I}}_{i;\tilde{\alpha}}(t) &= \eta \lim_{t\to\infty} \sum_{s=0}^{t-1} \sum_{j,\tilde{\alpha}_1} U_{ij;\tilde{\alpha}\tilde{\alpha}_1}(t-1-s)\, \mathbb{F}_{j;\tilde{\alpha}_1}(s) \\ &\propto \lim_{t\to\infty} \Big\{ \eta t \, \exp\Big[-(t-1)\eta \widehat{H}\Big] \Big\} = 0,\end{aligned} \tag{∞.128}$$

which is slightly slower than the convergence of the free solution $z^{\mathrm{F}}_{i;\tilde{\alpha}}(t) \propto \exp(-\widehat{H} t)$. Thus, overall the training algorithm converges:

$$\lim_{t\to\infty} z_{i;\tilde{\alpha}}(t) - y_{i;\tilde{\alpha}} = \lim_{t\to\infty} \Big[z^{\mathrm{F}}_{i;\tilde{\alpha}}(t) + z^{\mathrm{I}}_{i;\tilde{\alpha}}(t) \Big] - y_{i;\tilde{\alpha}} = 0, \tag{∞.129}$$

where here we also used the free solution (∞.90).[20] Incidentally, and of possible broader interest, this altogether shows that gradient descent converges exponentially quickly to a zero-error minimum for realistic deep neural networks of finite width and nonzero depth, up to errors that are at most quadratic in our effective theory cutoff L/n.

For general inputs, the interacting part of the output, $z^{\mathrm{I}}_{i;\delta}(t)$, is given by the expression (∞.124). With the convergence on the training set in mind, (∞.126), the expression in the end-of-training limit reduces to

$$\lim_{t\to\infty} z^{\mathrm{I}}_{i;\delta}(t) = \left[\eta \sum_{s=0}^{\infty} \mathbb{F}_{i;\delta}(s) \right] - \sum_{\tilde{\alpha}_1,\tilde{\alpha}_2} H_{\delta\tilde{\alpha}_1} \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_2} \left[\eta \sum_{s=0}^{\infty} \mathbb{F}_{i;\tilde{\alpha}_2}(s) \right]. \tag{∞.130}$$

[19] Note that since the step-evolution operator $U(t)$ is constructed in (∞.89) from the *step-independent* NTK, $\widehat{H}_{ij;\delta\tilde{\alpha}}$, the condition for this convergence is the same as the free analysis discussed in footnote 16.

[20] Remember that in this section we have stopped explicitly denoting that there are $O\big(1/n^2\big)$ corrections. Recalling our fully-trained condition (∞.66), this result (∞.129) should be understood to be true up to such corrections, cf. (∞.78) for our two-step solution where the situation was analogous.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

Thus, all that remains is for us to perform an infinite sum over the damping force:

$$\begin{aligned}\eta\sum_{s=0}^{\infty}\mathbb{F}_{i;\delta}(s) \equiv &-\sum_{j,\tilde{\alpha}_1}\left[\eta\sum_{s=0}^{\infty}H^{\text{I}}_{ij;\delta\tilde{\alpha}_1}(s)\,\epsilon^{\text{F}}_{j;\tilde{\alpha}_1}(s)\right] \\ &+\frac{\eta}{2}\sum_{j_1,j_2,\tilde{\alpha}_1,\tilde{\alpha}_2}\left[\eta\sum_{s=0}^{\infty}\text{d}H_{ij_1j_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2}(s)\,\epsilon^{\text{F}}_{j_1;\tilde{\alpha}_1}(s)\,\epsilon^{\text{F}}_{j_2;\tilde{\alpha}_2}(s)\right] \\ &-\frac{\eta^2}{6}\sum_{j_1,j_2,j_3,\tilde{\alpha}_1,\tilde{\alpha}_2,\tilde{\alpha}_3}\left[\eta\sum_{s=0}^{\infty}\widehat{\text{dd}_{\text{I}}H}_{ij_1j_2j_3;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}\,\epsilon^{\text{F}}_{j_1;\tilde{\alpha}_1}(s)\,\epsilon^{\text{F}}_{j_2;\tilde{\alpha}_2}(s)\,\epsilon^{\text{F}}_{j_3;\tilde{\alpha}_3}(s)\right].\end{aligned} \tag{∞.131}$$

The third sum is exactly the end-of-training limit of the third dynamical helper function $c_{j_1j_2j_3;\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}(t)$ that we already evaluated in (∞.116), giving

$$\begin{aligned}&\eta\sum_{s=0}^{\infty}\widehat{\text{dd}_{\text{I}}H}_{ij_1j_2j_3;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}\,\epsilon^{\text{F}}_{j_1;\tilde{\alpha}_1}(s)\,\epsilon^{\text{F}}_{j_2;\tilde{\alpha}_2}(s)\,\epsilon^{\text{F}}_{j_3;\tilde{\alpha}_3}(s) \\ &=\sum_{\tilde{\alpha}_4,\tilde{\alpha}_5,\tilde{\alpha}_6}\widehat{\text{dd}_{\text{I}}H}_{ij_1j_2j_3;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}X^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}_{\text{III}}\,(z_{j_1;\tilde{\alpha}_4}-y_{j_1;\tilde{\alpha}_4})\,(z_{j_2;\tilde{\alpha}_5}-y_{j_2;\tilde{\alpha}_5})\,(z_{j_3;\tilde{\alpha}_6}-y_{j_3;\tilde{\alpha}_6})\,.\end{aligned} \tag{∞.132}$$

Thus, we are left with two more sums to evaluate. Let's proceed slowly: everything is simple, but there are a lot of terms to get right.

To start, we can evaluate the second sum in (∞.131) as

$$\begin{aligned}&\eta\sum_{s=0}^{\infty}\widehat{\text{d}H}_{ij_1j_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2}(s)\,\epsilon^{\text{F}}_{j_1;\tilde{\alpha}_1}(s)\,\epsilon^{\text{F}}_{j_2;\tilde{\alpha}_2}(s) \\ &\quad=\widehat{\text{d}H}_{ij_1j_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2}\sum_{s=0}^{\infty}\left[\eta\,\epsilon^{\text{F}}_{j_1;\tilde{\alpha}_1}(s)\,\epsilon^{\text{F}}_{j_2;\tilde{\alpha}_2}(s)\right] \\ &\qquad-\sum_{k,\tilde{\alpha}_3,\tilde{\alpha}_4}\left(\widehat{\text{dd}_{\text{I}}H}_{ij_1j_2k;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}+2\widehat{\text{dd}_{\text{II}}H}_{ij_1j_2k;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}\right) \\ &\qquad\qquad\times\widetilde{H}^{\tilde{\alpha}_3\tilde{\alpha}_4}\sum_{s=0}^{\infty}\left\{\eta\,\epsilon^{\text{F}}_{j_1;\tilde{\alpha}_1}(s)\,\epsilon^{\text{F}}_{j_2;\tilde{\alpha}_2}(s)\left[z_{k;\tilde{\alpha}_4}-y_{k;\tilde{\alpha}_4}-\epsilon^{\text{F}}_{k;\tilde{\alpha}_4}(s)\right]\right\},\end{aligned} \tag{∞.133}$$

where we substituted in our dynamical dNTK solution, (∞.102), and used the fact that the overall expression is symmetric under $(\tilde{\alpha}_1,j_1)\leftrightarrow(\tilde{\alpha}_2,j_2)$ to combine the two $\text{dd}_{\text{II}}H$ terms. Then, using our already evaluated sums, (∞.115) and (∞.116), we get

$$\begin{aligned}&\eta\sum_{s=0}^{\infty}\widehat{\text{d}H}_{ij_1j_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2}(s)\,\epsilon^{\text{F}}_{j_1;\tilde{\alpha}_1}(s)\,\epsilon^{\text{F}}_{j_2;\tilde{\alpha}_2}(s) \\ &\quad=\sum_{\tilde{\alpha}_3,\tilde{\alpha}_4}\widehat{\text{d}H}_{ij_1j_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2}X^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}_{\text{II}}\,(z_{j_1;\tilde{\alpha}_3}-y_{j_1;\tilde{\alpha}_3})\,(z_{j_2;\tilde{\alpha}_4}-y_{j_2;\tilde{\alpha}_4}) \\ &\qquad-\sum_{k,\tilde{\alpha}_3,\ldots,\tilde{\alpha}_6}\left(\widehat{\text{dd}_{\text{I}}H}_{ij_1j_2k;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}+2\widehat{\text{dd}_{\text{II}}H}_{ij_1j_2k;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}\right) \\ &\qquad\qquad\times Y^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}_{1}\,(z_{j_1;\tilde{\alpha}_4}-y_{j_1;\tilde{\alpha}_4})\,(z_{j_2;\tilde{\alpha}_5}-y_{j_2;\tilde{\alpha}_5})\,(z_{k;\tilde{\alpha}_6}-y_{k;\tilde{\alpha}_6})\,,\end{aligned} \tag{∞.134}$$

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

where we introduced a shorthand

$$Y_1^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6} \equiv X_{\text{II}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_4\tilde{\alpha}_5}\widetilde{H}^{\tilde{\alpha}_3\tilde{\alpha}_6} - \sum_{\tilde{\alpha}_7}\widetilde{H}^{\tilde{\alpha}_3\tilde{\alpha}_7}X_{\text{III}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_7\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}, \tag{∞.135}$$

to ease the collection of terms later.

To finish, let us write the first sum in (∞.131) as

$$\begin{aligned}
&\eta\sum_{s=0}^{\infty} H^{\text{I}}_{ij;\delta\tilde{\alpha}_1}(s)\,\epsilon^{\text{F}}_{j;\tilde{\alpha}_1}(s) \qquad\qquad (\infty.136)\\
&\quad= -\sum_{k,\tilde{\alpha}_2}\Big(\widehat{\text{d}H}_{ijk;\delta\tilde{\alpha}_1\tilde{\alpha}_2} + \widehat{\text{d}H}_{jik;\tilde{\alpha}_1\delta\tilde{\alpha}_2}\Big)\sum_{s=0}^{\infty}\Big[\eta\,\epsilon^{\text{F}}_{j;\tilde{\alpha}_1}(s)\,a_{k;\tilde{\alpha}_2}(s)\Big]\\
&\qquad+\sum_{k_1,k_2,\tilde{\alpha}_2,\tilde{\alpha}_3,\tilde{\alpha}_4}\Big(\widehat{\text{dd}_{\text{I}}H}_{ijk_1k_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3} + \widehat{\text{dd}_{\text{II}}H}_{ijk_1k_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3} + \widehat{\text{dd}_{\text{II}}H}_{ik_1jk_2;\delta\tilde{\alpha}_2\tilde{\alpha}_1\tilde{\alpha}_3}\\
&\qquad\qquad+\widehat{\text{dd}_{\text{I}}H}_{jik_1k_2;\tilde{\alpha}_1\delta\tilde{\alpha}_2\tilde{\alpha}_3} + \widehat{\text{dd}_{\text{II}}H}_{jik_1k_2;\tilde{\alpha}_1\delta\tilde{\alpha}_2\tilde{\alpha}_3} + \widehat{\text{dd}_{\text{II}}H}_{jk_1ik_2;\tilde{\alpha}_1\tilde{\alpha}_2\delta\tilde{\alpha}_3}\Big)\\
&\qquad\qquad\times\widetilde{H}^{\tilde{\alpha}_3\tilde{\alpha}_4}\sum_{s=0}^{\infty}\Big\{\eta\,\epsilon^{\text{F}}_{j;\tilde{\alpha}_1}(s)\Big[\,(z_{k_2;\tilde{\alpha}_4}-y_{k_2;\tilde{\alpha}_4})\,a_{k_1;\tilde{\alpha}_2}(s) - b_{k_1k_2;\tilde{\alpha}_2\tilde{\alpha}_4}(s)\Big]\Big\}\\
&\qquad+\frac{\eta}{2}\sum_{k_1,k_2,\tilde{\alpha}_2,\tilde{\alpha}_3}\Big(\widehat{\text{dd}_{\text{I}}H}_{ijk_1k_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3} + \widehat{\text{dd}_{\text{I}}H}_{jik_1k_2;\tilde{\alpha}_1\delta\tilde{\alpha}_2\tilde{\alpha}_3} + 2\widehat{\text{dd}_{\text{II}}H}_{ijk_1k_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}\Big)\\
&\qquad\qquad\times\sum_{s=0}^{\infty}\Big[\eta\,\epsilon^{\text{F}}_{j;\tilde{\alpha}_1}(s)\,b_{k_1k_2;\tilde{\alpha}_2\tilde{\alpha}_3}(s)\Big]\,,
\end{aligned}$$

where we substituted in our solution for the interaction NTK, (∞.117). Then, substituting for the helper functions with (∞.92) and (∞.106), performing the *additional* sums over these terms, and then using the end-of-training limits (∞.114)–(∞.116), we get

$$\begin{aligned}
&\eta\sum_{s=0}^{\infty} H^{\text{I}}_{ij;\delta\tilde{\alpha}_1}(s)\epsilon^{\text{F}}_{j;\tilde{\alpha}_1}(s) \qquad\qquad (\infty.137)\\
&\quad= -\sum_{k,\tilde{\alpha}_2,\tilde{\alpha}_3,\tilde{\alpha}_4}\Big(\widehat{\text{d}H}_{ijk;\delta\tilde{\alpha}_1\tilde{\alpha}_2} + \widehat{\text{d}H}_{jik;\tilde{\alpha}_1\delta\tilde{\alpha}_2}\Big)Y_2^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}\,(z_{j;\tilde{\alpha}_3}-y_{j;\tilde{\alpha}_3})\,(z_{k;\tilde{\alpha}_4}-y_{k;\tilde{\alpha}_4})\\
&\qquad+\sum_{k_1,k_2,\tilde{\alpha}_2,\tilde{\alpha}_3,\tilde{\alpha}_4,\tilde{\alpha}_5,\tilde{\alpha}_6}\Big(\widehat{\text{dd}_{\text{I}}H}_{ijk_1k_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3} + \widehat{\text{dd}_{\text{II}}H}_{ijk_1k_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3} + \widehat{\text{dd}_{\text{II}}H}_{ik_1jk_2;\delta\tilde{\alpha}_2\tilde{\alpha}_1\tilde{\alpha}_3}\\
&\qquad\qquad+\widehat{\text{dd}_{\text{I}}H}_{jik_1k_2;\tilde{\alpha}_1\delta\tilde{\alpha}_2\tilde{\alpha}_3} + \widehat{\text{dd}_{\text{II}}H}_{jik_1k_2;\tilde{\alpha}_1\delta\tilde{\alpha}_2\tilde{\alpha}_3} + \widehat{\text{dd}_{\text{II}}H}_{jk_1ik_2;\tilde{\alpha}_1\tilde{\alpha}_2\delta\tilde{\alpha}_3}\Big)\\
&\qquad\qquad\times Y_3^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}\,(z_{j;\tilde{\alpha}_4}-y_{j;\tilde{\alpha}_4})\,(z_{k_1;\tilde{\alpha}_5}-y_{k_1;\tilde{\alpha}_5})\,(z_{k_2;\tilde{\alpha}_6}-y_{k_2;\tilde{\alpha}_6})\\
&\qquad+\frac{\eta}{2}\sum_{k_1,k_2,\tilde{\alpha}_2,\tilde{\alpha}_3,\tilde{\alpha}_4,\tilde{\alpha}_5,\tilde{\alpha}_6}\Big(\widehat{\text{dd}_{\text{I}}H}_{ijk_1k_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3} + \widehat{\text{dd}_{\text{I}}H}_{jik_1k_2;\tilde{\alpha}_1\delta\tilde{\alpha}_2\tilde{\alpha}_3} + 2\widehat{\text{dd}_{\text{II}}H}_{ijk_1k_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}\Big)\\
&\qquad\qquad\times Y_4^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}\,(z_{j;\tilde{\alpha}_4}-y_{j;\tilde{\alpha}_4})\,(z_{k_1;\tilde{\alpha}_5}-y_{k_1;\tilde{\alpha}_5})\,(z_{k_2;\tilde{\alpha}_6}-y_{k_2;\tilde{\alpha}_6})\,,
\end{aligned}$$

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

where we introduced additional shorthands

$$Y_2^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4} \equiv \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_3}\widetilde{H}^{\tilde{\alpha}_2\tilde{\alpha}_4} - \sum_{\tilde{\alpha}_5} \widetilde{H}^{\tilde{\alpha}_2\tilde{\alpha}_5} X_{\text{II}}^{\tilde{\alpha}_1\tilde{\alpha}_5\tilde{\alpha}_3\tilde{\alpha}_4}\,, \tag{∞.138}$$

$$\begin{aligned} Y_3^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6} &\equiv \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_4}\widetilde{H}^{\tilde{\alpha}_2\tilde{\alpha}_5}\widetilde{H}^{\tilde{\alpha}_3\tilde{\alpha}_6} - \sum_{\tilde{\alpha}_7} \widetilde{H}^{\tilde{\alpha}_2\tilde{\alpha}_7} X_{\text{II}}^{\tilde{\alpha}_1\tilde{\alpha}_7\tilde{\alpha}_4\tilde{\alpha}_5}\widetilde{H}^{\tilde{\alpha}_3\tilde{\alpha}_6} \\ &\quad - \sum_{\tilde{\alpha}_7} \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_4}\widetilde{H}^{\tilde{\alpha}_3\tilde{\alpha}_7} X_{\text{II}}^{\tilde{\alpha}_2\tilde{\alpha}_7\tilde{\alpha}_5\tilde{\alpha}_6} + \sum_{\tilde{\alpha}_7,\tilde{\alpha}_8,\tilde{\alpha}_9} \widetilde{H}^{\tilde{\alpha}_3\tilde{\alpha}_9} X_{\text{II}}^{\tilde{\alpha}_2\tilde{\alpha}_9\tilde{\alpha}_7\tilde{\alpha}_8} X_{\text{III}}^{\tilde{\alpha}_1\tilde{\alpha}_7\tilde{\alpha}_8\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}\,, \end{aligned} \tag{∞.139}$$

$$Y_4^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6} \equiv \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_4} X_{\text{II}}^{\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_5\tilde{\alpha}_6} - \sum_{\tilde{\alpha}_7,\tilde{\alpha}_8} X_{\text{II}}^{\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_7\tilde{\alpha}_8} X_{\text{III}}^{\tilde{\alpha}_1\tilde{\alpha}_7\tilde{\alpha}_8\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}. \tag{∞.140}$$

Now, it's time to frantically flip through pages and collect everything we computed. Plugging the sums (∞.137), (∞.134), and (∞.132) back into our expression for the sum over the damping force, (∞.131), plugging that back into our expression for the interaction part of the network output, (∞.130), and then combining with the free part of the network output, (∞.93), we obtain our fully-trained solution for finite-width networks trained by gradient descent:

$$\begin{aligned} &z_{i;\delta}(t=\infty) \\ &\quad\equiv z^{\text{F}}_{i;\delta}(t=\infty) + z^{\text{I}}_{i;\delta}(t=\infty) \\ &\quad= z_{i;\delta} - \sum_{j,k,\tilde{\alpha}_1,\tilde{\alpha}_2} \widehat{H}_{ij;\delta\tilde{\alpha}_1} \left(\widehat{H}^{-1}\right)^{\tilde{\alpha}_1\tilde{\alpha}_2}_{jk} \left(z_{k;\tilde{\alpha}_2} - y_{k;\tilde{\alpha}_2}\right) \\ &\qquad+ \sum_{j_1,j_2,\tilde{\alpha}_1,\tilde{\alpha}_2,\tilde{\alpha}_3,\tilde{\alpha}_4} \left[\widehat{\text{d}H}_{j_1ij_2;\tilde{\alpha}_1\delta\tilde{\alpha}_2} - \sum_{\tilde{\alpha}_5,\tilde{\alpha}_6} H_{\delta\tilde{\alpha}_5}\widetilde{H}^{\tilde{\alpha}_5\tilde{\alpha}_6}\widehat{\text{d}H}_{j_1ij_2;\tilde{\alpha}_1\tilde{\alpha}_6\tilde{\alpha}_2}\right] \\ &\qquad\qquad\times Z_{\text{A}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4} \left(z_{j_1;\tilde{\alpha}_3} - y_{j_1;\tilde{\alpha}_3}\right)\left(z_{j_2;\tilde{\alpha}_4} - y_{j_2;\tilde{\alpha}_4}\right) \\ &\qquad+ \sum_{j_1,j_2,\tilde{\alpha}_1,\tilde{\alpha}_2,\tilde{\alpha}_3,\tilde{\alpha}_4} \left[\widehat{\text{d}H}_{ij_1j_2;\delta\tilde{\alpha}_1\tilde{\alpha}_2} - \sum_{\tilde{\alpha}_5,\tilde{\alpha}_6} H_{\delta\tilde{\alpha}_5}\widetilde{H}^{\tilde{\alpha}_5\tilde{\alpha}_6}\widehat{\text{d}H}_{ij_1j_2;\tilde{\alpha}_6\tilde{\alpha}_1\tilde{\alpha}_2}\right] \\ &\qquad\qquad\times Z_{\text{B}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4} \left(z_{j_1;\tilde{\alpha}_3} - y_{j_1;\tilde{\alpha}_3}\right)\left(z_{j_2;\tilde{\alpha}_4} - y_{j_2;\tilde{\alpha}_4}\right) \\ &\qquad+ \sum_{\substack{j_1,j_2,j_3,\\ \tilde{\alpha}_1,\tilde{\alpha}_2,\tilde{\alpha}_3,\tilde{\alpha}_4,\tilde{\alpha}_5,\tilde{\alpha}_6}} \left[\widehat{\text{dd}_{\text{I}}H}_{j_1ij_2j_3;\tilde{\alpha}_1\delta\tilde{\alpha}_2\tilde{\alpha}_3} - \sum_{\tilde{\alpha}_7,\tilde{\alpha}_8} H_{\delta\tilde{\alpha}_7}\widetilde{H}^{\tilde{\alpha}_7\tilde{\alpha}_8}\widehat{\text{dd}_{\text{I}}H}_{j_1ij_2j_3;\tilde{\alpha}_1\tilde{\alpha}_8\tilde{\alpha}_2\tilde{\alpha}_3}\right] \\ &\qquad\qquad\times Z_{\text{IA}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6} \left(z_{j_1;\tilde{\alpha}_4} - y_{j_1;\tilde{\alpha}_4}\right)\left(z_{j_2;\tilde{\alpha}_5} - y_{j_2;\tilde{\alpha}_5}\right)\left(z_{j_3;\tilde{\alpha}_6} - y_{j_3;\tilde{\alpha}_6}\right) \\ &\qquad+ \sum_{\substack{j_1,j_2,j_3,\\ \tilde{\alpha}_1,\tilde{\alpha}_2,\tilde{\alpha}_3,\tilde{\alpha}_4,\tilde{\alpha}_5,\tilde{\alpha}_6}} \left[\widehat{\text{dd}_{\text{I}}H}_{ij_1j_2j_3;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3} - \sum_{\tilde{\alpha}_7,\tilde{\alpha}_8} H_{\delta\tilde{\alpha}_7}\widetilde{H}^{\tilde{\alpha}_7\tilde{\alpha}_8}\widehat{\text{dd}_{\text{I}}H}_{ij_1j_2j_3;\tilde{\alpha}_8\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}\right] \\ &\qquad\qquad\times Z_{\text{IB}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6} \left(z_{j_1;\tilde{\alpha}_4} - y_{j_1;\tilde{\alpha}_4}\right)\left(z_{j_2;\tilde{\alpha}_5} - y_{j_2;\tilde{\alpha}_5}\right)\left(z_{j_3;\tilde{\alpha}_6} - y_{j_3;\tilde{\alpha}_6}\right) \end{aligned} \tag{∞.141}$$

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

$$+\sum_{\substack{j_1,j_2,j_3,\\ \tilde{\alpha}_1,\tilde{\alpha}_2,\tilde{\alpha}_3,\tilde{\alpha}_4,\tilde{\alpha}_5,\tilde{\alpha}_6}}\left[\widehat{\mathrm{dd}_{\mathrm{II}}H}_{j_1j_2ij_3;\tilde{\alpha}_1\tilde{\alpha}_2\delta\tilde{\alpha}_3}-\sum_{\tilde{\alpha}_7,\tilde{\alpha}_8}H_{\delta\tilde{\alpha}_7}\widetilde{H}^{\tilde{\alpha}_7\tilde{\alpha}_8}\widehat{\mathrm{dd}_{\mathrm{II}}H}_{j_1j_2ij_3;\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_8\tilde{\alpha}_3}\right]$$
$$\times Z_{\mathrm{IIA}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}\left(z_{j_1;\tilde{\alpha}_4}-y_{j_1;\tilde{\alpha}_4}\right)\left(z_{j_2;\tilde{\alpha}_5}-y_{j_2;\tilde{\alpha}_5}\right)\left(z_{j_3;\tilde{\alpha}_6}-y_{j_3;\tilde{\alpha}_6}\right)$$
$$+\sum_{\substack{j_1,j_2,j_3,\\ \tilde{\alpha}_1,\tilde{\alpha}_2,\tilde{\alpha}_3,\tilde{\alpha}_4,\tilde{\alpha}_5,\tilde{\alpha}_6}}\left[\widehat{\mathrm{dd}_{\mathrm{II}}H}_{ij_1j_2j_3;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}-\sum_{\tilde{\alpha}_7,\tilde{\alpha}_8}H_{\delta\tilde{\alpha}_7}\widetilde{H}^{\tilde{\alpha}_7\tilde{\alpha}_8}\widehat{\mathrm{dd}_{\mathrm{II}}H}_{ij_1j_2j_3;\tilde{\alpha}_8\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}\right]$$
$$\times Z_{\mathrm{IIB}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}\left(z_{j_1;\tilde{\alpha}_4}-y_{j_1;\tilde{\alpha}_4}\right)\left(z_{j_2;\tilde{\alpha}_5}-y_{j_2;\tilde{\alpha}_5}\right)\left(z_{j_3;\tilde{\alpha}_6}-y_{j_3;\tilde{\alpha}_6}\right)$$
$$+O\left(\frac{1}{n^2}\right).$$

Here we defined our final tensors with our final alphabet letter (and various subscripts),

$$Z_{\mathrm{A}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}\equiv Y_2^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}, \tag{∞.142}$$
$$Z_{\mathrm{B}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}\equiv Y_2^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}+\frac{\eta}{2}X_{\mathrm{II}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}, \tag{∞.143}$$
$$Z_{\mathrm{IA}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}\equiv -Y_3^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}-\frac{\eta}{2}Y_4^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}, \tag{∞.144}$$
$$Z_{\mathrm{IB}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}\equiv -Y_3^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}-\frac{\eta}{2}Y_4^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}$$
$$-\frac{\eta}{2}Y_1^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}-\frac{\eta^2}{6}X_{\mathrm{III}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}, \tag{∞.145}$$
$$Z_{\mathrm{IIA}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}\equiv -Y_3^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}, \tag{∞.146}$$
$$Z_{\mathrm{IIB}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}\equiv -Y_3^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}-Y_3^{\tilde{\alpha}_2\tilde{\alpha}_1\tilde{\alpha}_3\tilde{\alpha}_5\tilde{\alpha}_4\tilde{\alpha}_6}-Y_3^{\tilde{\alpha}_1\tilde{\alpha}_3\tilde{\alpha}_2\tilde{\alpha}_4\tilde{\alpha}_6\tilde{\alpha}_5}$$
$$-\eta Y_4^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}-\eta Y_1^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}, \tag{∞.147}$$

making use of our previous shorthand tensors (∞.135) and (∞.138)–(∞.140).[21] These **algorithm projectors**, (∞.142)–(∞.147), serve to project the initial training error onto two different combinations of the dNTK, two different combinations of the first ddNTK, and two other different combinations of the second ddNTK, all according to the details of the gradient descent algorithm. As a final sanity check, note that as for inputs in the training set, $\delta\to\tilde{\alpha}\in\mathcal{A}$, the quantities in the square brackets in the finite-width solution (∞.141) each vanish, and we recover our fully-trained condition (∞.66).

Before we retire this subsection, let us elaborate on the algorithm dependence. First, note that our two-update solution (∞.79) has the same form as the gradient-descent solution (∞.141), but with different algorithm projectors:

$$Z_{\mathrm{B}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}\equiv\frac{1}{2}\widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_3}\widetilde{H}^{\tilde{\alpha}_2\tilde{\alpha}_4},\qquad Z_{\mathrm{IB}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}\equiv-\frac{1}{6}\widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_4}\widetilde{H}^{\tilde{\alpha}_2\tilde{\alpha}_5}\widetilde{H}^{\tilde{\alpha}_3\tilde{\alpha}_6}, \tag{∞.148}$$

and all the others vanishing. Clearly these algorithms have very different inductive biases! Second, we can study the ODE limit of the dynamics by taking $\eta\to 0$, cf. footnote 17:

[21] Note that for the last of these tensors, (∞.147), in order to coax the various contributions in our solution (∞.141) into the proper form, we used the symmetry of $\widehat{\mathrm{dd}_{\mathrm{II}}H}_{ij_1j_2j_3;\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3}$ and relabeled various dummy sample indices.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

in this case, we see that the ODE dynamics have a solution given by

$$Z_{\mathrm{A}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4} = Z_{\mathrm{B}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4} \equiv Y_2^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}, \tag{∞.149}$$

$$Z_{\mathrm{IA}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6} = Z_{\mathrm{IB}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6} = Z_{\mathrm{IIA}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6} \equiv -Y_3^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6} \tag{∞.150}$$

$$Z_{\mathrm{IIB}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6} \equiv -Y_3^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6} - Y_3^{\tilde{\alpha}_2\tilde{\alpha}_1\tilde{\alpha}_3\tilde{\alpha}_5\tilde{\alpha}_4\tilde{\alpha}_6} - Y_3^{\tilde{\alpha}_1\tilde{\alpha}_3\tilde{\alpha}_2\tilde{\alpha}_4\tilde{\alpha}_6\tilde{\alpha}_5}, \tag{∞.151}$$

where $Y_2^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}$ and $Y_3^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}$ are given by (∞.138) and (∞.139), respectively, the inverting tensor $X_{\mathrm{II}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}$, (∞.107), now satisfies

$$\sum_{\tilde{\alpha}_3,\tilde{\alpha}_4\in\mathcal{A}} X_{\mathrm{II}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}\left(\widetilde{H}_{\tilde{\alpha}_3\tilde{\alpha}_5}\delta_{\tilde{\alpha}_4\tilde{\alpha}_6} + \delta_{\tilde{\alpha}_3\tilde{\alpha}_5}\widetilde{H}_{\tilde{\alpha}_4\tilde{\alpha}_6}\right) = \delta^{\tilde{\alpha}_1}_{\tilde{\alpha}_5}\delta^{\tilde{\alpha}_2}_{\tilde{\alpha}_6}, \tag{∞.152}$$

and the other inverting tensor $X_{\mathrm{II}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}$, (∞.110), now satisfies

$$\begin{aligned}&\delta^{\tilde{\alpha}_1}_{\tilde{\alpha}_7}\delta^{\tilde{\alpha}_2}_{\tilde{\alpha}_8}\delta^{\tilde{\alpha}_3}_{\tilde{\alpha}_9} \\ &\quad= \sum_{\tilde{\alpha}_4,\tilde{\alpha}_5,\tilde{\alpha}_6} X_{\mathrm{III}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}\Big(\widetilde{H}_{\tilde{\alpha}_4\tilde{\alpha}_7}\delta_{\tilde{\alpha}_5\tilde{\alpha}_8}\delta_{\tilde{\alpha}_6\tilde{\alpha}_9} + \delta_{\tilde{\alpha}_4\tilde{\alpha}_7}\widetilde{H}_{\tilde{\alpha}_5\tilde{\alpha}_8}\delta_{\tilde{\alpha}_6\tilde{\alpha}_9} + \delta_{\tilde{\alpha}_4\tilde{\alpha}_7}\delta_{\tilde{\alpha}_5\tilde{\alpha}_8}\widetilde{H}_{\tilde{\alpha}_6\tilde{\alpha}_9}\Big).\end{aligned} \tag{∞.153}$$

This entirely captures the difference between gradient flow and gradient descent for these fully-trained networks. In general, we conjecture that for finite-width networks, at leading order the fully-trained solution takes the universal form of (∞.141), with all of the *algorithm dependence* encoded by the six *algorithm projectors*: $Z_{\mathrm{A}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}$, $Z_{\mathrm{B}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}$, $Z_{\mathrm{IA}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}$, $Z_{\mathrm{IB}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}$, $Z_{\mathrm{IIA}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}$, and $Z_{\mathrm{IIB}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}$.[22]

Finally, let's understand what is meant by the remaining error in our solution (∞.141). It is not really the error of an actual network that is instantiated and then fully trained through many many gradient-descent steps, but instead it is the error in our effective description of such a particular fully-trained network. Of course, our effective theory formalism can compute higher-order corrections, if they're of interest. However, the leading-order finite-width corrections should really be sufficient for most cases: for instance, for a network of depth $L = 10$ layers and of hidden-layer width $n = 100$ neurons each, our effective description will only be off by $\sim (L/n)^2 = 1\%$.

Furthermore, for any theoretical analysis, the main qualitative difference in the solution appears when going from infinite width to finite width, as we go from a free theory to an interacting theory and from linear dynamics to nonlinear dynamics. Thus, the effective theory that gave us the solution (∞.141) really is "as simple... as possible" while still providing an extremely accurate description of real deep learning models.

∞.2.3 Prediction at Finite Width

Having solved the training dynamics in two different ways, we can now rather generally study the predictions of our networks on novel inputs $x_{\dot{\beta}}$ from the test set $\dot{\beta} \in \mathcal{B}$.

[22] More precisely, we conjecture that our conjecture, as stated, holds for the MSE loss. For the cross-entropy loss the solution will take a slightly different form but with a similar partition into an algorithm-independent part and an algorithm-dependent part described by similar algorithm projectors.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

At finite width, the predictions of a fully-trained network are universally governed by the stochastic equation

$$
\begin{aligned}
&z_{i;\dot\beta}(t=T) \qquad\qquad (\infty.154)\\
&\quad = z_{i;\dot\beta} - \sum_{\tilde\alpha_1,\tilde\alpha_2} H_{\dot\beta\tilde\alpha_1}\widetilde{H}^{\tilde\alpha_1\tilde\alpha_2}(z_{i;\tilde\alpha_2}-y_{i;\tilde\alpha_2})\\
&\qquad + \sum_{j,\tilde\alpha_1,\tilde\alpha_2}\left[\widehat{\Delta H}_{ij;\dot\beta\tilde\alpha_1} - \sum_{\tilde\alpha_3,\tilde\alpha_4\in\mathcal{A}} H_{\dot\beta\tilde\alpha_3}\widetilde{H}^{\tilde\alpha_3\tilde\alpha_4}\widehat{\Delta H}_{ij;\tilde\alpha_4\tilde\alpha_1}\right]\widetilde{H}^{\tilde\alpha_1\tilde\alpha_2}(z_{j;\tilde\alpha_2}-y_{j;\tilde\alpha_2})\\
&\quad - \sum_{\substack{j,k\\ \tilde\alpha_1,\ldots,\tilde\alpha_4}}\left[\widehat{\Delta H}_{ij;\dot\beta\tilde\alpha_1} - \sum_{\tilde\alpha_5,\tilde\alpha_6} H_{\dot\beta\tilde\alpha_5}\widetilde{H}^{\tilde\alpha_5\tilde\alpha_6}\widehat{\Delta H}_{ij;\tilde\alpha_6\tilde\alpha_1}\right]\widetilde{H}^{\tilde\alpha_1\tilde\alpha_2}\widehat{\Delta H}_{jk;\tilde\alpha_2\tilde\alpha_3}\widetilde{H}^{\tilde\alpha_3\tilde\alpha_4}(z_{k;\tilde\alpha_4}-y_{k;\tilde\alpha_4})\\
&\qquad + \sum_{j_1,j_2,\tilde\alpha_1,\tilde\alpha_2,\tilde\alpha_3,\tilde\alpha_4}\left[\widehat{\mathrm{d}H}_{j_1ij_2;\tilde\alpha_1\dot\beta\tilde\alpha_2} - \sum_{\tilde\alpha_5,\tilde\alpha_6} H_{\dot\beta\tilde\alpha_5}\widetilde{H}^{\tilde\alpha_5\tilde\alpha_6}\widehat{\mathrm{d}H}_{j_1ij_2;\tilde\alpha_1\tilde\alpha_6\tilde\alpha_2}\right]\\
&\qquad\qquad \times Z_{\mathrm{A}}^{\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3\tilde\alpha_4}\,(z_{j_1;\tilde\alpha_3}-y_{j_1;\tilde\alpha_3})\,(z_{j_2;\tilde\alpha_4}-y_{j_2;\tilde\alpha_4})\\
&\qquad + \sum_{j_1,j_2,\tilde\alpha_1,\tilde\alpha_2,\tilde\alpha_3,\tilde\alpha_4}\left[\widehat{\mathrm{d}H}_{ij_1j_2;\dot\beta\tilde\alpha_1\tilde\alpha_2} - \sum_{\tilde\alpha_5,\tilde\alpha_6} H_{\dot\beta\tilde\alpha_5}\widetilde{H}^{\tilde\alpha_5\tilde\alpha_6}\widehat{\mathrm{d}H}_{ij_1j_2;\tilde\alpha_6\tilde\alpha_1\tilde\alpha_2}\right]\\
&\qquad\qquad \times Z_{\mathrm{B}}^{\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3\tilde\alpha_4}\,(z_{j_1;\tilde\alpha_3}-y_{j_1;\tilde\alpha_3})\,(z_{j_2;\tilde\alpha_4}-y_{j_2;\tilde\alpha_4})\\
&\qquad + \sum_{\substack{j_1,j_2,j_3,\\ \tilde\alpha_1,\tilde\alpha_2,\tilde\alpha_3,\tilde\alpha_4,\tilde\alpha_5,\tilde\alpha_6}}\left[\widehat{\mathrm{dd_I}H}_{j_1ij_2j_3;\tilde\alpha_1\dot\beta\tilde\alpha_2\tilde\alpha_3} - \sum_{\tilde\alpha_7,\tilde\alpha_8} H_{\dot\beta\tilde\alpha_7}\widetilde{H}^{\tilde\alpha_7\tilde\alpha_8}\widehat{\mathrm{dd_I}H}_{j_1ij_2j_3;\tilde\alpha_1\tilde\alpha_8\tilde\alpha_2\tilde\alpha_3}\right]\\
&\qquad\qquad \times Z_{\mathrm{IA}}^{\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3\tilde\alpha_4\tilde\alpha_5\tilde\alpha_6}\,(z_{j_1;\tilde\alpha_4}-y_{j_1;\tilde\alpha_4})\,(z_{j_2;\tilde\alpha_5}-y_{j_2;\tilde\alpha_5})\,(z_{j_3;\tilde\alpha_6}-y_{j_3;\tilde\alpha_6})\\
&\qquad + \sum_{\substack{j_1,j_2,j_3,\\ \tilde\alpha_1,\tilde\alpha_2,\tilde\alpha_3,\tilde\alpha_4,\tilde\alpha_5,\tilde\alpha_6}}\left[\widehat{\mathrm{dd_I}H}_{ij_1j_2j_3;\dot\beta\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3} - \sum_{\tilde\alpha_7,\tilde\alpha_8} H_{\dot\beta\tilde\alpha_7}\widetilde{H}^{\tilde\alpha_7\tilde\alpha_8}\widehat{\mathrm{dd_I}H}_{ij_1j_2j_3;\tilde\alpha_8\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3}\right]\\
&\qquad\qquad \times Z_{\mathrm{IB}}^{\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3\tilde\alpha_4\tilde\alpha_5\tilde\alpha_6}\,(z_{j_1;\tilde\alpha_4}-y_{j_1;\tilde\alpha_4})\,(z_{j_2;\tilde\alpha_5}-y_{j_2;\tilde\alpha_5})\,(z_{j_3;\tilde\alpha_6}-y_{j_3;\tilde\alpha_6})\\
&\qquad + \sum_{\substack{j_1,j_2,j_3,\\ \tilde\alpha_1,\tilde\alpha_2,\tilde\alpha_3,\tilde\alpha_4,\tilde\alpha_5,\tilde\alpha_6}}\left[\widehat{\mathrm{dd_{II}}H}_{j_1j_2ij_3;\tilde\alpha_1\tilde\alpha_2\dot\beta\tilde\alpha_3} - \sum_{\tilde\alpha_7,\tilde\alpha_8} H_{\dot\beta\tilde\alpha_7}\widetilde{H}^{\tilde\alpha_7\tilde\alpha_8}\widehat{\mathrm{dd_{II}}H}_{j_1j_2ij_3;\tilde\alpha_1\tilde\alpha_2\tilde\alpha_8\tilde\alpha_3}\right]\\
&\qquad\qquad \times Z_{\mathrm{IIA}}^{\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3\tilde\alpha_4\tilde\alpha_5\tilde\alpha_6}\,(z_{j_1;\tilde\alpha_4}-y_{j_1;\tilde\alpha_4})\,(z_{j_2;\tilde\alpha_5}-y_{j_2;\tilde\alpha_5})\,(z_{j_3;\tilde\alpha_6}-y_{j_3;\tilde\alpha_6})\\
&\qquad + \sum_{\substack{j_1,j_2,j_3,\\ \tilde\alpha_1,\tilde\alpha_2,\tilde\alpha_3,\tilde\alpha_4,\tilde\alpha_5,\tilde\alpha_6}}\left[\widehat{\mathrm{dd_{II}}H}_{ij_1j_2j_3;\dot\beta\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3} - \sum_{\tilde\alpha_7,\tilde\alpha_8} H_{\dot\beta\tilde\alpha_7}\widetilde{H}^{\tilde\alpha_7\tilde\alpha_8}\widehat{\mathrm{dd_{II}}H}_{ij_1j_2j_3;\tilde\alpha_8\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3}\right]\\
&\qquad\qquad \times Z_{\mathrm{IIB}}^{\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3\tilde\alpha_4\tilde\alpha_5\tilde\alpha_6}\,(z_{j_1;\tilde\alpha_4}-y_{j_1;\tilde\alpha_4})\,(z_{j_2;\tilde\alpha_5}-y_{j_2;\tilde\alpha_5})\,(z_{j_3;\tilde\alpha_6}-y_{j_3;\tilde\alpha_6})\\
&\qquad + O\left(\frac{1}{n^2}\right).
\end{aligned}
$$

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

This formula could borderline fit on a T-shirt.[23] For this expression, we've expanded the complete inverse of the stochastic NTK sub-tensor as per (∞.71); in particular, the second line is more or less the infinite-width kernel prediction (10.39), and the terms on the third and fourth lines are finite-width corrections due to NTK fluctuations. Further, the algorithm projectors (∞.142)–(∞.147) contain all the finite-width algorithm dependence of the solution, and thus this general solution (∞.154) can describe all our explicit solutions, whether we train in two steps (∞.79), in many many steps (∞.141), or with any other choice of optimization algorithm that uses the MSE loss.

[23] To better facilitate such brand awareness, first recall that for nearly-kernel methods we were able to compress the model predictions in terms of a *trained kernel* (11.142); similarly, we can compress the predictions of finite-width networks (∞.154) in terms of a **trained NTK**, $H^{\sharp}_{ij;\delta\tilde{\alpha}}$, as

$$z_{i;\dot{\beta}}(t=T) = z_{i;\dot{\beta}} - \sum_{j,k,\tilde{\alpha}_1,\tilde{\alpha}_2} H^{\sharp}_{ij;\dot{\beta}\tilde{\alpha}_1} \widetilde{H^{\sharp}}^{\tilde{\alpha}_1\tilde{\alpha}_2}_{jk} \left(z_{k;\tilde{\alpha}_2} - y_{k;\tilde{\alpha}_2}\right) + O\left(\frac{1}{n^2}\right), \tag{∞.155}$$

taking the form of a *(neural tangent) kernel prediction* (10.39). To see how this works, let us decompose the trained NTK into free and training-dependent terms:

$$H^{\sharp}_{ij;\delta\tilde{\alpha}} \equiv \widehat{H}_{ij;\delta\tilde{\alpha}} + \mathbb{H}_{ij;\delta\tilde{\alpha}}. \tag{∞.156}$$

Please don't confuse this decomposition with our earlier decomposition (∞.95): that former one was convenient for solving the training dynamics, while this new one is useful for determining the trained NTK in (∞.155). Considering the inverse of the trained NTK restricted to the training set only,

$$\widetilde{H^{\sharp}}^{\tilde{\alpha}_1\tilde{\alpha}_2}_{jk} \equiv \left(\widehat{H}^{-1}\right)^{\tilde{\alpha}_1\tilde{\alpha}_2}_{jk} - \sum_{\tilde{\alpha}_3,\tilde{\alpha}_4} \mathbb{H}_{jk;\tilde{\alpha}_3\tilde{\alpha}_4} \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_3} \widetilde{H}^{\tilde{\alpha}_2\tilde{\alpha}_4} + O\left(\frac{1}{n^2}\right), \tag{∞.157}$$

and plugging it along with the decomposition (∞.156) into the formula (∞.155), we get

$$\begin{aligned} z_{i;\dot{\beta}}(t=T) = z_{i;\dot{\beta}} &- \sum_{j,k,\tilde{\alpha}_1,\tilde{\alpha}_2} \widehat{H}_{ij;\dot{\beta}\tilde{\alpha}_1} \left(\widehat{H}^{-1}\right)^{\tilde{\alpha}_1\tilde{\alpha}_2}_{jk} \left(z_{k;\tilde{\alpha}_2} - y_{k;\tilde{\alpha}_2}\right) \\ &- \sum_{j,\tilde{\alpha}_1,\tilde{\alpha}_2} \left[\mathbb{H}_{ij;\dot{\beta}\tilde{\alpha}_1} - \sum_{\tilde{\alpha}_3,\tilde{\alpha}_4} H_{\dot{\beta}\tilde{\alpha}_3} \widetilde{H}^{\tilde{\alpha}_3\tilde{\alpha}_4} \mathbb{H}_{ij;\tilde{\alpha}_4\tilde{\alpha}_1}\right] \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_2} \left(z_{j;\tilde{\alpha}_2} - y_{j;\tilde{\alpha}_2}\right) + O\left(\frac{1}{n^2}\right). \end{aligned} \tag{∞.158}$$

The terms on the first line of the right-hand side of this expression give the free contribution to the solution, (∞.93), while the terms on the second line give the interacting contribution, (∞.130), encapsulating the effect of nontrivial representation learning at finite width. The specific form of $H^{\sharp}_{ij;\delta\tilde{\alpha}}$ can be found by matching the terms on the right-hand sides of (∞.158) and (∞.154). You can also express the training-dependent part, $\mathbb{H}_{ij;\delta\tilde{\alpha}}$, implicitly in terms of the damping force (∞.119) as

$$\eta \sum_{s=0}^{\infty} \mathbb{F}_{i;\delta}(s) = - \sum_{j,\tilde{\alpha}_1,\tilde{\alpha}_2} \mathbb{H}_{ij;\delta\tilde{\alpha}_1} \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_2} \left(z_{j;\tilde{\alpha}_2} - y_{j;\tilde{\alpha}_2}\right), \tag{∞.159}$$

as is clear by comparing (∞.158) with (∞.130). This implicit expression also makes it clear that the form of $\mathbb{H}_{ij;\delta\tilde{\alpha}}$ isn't unique and can always be adjusted by the addition of a term orthogonal to $\sum_{\tilde{\alpha}_2} \widetilde{H}^{\tilde{\alpha}_1\tilde{\alpha}_2} \left(z_{j;\tilde{\alpha}_2} - y_{j;\tilde{\alpha}_2}\right)$. In any event, with the trained NTK you can now fit the finite-width prediction formula (∞.155) on any newborn AI's onesie.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

Furthermore, this solution (∞.154) describes the predictions of a *particular* network from our ensemble: the instantiation-to-instantiation difference is encoded in the particular initial network output, z, the NTK fluctuation, $\widehat{\Delta H}$, the dNTK, $\widehat{\mathrm{d}H}$, the first ddNTK, $\widehat{\mathrm{dd}_{\mathrm{I}}H}$, and the second ddNTK, $\widehat{\mathrm{dd}_{\mathrm{II}}H}$. Since we know explicitly the statistics of these variables at initialization, we can also analyze the statistics of the fully-trained distribution in full. With an eye toward a discussion of the depth dependence of these statistics, we'll now revive the layer indices.

First and foremost, the mean prediction is given by

$$\begin{aligned} m_{i;\dot\beta} &\equiv \mathbb{E}\left[z^{(L)}_{i;\dot\beta}(T)\right] \qquad (\infty.160)\\ &= m^{\mathrm{NTK}}_{i;\dot\beta} + \frac{1}{n_{L-1}}\left(m^{\Delta\mathrm{NTK}}_{i;\dot\beta} + m^{\mathrm{dNTK}}_{i;\dot\beta} + m^{\mathrm{ddNTK\text{-}I}}_{i;\dot\beta} + m^{\mathrm{ddNTK\text{-}II}}_{i;\dot\beta}\right)\\ &\quad - \frac{1}{n_{L-1}}\sum_{\tilde\alpha_1,\tilde\alpha_2} H^{(L)}_{\dot\beta\tilde\alpha_1}\widetilde{H}^{\tilde\alpha_1\tilde\alpha_2}_{(L)}\left(m^{\Delta\mathrm{NTK}}_{i;\tilde\alpha_2} + m^{\mathrm{dNTK}}_{i;\tilde\alpha_2} + m^{\mathrm{ddNTK\text{-}I}}_{i;\tilde\alpha_2} + m^{\mathrm{ddNTK\text{-}II}}_{i;\tilde\alpha_2}\right)\\ &\quad + O\left(\frac{1}{n^2}\right), \end{aligned}$$

where the first term is the (neural tangent) kernel prediction

$$m^{\mathrm{NTK}}_{i;\dot\beta} \equiv \sum_{\tilde\alpha_1,\tilde\alpha_2} H^{(L)}_{\dot\beta\tilde\alpha_1}\widetilde{H}^{\tilde\alpha_1\tilde\alpha_2}_{(L)} y_{i;\tilde\alpha_2}, \qquad (\infty.161)$$

and the four other kinds of terms come from the leading-order finite-width correction. Specifically, *(i)* the fluctuation of the NTK gives

$$m^{\Delta\mathrm{NTK}}_{i;\delta} \equiv \sum_{\tilde\alpha_1,\ldots,\tilde\alpha_4}\left(A^{(L)}_{(\delta\tilde\alpha_1)(\tilde\alpha_2\tilde\alpha_3)} + B^{(L)}_{\delta\tilde\alpha_2\tilde\alpha_1\tilde\alpha_3} + n_L B^{(L)}_{\delta\tilde\alpha_3\tilde\alpha_1\tilde\alpha_2}\right)\widetilde{H}^{\tilde\alpha_1\tilde\alpha_2}_{(L)}\widetilde{H}^{\tilde\alpha_3\tilde\alpha_4}_{(L)} y_{i;\tilde\alpha_4}\,, \quad (\infty.162)$$

where we used (8.82) to evaluate the NTK variance in terms of our decomposition into tensors $A^{(L)}$ and $B^{(L)}$; *(ii)* the dNTK gives

$$\begin{aligned} m^{\mathrm{dNTK}}_{i;\delta} \equiv -\sum_{\tilde\alpha_1,\ldots,\tilde\alpha_4}\Bigg[&2\left(P^{(L)}_{\delta\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3} + Q^{(L)}_{\delta\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3} + n_L Q^{(L)}_{\delta\tilde\alpha_2\tilde\alpha_1\tilde\alpha_3}\right)Z^{\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3\tilde\alpha_4}_{\mathrm{B}} \qquad (\infty.163)\\ &+\left(n_L P^{(L)}_{\tilde\alpha_1\delta\tilde\alpha_2\tilde\alpha_3} + Q^{(L)}_{\tilde\alpha_1\delta\tilde\alpha_2\tilde\alpha_3} + Q^{(L)}_{\tilde\alpha_1\tilde\alpha_2\delta\tilde\alpha_3}\right)Z^{\tilde\alpha_1\tilde\alpha_2\tilde\alpha_3\tilde\alpha_4}_{\mathrm{A}}\\ &+\left(P^{(L)}_{\tilde\alpha_1\delta\tilde\alpha_2\tilde\alpha_3} + n_L Q^{(L)}_{\tilde\alpha_1\delta\tilde\alpha_2\tilde\alpha_3} + Q^{(L)}_{\tilde\alpha_1\tilde\alpha_2\delta\tilde\alpha_3}\right)Z^{\tilde\alpha_1\tilde\alpha_2\tilde\alpha_4\tilde\alpha_3}_{\mathrm{A}}\Bigg] y_{i;\tilde\alpha_4}\,, \end{aligned}$$

where we used (11.42) to evaluate the dNTK–preactivation cross correlators in terms of our decomposition into tensors $P^{(L)}$ and $Q^{(L)}$; *(iii)* the first ddNTK gives

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

$$m^{\text{ddNTK-I}}_{i;\delta} \tag{∞.164}$$
$$\equiv -\sum_{\tilde{\alpha}_1,\ldots,\tilde{\alpha}_6} \Big[R^{(L)}_{\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3} \Big(Z^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}_{\text{IB}} + Z^{\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_1\tilde{\alpha}_5\tilde{\alpha}_6\tilde{\alpha}_4}_{\text{IB}} + Z^{\tilde{\alpha}_3\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_6\tilde{\alpha}_4\tilde{\alpha}_5}_{\text{IB}} \Big)$$
$$+ R^{(L)}_{\tilde{\alpha}_1\delta\tilde{\alpha}_2\tilde{\alpha}_3} Z^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}_{\text{IA}} + R^{(L)}_{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\delta} Z^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_5\tilde{\alpha}_6\tilde{\alpha}_4}_{\text{IA}} + R^{(L)}_{\tilde{\alpha}_1\tilde{\alpha}_3\delta\tilde{\alpha}_2} Z^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_6\tilde{\alpha}_4\tilde{\alpha}_5}_{\text{IA}} \Big]$$
$$\times \Big[y_{i;\tilde{\alpha}_4} \Big(\sum_j y_{j;\tilde{\alpha}_5} y_{j;\tilde{\alpha}_6} + n_L K^{(L)}_{\tilde{\alpha}_5\tilde{\alpha}_6} \Big) + y_{i;\tilde{\alpha}_5} K^{(L)}_{\tilde{\alpha}_6\tilde{\alpha}_4} + y_{i;\tilde{\alpha}_6} K^{(L)}_{\tilde{\alpha}_4\tilde{\alpha}_5} \Big] ,$$

where we used (∞.11) to evaluate the first ddNTK mean in terms of decomposition into the tensor $R^{(L)}$; and *(iv)* the second ddNTK gives

$$m^{\text{ddNTK-II}}_{i;\delta} \tag{∞.165}$$
$$\equiv -\sum_{\tilde{\alpha}_1,\ldots,\tilde{\alpha}_6} \Big[S^{(L)}_{\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3} Z^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}_{\text{IIB}} + T^{(L)}_{\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3} Z^{\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_1\tilde{\alpha}_5\tilde{\alpha}_6\tilde{\alpha}_4}_{\text{IIB}} + U^{(L)}_{\delta\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3} Z^{\tilde{\alpha}_3\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_6\tilde{\alpha}_4\tilde{\alpha}_5}_{\text{IIB}}$$
$$+ S^{(L)}_{\tilde{\alpha}_1\tilde{\alpha}_2\delta\tilde{\alpha}_3} Z^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_5\tilde{\alpha}_6\tilde{\alpha}_4}_{\text{IIA}} + T^{(L)}_{\tilde{\alpha}_1\delta\tilde{\alpha}_3\tilde{\alpha}_2} Z^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4\tilde{\alpha}_5\tilde{\alpha}_6}_{\text{IIA}} + U^{(L)}_{\tilde{\alpha}_1\tilde{\alpha}_3\tilde{\alpha}_2\delta} Z^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_6\tilde{\alpha}_4\tilde{\alpha}_5}_{\text{IIA}} \Big]$$
$$\times \Big[y_{i;\tilde{\alpha}_4} \Big(\sum_j y_{j;\tilde{\alpha}_5} y_{j;\tilde{\alpha}_6} + n_L K^{(L)}_{\tilde{\alpha}_5\tilde{\alpha}_6} \Big) + y_{i;\tilde{\alpha}_5} K^{(L)}_{\tilde{\alpha}_6\tilde{\alpha}_4} + y_{i;\tilde{\alpha}_6} K^{(L)}_{\tilde{\alpha}_4\tilde{\alpha}_5} \Big] ,$$

where we used (∞.12) to evaluate the second ddNTK mean in terms of decomposition into the tensors $S^{(L)}$, $T^{(L)}$, and $U^{(L)}$.

Interestingly, we see that not only does the mean prediction depend on the NTK differentials – as we expect, given the nontrivial representation learning at finite width – but also it depends on the NTK variance as well, cf. (∞.162). This is natural as each network fits the training data with its own *particular* NTK, and so the resulting fully-trained particular output (∞.154) depends on the NTK fluctuation, as we've already emphasized. In general, it is really important to understand the tradeoffs between both contributions, and in the next subsubsection we will return to comment on this interplay between random fluctuations and directed representation learning in the overall ensemble.

In addition, looking at the terms that are cubic in $y_{i;\tilde{\alpha}}$ in the contributions from the ddNTKs, (∞.164) and (∞.165), we see that the j-th component of the observed outputs influences the i-th component of the mean prediction for $i \neq j$. Just as we saw for the mean prediction of Bayesian inference, (6.88), this is one consequence of the wiring property of finite-width neural networks. To see another manifestation of this wiring property, let's consider the covariance:

$$\text{Cov}\Big[z^{(L)}_{i_1;\dot{\beta}_1}(T),\, z^{(L)}_{i_2;\dot{\beta}_2}(T) \Big] \equiv \mathbb{E}\Big[z^{(L)}_{i_1;\dot{\beta}_1}(T)\, z^{(L)}_{i_2;\dot{\beta}_2}(T) \Big] - \mathbb{E}\Big[z^{(L)}_{i_1;\dot{\beta}_1}(T) \Big] \mathbb{E}\Big[z^{(L)}_{i_2;\dot{\beta}_2}(T) \Big] . \tag{∞.166}$$

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

While we won't print this quantity in full – the full expression doesn't really play nicely with the constraints of the page – you can easily extract insight by considering some specific contributions. For instance, we can see the imprint of output-component wiring by looking at the following contribution to the covariance,

$$\sum_{\substack{j_1,j_2\\ \tilde{\alpha}_1,\ldots,\tilde{\alpha}_4}} \mathbb{E}\left[z^{(L)}_{i_2;\dot{\beta}_2}\widehat{\mathrm{d}H}^{(L)}_{i_1 j_1 j_2;\dot{\beta}_1\tilde{\alpha}_1\tilde{\alpha}_2}\right] Z_{\mathrm{B}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}\mathbb{E}\left[\left(z^{(L)}_{j_1;\tilde{\alpha}_3}-y_{j_1;\tilde{\alpha}_3}\right)\left(z^{(L)}_{j_2;\tilde{\alpha}_4}-y_{j_2;\tilde{\alpha}_4}\right)\right]$$

$$= \frac{1}{n_{L-1}}\delta_{i_1 i_2}\sum_{\tilde{\alpha}_1,\ldots,\tilde{\alpha}_4}\Bigg[\left(n_L P^{(L)}_{\dot{\beta}_1\tilde{\alpha}_1\tilde{\alpha}_2\dot{\beta}_2}+Q^{(L)}_{\dot{\beta}_1\tilde{\alpha}_1\tilde{\alpha}_2\dot{\beta}_2}+Q^{(L)}_{\dot{\beta}_1\tilde{\alpha}_2\tilde{\alpha}_1\dot{\beta}_2}\right)G^{(L)}_{\tilde{\alpha}_3\tilde{\alpha}_4}$$

$$+P^{(L)}_{\dot{\beta}_1\tilde{\alpha}_1\tilde{\alpha}_2\dot{\beta}_2}\Big(\sum_j y_{j;\tilde{\alpha}_3}y_{j;\tilde{\alpha}_4}\Big)\Bigg]Z_{\mathrm{B}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}$$

$$+\frac{1}{n_{L-1}}\sum_{\tilde{\alpha}_1,\ldots,\tilde{\alpha}_4}Q^{(L)}_{\dot{\beta}_1\tilde{\alpha}_1\tilde{\alpha}_2\dot{\beta}_2}Z_{\mathrm{B}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}\left(y_{i_1;\tilde{\alpha}_3}y_{i_2;\tilde{\alpha}_4}+y_{i_1;\tilde{\alpha}_4}y_{i_2;\tilde{\alpha}_3}\right), \qquad (\infty.167)$$

which comes from the cross correlation between a factor of $z^{(L)}_{i_2;\dot{\beta}_2}$ from $z^{(L)}_{i_2;\dot{\beta}_2}(T)$ and one of the dNTK terms from $z^{(L)}_{i_1;\dot{\beta}_1}(T)$ that involves the algorithm projector $Z_{\mathrm{B}}^{\tilde{\alpha}_1\tilde{\alpha}_2\tilde{\alpha}_3\tilde{\alpha}_4}$. In particular, we see that wiring is exhibited in the last term on the final line when the true outputs have nonzero components for both i_1 and i_2. In this case, this correlation (∞.167) implies that the test-set predictions for $z^{(L)}_{i_1;\dot{\beta}_1}(T)$ can be shifted given $z^{(L)}_{i_2;\dot{\beta}_2}(T)$, for $i_1 \neq i_2$: cf. our related discussion of Hebbian learning in terms of the *fire-together* inductive bias in the finite-width prior in §6.4.1 and then our following discussion of how that leads to the posterior distribution's *wiring together* in §6.4.2. Here, the presence of this wiring in the covariance of the solution means that this contribution to wiring occurs differently for each network in the ensemble: the fluctuations from instantiation to instantiation of the parameters at initialization break the permutation symmetry among the output neurons. This type of wiring suggests that each network is able to use the fluctuations between the different initial output components to its advantage in order to correlate such components together over the course of learning.

More broadly, unlike the kernel prediction at infinite width (10.39), the prediction at finite width (∞.154) now has non-Gaussian statistics. In particular, since finite-width prediction is a nontrivial functional of the network output, the NTK, the dNTK, and the ddNTKs – all at initialization – and since we know that the joint distribution of those quantities $p(z^{(L)}, \widehat{H}^{(L)}, \widehat{\mathrm{d}H}^{(L)}, \widehat{\mathrm{dd}_{\mathrm{I}}H}^{(L)}, \widehat{\mathrm{dd}_{\mathrm{II}}H}^{(L)}|\mathcal{D})$ is a *nearly-Gaussian distribution*, then so is the fully-trained distribution $p(z^{(L)}(T)\,|\mathcal{D})$. In particular, there are nontrivial higher-point connected correlators; the explicit expressions of such correlators are challenging to display in any media format, though all the information needed to do so is contained in (∞.154), and it's not that hard to zero in on any particular term of interest. The information carried in such correlators probably contains useful insight into some of the behavior of fully-trained networks and is likely worth further consideration.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

Generalization at Finite Width

Having discussed many of the qualitative differences between the finite-width prediction (∞.154) and the infinite-width kernel prediction (10.39), now let's make some quantitative statements. In order to get a high-level understanding of how generalization is modified at finite width as compared to our extensive infinite-width analysis in §10.3, we need to determine the size of the relative importance of the finite-width corrections to our predictions, namely how these corrections depend on the widths of the hidden layers n_ℓ and the depth of the network L. In particular, we want to understand corrections to our generalized bias–variance tradeoff (10.52) at finite width.

Let's start by considering the bias term, $m_{i;\dot\beta} - y_{i;\dot\beta}$, for which we just need to look at the mean prediction $m_{i;\dot\beta}$ in (∞.160). In particular, we need to compare the first term, (∞.161), which more or less corresponds to the infinite-width kernel prediction $m^{\infty}_{i;\dot\beta}$ – up to the subleading and unimportant correction to the NTK mean – against the other set of finite-width contributions to the mean, (∞.162)–(∞.165).

Now, among the finite-width contributions in (∞.160), the terms inside the first set of large parentheses have a sample index corresponding to a test input, $\dot\beta$, in each of the tensors $A^{(L)}$, $B^{(L)}$, $P^{(L)}$, $Q^{(L)}$, $R^{(L)}$, $S^{(L)}$, $T^{(L)}$, and $U^{(L)}$. Thus, even for a training set with one training sample as we studied in §10.3.1, the particular details of these terms require an understanding of asymptotic multiple-input solutions of the recursions for the NTK variance, the preactivation–dNTK cross correlation, and the ddNTK mean; such an analysis is kind of annoying and was previously left as an adventure for thrill seekers, with a brief instruction manual for those interested buried in footnote 8 of §11.3. Unfortunately, we have no plans here to update that manual any further, and we don't expect to miss much by not doing so.[24]

In contrast, the terms inside the second set of large parentheses in (∞.160) can be analyzed with the results we already have in hand. Thus, to nonlinearly gain a large amount of intuition with a small amount of effort, let's compare the infinite-width term (∞.161) against only this last set of terms. In particular, since both of these terms are preceded by a common prefactor $\sum_{\tilde\alpha_1,\tilde\alpha_2} H^{(L)}_{\dot\beta\tilde\alpha_1} \widetilde{H}^{\tilde\alpha_1\tilde\alpha_2}_{(L)}$ – whose effect was analyzed in §10.3 – we simply need to understand the depth and width dependence of the tensors inside the second set of brackets in (∞.160). Here we'll evaluate these tensors for a single training set input $x_{\tilde\alpha} = x$, dropping the training sample indices for the rest of this analysis for notational simplicity.

To see the physics, it's simplest to look at the ODE limit of many many steps of gradient descent, $\eta \searrow 0$, for which the algorithm projectors take particularly simple form,

[24] In particular, we expect that this first set of terms will be qualitatively similar to the second set of terms that we will discuss in the next paragraph, though the particular order-one-level details may differ.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

$$Z_{\text{A}} = Z_{\text{B}} = \frac{1}{2\left(\widetilde{H}^{(L)}\right)^2}\,, \quad Z_{\text{IA}} = Z_{\text{IB}} = Z_{\text{IIA}} = \frac{-1}{6\left(\widetilde{H}^{(L)}\right)^3}\,, \quad Z_{\text{IIB}} = \frac{-1}{2\left(\widetilde{H}^{(L)}\right)^3}; \tag{∞.168}$$

cf. (∞.138), (∞.139), and (∞.149)–(∞.153) to derive these expressions.[25] Substituting these projectors (∞.168) into the dNTK and ddNTK contributions to the mean prediction, (∞.163)–(∞.165), and then considering the $1/n$ part of the mean prediction $m_{i;\dot{\beta}}$ (∞.160), we see that all the individual terms are proportional to one of the following dimensionless ratios:

$$\frac{A^{(L)}}{n_{L-1}\left(\widetilde{H}^{(L)}\right)^2}\,, \quad \frac{B^{(L)}}{n_{L-1}\left(\widetilde{H}^{(L)}\right)^2}\,, \quad \frac{P^{(L)}}{n_{L-1}\left(\widetilde{H}^{(L)}\right)^2}\,, \quad \frac{Q^{(L)}}{n_{L-1}\left(\widetilde{H}^{(L)}\right)^2}, \tag{∞.169}$$

$$\frac{R^{(L)}K^{(L)}}{n_{L-1}\left(\widetilde{H}^{(L)}\right)^3}\,, \quad \frac{S^{(L)}K^{(L)}}{n_{L-1}\left(\widetilde{H}^{(L)}\right)^3}\,, \quad \frac{T^{(L)}K^{(L)}}{n_{L-1}\left(\widetilde{H}^{(L)}\right)^3}\,, \quad \frac{U^{(L)}K^{(L)}}{n_{L-1}\left(\widetilde{H}^{(L)}\right)^3}.$$

Thus, these eight ratios determine the size of the finite-width effects as compared to the leading infinite-width term.[26]

Importantly, recalling our scaling laws for the NTK variance, (9.27), for the dNTK–preactivation cross correlation, (11.65), and for the ddNTK–preactivation cross correlations, (∞.21) and (∞.22), we see that each of these dimensionless ratios will scale like the depth-to-width ratio L/n (except for the subdominant $U^{(L)}$ contribution). Thus, overall we should find for the finite-width corrections that

$$m_{i;\dot{\beta}} - m^{\infty}_{i;\dot{\beta}} = O\left(\frac{L}{n}\right), \tag{∞.170}$$

[25] Backing off the ODE limit, for many many steps of gradient descent with a finite learning rate η, the single-input dNTK algorithm projectors, (∞.142) and (∞.143), are instead given by

$$Z_{\text{A}} = \frac{1}{\left(\widetilde{H}^{(L)}\right)^2}\left(\frac{1-\eta\widetilde{H}^{(L)}}{2-\eta\widetilde{H}^{(L)}}\right), \qquad Z_{\text{B}} = \frac{1}{2\left(\widetilde{H}^{(L)}\right)^2}. \tag{∞.171}$$

In conjunction with similar single-input limits of the ddNTK algorithm projectors, this will mean that the same set of ratios, (∞.169), determine the generalization error bias term, though now with an additional η dependence. In particular, the projector Z_{A} diverges as the global learning rate approaches from below $\eta \nearrow 2/\widetilde{H}^{(L)}$. This divergence is expected: as we noted in footnote 16, we need $||I - \eta\widehat{H}||_\infty < 1$ for the training dynamics to converge after many many steps, $t \to \infty$. If you check, you'll also see that some of the ddNTK algorithm projectors have the same divergence.

[26] Incidentally, the form of the final two ratios on the first line of (∞.169) is the ultimate justification for why in (11.62) we divided the dNTK–preactivation cross-correlation tensors by two factors of the NTK mean, cf. our discussion of dimensional analysis around (11.64). Similarly the form of the four ratios on the second line of (∞.169) is the ultimate justification for the ratios (∞.21) and (∞.22). For this latter set of ratios, we should have really written $K^{(L)} + n_L^{-1}\sum_j y_j^2$ instead of just $K^{(L)}$, especially when $K^{(L)}$ behaves as a nontrivial power law in L; in such cases, we should really rescale the target output $y_{i;\tilde{\alpha}}$ by the power-law factor $L^{-p_0/2}$ as discussed around (∞.26).

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

where the exact order-one constant is not important. What *is* important is that we confirmed our expectation for the overall scaling of this correction, going as the *cutoff* of our effective theory: $r \equiv L/n$. Similarly, we could do an analysis for the variance term of the generalization error by looking at the covariance (∞.166), which will be a whole lot of work with very little payoff; such an analysis would merely confirm that the leading finite-width corrections will again be of order $O(L/n)$.

In general, given that the aspect ratio L/n controls both the fluctuations in the ensemble *and* representation learning, the optimal value of the ratio is likely nonzero but also small. In particular, representation learning is enhanced by the depth, but networks with too large a value of L/n will have an effect on the mean prediction of the ensemble, but perhaps even more importantly will lead to exponentially-large problems when working with only a *particular* network: for large enough L/n, our principle of typicality can break down, and so the generalization error can begin to exhibit exponential behavior.[27]

This further explains the success of our effective theory description at $O(L/n)$: a description with vanishing L/n, i.e., the infinite-width limit, is too simple to model the properties of deep neural networks in practice; a description for larger values of L/n, i.e., a *small*-width or *overly*-deep regime that includes many higher-order corrections, describes networks that are unlikely to be trainable; a description with small but nonzero L/n, i.e., the leading finite-width effective theory accurate to $O(L/n)$, is as simple as it can be and still accurately describe the networks that work well in practice.[28]

In summary, we've seen that the leading finite-width corrections all scale exactly according to our long-standing expectations, and we've discussed at a high level the potential tradeoff of depth versus width. In principle, we could go further in our analysis, evaluating the multi-input recursions for nearby inputs, evaluating all the terms in the covariance, and finding all the $O(L/n)$ contributions to the generalization error with specific coefficients.[29] If we did this, what could it tell us? In particular, can we theoretically optimize the aspect ratio L/n for a particular activation function without any experimentation?

Unfortunately at the order we're working, we won't be able to optimize the aspect ratio L/n using the prediction formula (∞.154) from the end of training: the linear

[27] In our discussion of *fluctuations* in §5.4, we explained that too large a value of L/n may lead to a difficult fine-tuning problem in terms of getting a particular network to behave critically.

On the one hand, the contribution of such fluctuations to the bias part of the generalization error is one way to see how the downstream effect of such fluctuations may lead to problems after training. On the other hand, the fact that fluctuations can ruin criticality for a particular network is another problem. The former problem affects the ensemble as a whole, while the latter problem affects individual networks.

[28] To better understand the scale that separates the trainable regime from the overly-deep regime, see Appendix A. For a discussion of how to extend this trainable regime to greater depths for fixed width, see our discussion of residual networks in Appendix B.

[29] You can also more easily evaluate the multi-input version of the relevant recursions numerically with this scaling in mind in order to determine the overall coefficient for a particular activation function of interest. This would be one way to analyze these solutions for the `ReLU`-like `GELU` and `SWISH` networks.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

dependence of the generalization on the ratio means that its derivative is independent of the ratio; instead, we'd need to compute the higher-order corrections of order $O(L^2/n^2)$ to optimize the ratio – which is hard, though at least straightforward to do in the effective theory framework we've developed.[30] At order $O(L/n)$, the best we could hope to see is whether nonzero L/n improves generalization or not by looking at the sign of its overall coefficient. Rather than going through that hassle, we'll instead get a little more mileage out of the mean prediction $m_{i;\dot\beta}$ by trying to understand how the *algorithm dependence* at finite width leads to additional tradeoffs within the bias term of the generalization error.

Inductive Bias of the Training Algorithm

As multiply mentioned, one of the key differences between the infinite-width and finite-width networks optimized via gradient-based learning is the *algorithm dependence* of the fully-trained solution for finite-width networks. In particular, the solution – at least for MSE losses – takes a universal form, (∞.154), with all of the dependence of the solution on the training algorithm encoded in the *algorithm projectors*, Z_{A}, Z_{B}, Z_{IA}, Z_{IB}, Z_{IIA}, and Z_{IIB}, whose functional forms we've established explicitly for the two second-order updates in (∞.148), for many many steps of gradient descent in (∞.142)–(∞.147), and for the ODE limit of gradient descent in (∞.149)–(∞.151). Through this universal projection, we now have a theoretical means of isolating the *inductive bias of the training algorithm* from all the other inductive biases that are present in a fully-trained neural network. This provides a natural way of theoretically evaluating the relative merits of different training algorithms.

One aspect that is apparent just from staring at the stochastic prediction (∞.154) is that the algorithm projectors only induce projections on the NTK-differential part of the finite-width prediction and cannot affect the NTK-fluctuation contribution. More concretely, let's continue our discussion of the bias term in the generalization error. In particular, the bias is given by the following difference:

$$m_{i;\dot\beta} - y_{i;\dot\beta}. \tag{∞.172}$$

On the one hand, it's clear that the NTK-variance contribution to the mean prediction, (∞.162), encoded in $A^{(L)}$ and $B^{(L)}$ is irreducible, entirely independent of the training algorithm; this irreducible NTK-variance contribution depends on the training data in a fixed way and arises due to instantiation-to-instantiation fluctuations in the NTK across different realizations of the parameters. On the other hand, the projectors always act on

[30] In §A.3, we'll compute such higher-order effects from an *information-theoretic* perspective. This analysis will give a heuristic prescription for optimizing the network's depth-to-width ratio r: we consider an auxiliary *unsupervised learning* objective thought to be beneficial for building representations and optimize the aspect ratio in terms of *that* criterion rather than the generalization error. In this setting, we're able to determine optimal aspect ratios for different activation functions by trading off leading-order effects against higher-order effects. This is somewhat similar in spirit to the way in which Newton's method trades off the leading decrease of the loss against the subleading loss increase in order to optimize the overall learning rate, cf. our discussion of Newton's method in footnote 8 of §10.2.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

the NTK-differential tensors $P^{(L)}$, $Q^{(L)}$, $R^{(L)}$, $S^{(L)}$, $T^{(L)}$, and $U^{(L)}$; this means that the dNTK's and ddNTKs' effect on the network's predictions is adjustable by the training algorithm and can be tuned differently for different datasets and tasks.

To reiterate our previous discussion of the tradeoff, on the one hand the NTK-fluctuation contribution is likely harmful as it leads to a breakdown of criticality for any *particular* network. On the other hand, the latter NTK-differential contribution is the ultimate source of the nontrivial representation learning at finite width, and so it would be nice to make this contribution large. With these algorithm projectors, we now have direct means to enhance the latter benefit while keeping fixed the former cost.

One way of understanding these algorithm projectors is as the sample-space *dual description* of a learning algorithm, analogous to the relationship between the feature functions $\phi_j(x)$ and the kernel $k(x_{\delta_1}, x_{\delta_2})$ that we explored in §10.4 and §11.4. From the parameter-space microscopic perspective, a learning algorithm, such as gradient descent, explicitly operates on the parameters. At the end of training, all the details of the algorithm are left implicit in the trained parameters $\theta^\star$, making it difficult to understand its effect on the model's predictions. From this perspective, the algorithm is simple to define (§7.2) but decidedly difficult to analyze for general interacting machine-learning models (§∞.2.2). However, if you can analyze or *solve* a model to find its sample-space dual description, then the algorithm projectors make the influence of the learning algorithm explicit.[31]

Analogously to the (nearly-)kernel perspective on (nearly-)linear models, the appearance of the algorithm projectors in the generalization error means that we can begin to think about *engineering* them directly, either by picking them to be directly used with a prediction formula such as (∞.154) or alternatively by attempting to find the parameter-space dual of an algorithm projector. While we expect this latter task to be difficult – just as it's often hard to find the feature functions that correspond to a particular kernel – it could be very worthwhile to explore as a means of engineering the inductive bias of the training algorithm directly.[32]

In particular, we expect that this could significantly improve generalization by designing algorithms that are better tailored to the details of an architecture or the properties of the underlying dataset or task.[33] This design process could in principle proceed as follows: *(i)* engineer a desired functional form for the algorithm projectors and then *(ii)* determine the parameter-space training algorithm that leads to the projectors having those desired properties or specific functional form. While further details of such **inverse algorithm design** is outside the scope of this book, we think that this line of *dual* analysis has the potential to unlock a much deeper understanding of the relationships

[31] This is the sense in which we meant that the conditioning on the learning algorithm would be *simple* for the trained ensemble in (0.9).

[32] This can also be seen as another advantage of nonlinear or interacting machine learning models. For linear models, the solution is always independent of the details of the learning algorithm (§10.2.2).

[33] Since the algorithm projectors have sample indices, the *optimal* choice of the algorithm will in general depend on the details and structure of the training set in addition to the definition of the model.

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

between the inductive bias of the training algorithm, the inductive bias of the network architecture, and the ultimate success of the model.

There's No Place Where Gradient Descent = Exact Bayesian Inference

Finally, a curious reader might wonder whether the connection that we detailed in §10.2.4 between gradient descent optimization and exact Bayesian inference for the infinite-width limit persists for more realistic networks at finite width. In short, the answer is no.

In long, recall the training hyperparameter settings (10.46) and (10.48) that matched gradient-based learning with Bayesian inference at infinite width. In particular, the latter condition (10.48) required turning off any learning in the hidden layers:

$$\lambda_b^{(\ell)} = 0\,, \qquad \lambda_W^{(\ell)} = 0\,, \qquad \text{for} \quad \ell < L. \tag{∞.173}$$

Importantly, this makes the hidden-layer NTK vanish exactly: $\widehat{H}^{(\ell)}_{i_1 i_2;\delta_1\delta_2} = 0$, for $\ell < L$ at any finite width, cf. (8.12). Next, flip back and look at the P- and Q-recursions for the dNTK–preactivation cross-correlation tensors, (11.49) and (11.52), and then you'll probably also want to flip further back and visit the F- and B-recursions, (8.79) and (8.89), respectively. (We'll be waiting here for you when you return.) Immediately, you should notice a problem: if the hidden-layer NTK mean vanishes, then at this order $B^{(\ell)} = 0$, $F^{(\ell)} = 0$, and all this together implies that $P^{(L)} = Q^{(L)} = 0$. Thus, all the effects of the dNTK are turned off. Similarly, if you flip forth and look at the R-, S-, T-, and U-recursions, (∞.177) and (∞.179)–(∞.181), then you'll notice that all the effects of the ddNTKs are turned off as well. Thus, the mean prediction of our finite-width gradient-based-learning ensemble (∞.160) *cannot* match the exact Bayesian posterior mean at finite width (6.88).

Note that this mismatch is obvious in hindsight; by only training the last layer (∞.173), we get a linear model (10.134). In other words, since the hyperparameter choices of (∞.173) lead to a model with random features that are fixed over the course of training, there's no representation learning possible for these settings. In contrast, we found nontrivial representation learning when studying exact Bayesian inference at finite width in §6.4.3.

Ultimately, for Bayesian inference we only care about the preactivation distribution $p\big(z^{(\ell)}|\mathcal{D}\big)$, while for gradient-based learning we need to consider the joint preactivation–NTK–dNTK–ddNTKs distribution $p\big(z^{(L)}, \widehat{H}^{(L)}, \widehat{\mathrm{d}H}^{(L)}, \widehat{\mathrm{dd_I}H}^{(L)}, \widehat{\mathrm{dd_{II}}H}^{(L)}|\mathcal{D}\big)$, which incorporates the statistics of the derivatives of the preactivations at initialization: such derivatives of the model output are invisible to the exact Bayesian inferencer.

∞.3 RG Flow of the ddNTKs: The Full Expressions

These expressions were kind of horrible, so we decided to hide them here at the end of the chapter. As such, they are only really needed for three reasons: *(i)* to explicitly check the absence of any NTK differentials for the hyperparameter setup (∞.173) and thus confirm

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

that the connection between gradient descent and Bayesian inference doesn't persist at finite width, *(ii)* to check the details of the depth-to-width scaling of the ddNTKs that we discussed in §∞.1, and *(iii)* to more generally evaluate the ddNTKs' contributions to the ensemble's mean prediction, (∞.164) and (∞.165), for multiple inputs – analytically or numerically – or to compute other higher-order statistics of our stochastic prediction (∞.154).

$\widehat{\mathrm{dd_I}H}$ Stochastic Forward Equation

$$
\begin{aligned}
&\widehat{\mathrm{dd_I}H}^{(\ell+1)}_{i_0 i_1 i_2 i_3;\delta_0\delta_1\delta_2\delta_3} \qquad (\infty.174)\\
&\quad= \delta_{i_0 i_1}\frac{\lambda_W^{(\ell+1)}}{n_\ell}\sum_{j_0,j_1,j_2,j_3=1}^{n_\ell}\delta_{j_0 j_1}W^{(\ell+1)}_{i_2 j_2}W^{(\ell+1)}_{i_3 j_3}\sigma^{(\ell)}_{j_1;\delta_1}\sigma'^{(\ell)}_{j_2;\delta_2}\sigma'^{(\ell)}_{j_3;\delta_3}\\
&\qquad\qquad\times\Big[\sigma''^{(\ell)}_{j_0;\delta_0}\widehat{H}^{(\ell)}_{j_0 j_2;\delta_0\delta_2}\widehat{H}^{(\ell)}_{j_0 j_3;\delta_0\delta_3}+\sigma'^{(\ell)}_{j_0;\delta_0}\widehat{\mathrm{d}H}^{(\ell)}_{j_0 j_2 j_3;\delta_0\delta_2\delta_3}\Big]\\
&\quad+\delta_{i_0 i_2}\frac{\lambda_W^{(\ell+1)}}{n_\ell}\sum_{j_0,j_1,j_2,j_3=1}^{n_\ell}\delta_{j_0 j_2}W^{(\ell+1)}_{i_3 j_3}W^{(\ell+1)}_{i_1 j_1}\sigma^{(\ell)}_{j_2;\delta_2}\sigma'^{(\ell)}_{j_3;\delta_3}\sigma'^{(\ell)}_{j_1;\delta_1}\\
&\qquad\qquad\times\Big[\sigma''^{(\ell)}_{j_0;\delta_0}\widehat{H}^{(\ell)}_{j_0 j_3;\delta_0\delta_3}\widehat{H}^{(\ell)}_{j_0 j_1;\delta_0\delta_1}+\sigma'^{(\ell)}_{j_0;\delta_0}\widehat{\mathrm{d}H}^{(\ell)}_{j_0 j_3 j_1;\delta_0\delta_3\delta_1}\Big]\\
&\quad+\delta_{i_0 i_3}\frac{\lambda_W^{(\ell+1)}}{n_\ell}\sum_{j_0,j_1,j_2,j_3=1}^{n_\ell}\delta_{j_0 j_3}W^{(\ell+1)}_{i_1 j_1}W^{(\ell+1)}_{i_2 j_2}\sigma^{(\ell)}_{j_3;\delta_3}\sigma'^{(\ell)}_{j_1;\delta_1}\sigma'^{(\ell)}_{j_2;\delta_2}\\
&\qquad\qquad\times\Big[\sigma''^{(\ell)}_{j_0;\delta_0}\widehat{H}^{(\ell)}_{j_0 j_1;\delta_0\delta_1}\widehat{H}^{(\ell)}_{j_0 j_2;\delta_0\delta_2}+\sigma'^{(\ell)}_{j_0;\delta_0}\widehat{\mathrm{d}H}^{(\ell)}_{j_0 j_1 j_2;\delta_0\delta_1\delta_2}\Big]\\
&\quad+\sum_{j_0,j_1,j_2,j_3=1}^{n_\ell}W^{(\ell+1)}_{i_0 j_0}W^{(\ell+1)}_{i_1 j_1}W^{(\ell+1)}_{i_2 j_2}W^{(\ell+1)}_{i_3 j_3}\sigma'^{(\ell)}_{j_1;\delta_1}\sigma'^{(\ell)}_{j_2;\delta_2}\sigma'^{(\ell)}_{j_3;\delta_3}\\
&\qquad\qquad\times\Big[\sigma'^{(\ell)}_{j_0;\delta_0}\widehat{\mathrm{dd_I}H}^{(\ell)}_{j_0 j_1 j_2 j_3;\delta_0\delta_1\delta_2\delta_3}+\sigma''^{(\ell)}_{j_0;\delta_0}\widehat{\mathrm{d}H}^{(\ell)}_{j_0 j_1 j_2;\delta_0\delta_1\delta_2}\widehat{H}^{(\ell)}_{j_0 j_3;\delta_0\delta_3}\\
&\qquad\qquad\quad+\sigma''^{(\ell)}_{j_0;\delta_0}\widehat{\mathrm{d}H}^{(\ell)}_{j_0 j_2 j_3;\delta_0\delta_2\delta_3}\widehat{H}^{(\ell)}_{j_0 j_1;\delta_0\delta_1}+\sigma''^{(\ell)}_{j_0;\delta_0}\widehat{\mathrm{d}H}^{(\ell)}_{j_0 j_3 j_1;\delta_0\delta_3\delta_1}\widehat{H}^{(\ell)}_{j_0 j_2;\delta_0\delta_2}\\
&\qquad\qquad\quad+\sigma'''^{(\ell)}_{j_0\delta_0}\widehat{H}^{(\ell)}_{j_0 j_1;\delta_0\delta_1}\widehat{H}^{(\ell)}_{j_0 j_2;\delta_0\delta_2}\widehat{H}^{(\ell)}_{j_0 j_3;\delta_0\delta_3}\Big].
\end{aligned}
$$

$\widehat{\mathrm{dd_{II}}H}$ Stochastic Forward Equation

$$
\begin{aligned}
&\widehat{\mathrm{dd_{II}}H}^{(\ell+1)}_{i_1 i_2 i_3 i_4;\delta_1\delta_2\delta_3\delta_4} \qquad (\infty.175)\\
&\quad=\delta_{i_1 i_3}\delta_{i_2 i_4}\left(\frac{\lambda_W^{(\ell+1)}}{n_\ell}\right)^2\sum_{j,k=1}^{n_\ell}\sigma'^{(\ell)}_{j;\delta_1}\sigma'^{(\ell)}_{k;\delta_2}\sigma^{(\ell)}_{j;\delta_3}\sigma^{(\ell)}_{k;\delta_4}\widehat{H}^{(\ell)}_{jk;\delta_1\delta_2}\\
&\quad+\delta_{i_1 i_2}\frac{\lambda_W^{(\ell+1)}}{n_\ell}\sum_{j_1,\ldots,j_4=1}^{n_\ell}\delta_{j_1 j_2}W^{(\ell+1)}_{i_3 j_3}W^{(\ell+1)}_{i_4 j_4}\sigma'^{(\ell)}_{j_1;\delta_1}\sigma'^{(\ell)}_{j_2;\delta_2}\sigma'^{(\ell)}_{j_3;\delta_3}\sigma'^{(\ell)}_{j_4;\delta_4}\widehat{H}^{(\ell)}_{j_1 j_3;\delta_1\delta_3}\widehat{H}^{(\ell)}_{j_2 j_4;\delta_2\delta_4}
\end{aligned}
$$

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

$$
\begin{aligned}
&+ \delta_{i_1 i_3} \frac{\lambda_W^{(\ell+1)}}{n_\ell} \sum_{j_1,\ldots,j_4=1}^{n_\ell} \delta_{j_1 j_3} W_{i_2 j_2}^{(\ell+1)} W_{i_4 j_4}^{(\ell+1)} \sigma_{j_3;\delta_3}^{(\ell)} \sigma'^{(\ell)}_{j_4;\delta_4} \sigma'^{(\ell)}_{j_1;\delta_1} \\
&\qquad \times \left[\sigma''^{(\ell)}_{j_2;\delta_2} \widehat{H}^{(\ell)}_{j_2 j_1;\delta_2\delta_1} \widehat{H}^{(\ell)}_{j_2 j_4;\delta_2\delta_4} + \sigma'^{(\ell)}_{j_2;\delta_2} \widehat{\mathrm{d}H}^{(\ell)}_{j_2 j_1 j_4;\delta_2\delta_1\delta_4} \right] \\
&+ \delta_{i_2 i_4} \frac{\lambda_W^{(\ell+1)}}{n_\ell} \sum_{j_1,\ldots,j_4=1}^{n_\ell} \delta_{j_2 j_4} W_{i_1 j_1}^{(\ell+1)} W_{i_3 j_3}^{(\ell+1)} \sigma_{j_4;\delta_4}^{(\ell)} \sigma'^{(\ell)}_{j_3;\delta_3} \sigma'^{(\ell)}_{j_2;\delta_2} \\
&\qquad \times \left[\sigma''^{(\ell)}_{j_1;\delta_1} \widehat{H}^{(\ell)}_{j_1 j_2;\delta_1\delta_2} \widehat{H}^{(\ell)}_{j_1 j_3;\delta_1\delta_3} + \sigma'^{(\ell)}_{j_1;\delta_1} \widehat{\mathrm{d}H}^{(\ell)}_{j_1 j_2 j_3;\delta_1\delta_2\delta_3} \right] \\
&+ \sum_{j_1,j_2,j_3,j_4=1}^{n_\ell} W_{i_1 j_1}^{(\ell+1)} W_{i_2 j_2}^{(\ell+1)} W_{i_3 j_3}^{(\ell+1)} W_{i_4 j_4}^{(\ell+1)} \sigma'^{(\ell)}_{j_3;\delta_3} \sigma'^{(\ell)}_{j_4;\delta_4} \\
&\qquad \times \Big[\sigma'^{(\ell)}_{j_1;\delta_1} \sigma'^{(\ell)}_{j_2;\delta_2} \widehat{\mathrm{dd}_{\mathrm{II}} H}^{(\ell)}_{j_1 j_2 j_3 j_4;\delta_1\delta_2\delta_3\delta_4} + \sigma''^{(\ell)}_{j_1;\delta_1} \sigma''^{(\ell)}_{j_2;\delta_2} \widehat{H}^{(\ell)}_{j_1 j_2;\delta_1\delta_2} \widehat{H}^{(\ell)}_{j_1 j_3;\delta_1\delta_3} \widehat{H}^{(\ell)}_{j_2 j_4;\delta_2\delta_4} \\
&\qquad\quad + \sigma'^{(\ell)}_{j_1;\delta_1} \sigma''^{(\ell)}_{j_2;\delta_2} \widehat{H}^{(\ell)}_{j_2 j_4;\delta_2\delta_4} \widehat{\mathrm{d}H}^{(\ell)}_{j_1 j_2 j_3;\delta_1\delta_2\delta_3} + \sigma'^{(\ell)}_{j_2;\delta_2} \sigma''^{(\ell)}_{j_1;\delta_1} \widehat{H}^{(\ell)}_{j_1 j_3;\delta_1\delta_3} \widehat{\mathrm{d}H}^{(\ell)}_{j_2 j_1 j_4;\delta_2\delta_1\delta_4} \Big] .
\end{aligned}
$$

$\widehat{\mathrm{dd}_{\mathrm{I}} H}$ Recursion

The mean of the first ddNTK decomposes as

$$
\begin{aligned}
&\mathbb{E}\left[\widehat{\mathrm{dd}_{\mathrm{I}} H}^{(\ell)}_{i_0 i_1 i_2 i_3;\delta_0\delta_1\delta_2\delta_3} \right] \\
&= \frac{1}{n_{\ell-1}} \left[\delta_{i_0 i_1}\delta_{i_2 i_3} R^{(\ell)}_{\delta_0\delta_1\delta_2\delta_3} + \delta_{i_0 i_2}\delta_{i_3 i_1} R^{(\ell)}_{\delta_0\delta_2\delta_3\delta_1} + \delta_{i_0 i_3}\delta_{i_1 i_2} R^{(\ell)}_{\delta_0\delta_3\delta_1\delta_2} \right] .
\end{aligned}
\tag{∞.176}
$$

The tensor $R^{(\ell)}$ satisfies the following layer-to-layer recursion:

$$
\begin{aligned}
&R^{(\ell+1)}_{\delta_0\delta_1\delta_2\delta_3} \\
&= \lambda_W^{(\ell+1)} C_W^{(\ell+1)} \left\langle \sigma''_{\delta_0} \sigma_{\delta_1} \sigma'_{\delta_2} \sigma'_{\delta_3} \right\rangle_{G^{(\ell)}} H^{(\ell)}_{\delta_0\delta_2} H^{(\ell)}_{\delta_0\delta_3} \\
&\quad + \left(\frac{n_\ell}{n_{\ell-1}} \right) \left(C_W^{(\ell+1)} \left\langle \sigma'_{\delta_2} \sigma'_{\delta_3} \right\rangle_{G^{(\ell)}} \right) \left(\lambda_W^{(\ell+1)} \left\langle \sigma''_{\delta_0} \sigma_{\delta_1} \right\rangle_{G^{(\ell)}} \right) B^{(\ell)}_{\delta_0\delta_0\delta_2\delta_3} \\
&\quad + \left(\frac{n_\ell}{n_{\ell-1}} \right) \left(C_W^{(\ell+1)} \left\langle \sigma'_{\delta_2} \sigma'_{\delta_3} \right\rangle_{G^{(\ell)}} \right) \left[\lambda_W^{(\ell+1)} \left(\left\langle \sigma''_{\delta_0} \sigma_{\delta_1} \right\rangle_{G^{(\ell)}} P^{(\ell)}_{\delta_0\delta_2\delta_3\delta_0} + \left\langle \sigma'_{\delta_0} \sigma'_{\delta_1} \right\rangle_{G^{(\ell)}} P^{(\ell)}_{\delta_0\delta_2\delta_3\delta_1} \right) \right] \\
&\quad + \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} H^{(\ell)}_{\delta_0\delta_3} \\
&\quad + \left(\frac{n_\ell}{n_{\ell-1}} \right) \left(C_W^{(\ell+1)} \left\langle \sigma'_{\delta_2} \sigma'_{\delta_3} \right\rangle_{G^{(\ell)}} \right) \left(C_W^{(\ell+1)} \left\langle \sigma'''_{\delta_0} \sigma'_{\delta_1} \right\rangle_{G^{(\ell)}} \right) B^{(\ell)}_{\delta_0\delta_0\delta_2\delta_3} H^{(\ell)}_{\delta_0\delta_1} \\
&\quad + \left(\frac{n_\ell}{n_{\ell-1}} \right) \left(C_W^{(\ell+1)} \left\langle \sigma'_{\delta_2} \sigma'_{\delta_3} \right\rangle_{G^{(\ell)}} \right) \\
&\qquad\qquad \left[C_W^{(\ell+1)} \left(\left\langle \sigma'''_{\delta_0} \sigma'_{\delta_1} \right\rangle_{G^{(\ell)}} P^{(\ell)}_{\delta_0\delta_2\delta_3\delta_0} + \left\langle \sigma''_{\delta_0} \sigma''_{\delta_1} \right\rangle_{G^{(\ell)}} P^{(\ell)}_{\delta_0\delta_2\delta_3\delta_1} \right) \right] H^{(\ell)}_{\delta_0\delta_1} \\
&\quad + \left(\frac{n_\ell}{n_{\ell-1}} \right) \left(C_W^{(\ell+1)} \left\langle \sigma'_{\delta_2} \sigma'_{\delta_3} \right\rangle_{G^{(\ell)}} \right) \left(C_W^{(\ell+1)} \left\langle \sigma'_{\delta_0} \sigma'_{\delta_1} \right\rangle_{G^{(\ell)}} \right) R^{(\ell)}_{\delta_0\delta_1\delta_2\delta_3} + O\left(\frac{1}{n} \right) .
\end{aligned}
\tag{∞.177}
$$

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

$\widehat{\mathbf{dd}_{\mathbf{II}}H}$ Recursions

The mean of the second ddNTK decomposes as

$$\mathbb{E}\left[\widehat{\mathrm{dd}_{\mathrm{II}}H}^{(\ell)}_{i_1i_2i_3i_4;\delta_1\delta_2\delta_3\delta_4}\right] = \frac{1}{n_{\ell-1}}\left[\delta_{i_1i_2}\delta_{i_3i_4}S^{(\ell)}_{\delta_1\delta_2\delta_3\delta_4} + \delta_{i_1i_3}\delta_{i_4i_2}T^{(\ell)}_{\delta_1\delta_3\delta_4\delta_2} + \delta_{i_1i_4}\delta_{i_2i_3}U^{(\ell)}_{\delta_1\delta_4\delta_2\delta_3}\right]. \tag{∞.178}$$

The tensor $S^{(\ell)}$ satisfies the following layer-to-layer recursion:

$$\begin{aligned}
&S^{(\ell+1)}_{\delta_1\delta_2\delta_3\delta_4} \\
&\quad= C_W^{(\ell+1)}\lambda_W^{(\ell+1)}\left\langle\sigma'_{\delta_1}\sigma'_{\delta_2}\sigma'_{\delta_3}\sigma'_{\delta_4}\right\rangle_{G^{(\ell)}} H^{(\ell)}_{\delta_1\delta_3}H^{(\ell)}_{\delta_2\delta_4} \\
&\qquad+\left(\frac{n_\ell}{n_{\ell-1}}\right)\left(C_W^{(\ell+1)}\left\langle\sigma'_{\delta_3}\sigma'_{\delta_4}\right\rangle_{G^{(\ell)}}\right) \\
&\qquad\qquad\left[\lambda_W^{(\ell+1)}\left\langle\sigma'_{\delta_1}\sigma'_{\delta_2}\right\rangle_{G^{(\ell)}} + C_W^{(\ell+1)}H^{(\ell)}_{\delta_1\delta_2}\left\langle\sigma''_{\delta_1}\sigma''_{\delta_2}\right\rangle_{G^{(\ell)}}\right]B^{(\ell)}_{\delta_1\delta_2\delta_3\delta_4} \\
&\qquad+\left(C_W^{(\ell+1)}\right)^2\left\langle\sigma''_{\delta_1}\sigma''_{\delta_2}\sigma'_{\delta_3}\sigma'_{\delta_4}\right\rangle_{G^{(\ell)}} H^{(\ell)}_{\delta_1\delta_2}H^{(\ell)}_{\delta_1\delta_3}H^{(\ell)}_{\delta_2\delta_4} \\
&\qquad+\left(\frac{n_\ell}{n_{\ell-1}}\right)\left(C_W^{(\ell+1)}\left\langle\sigma'_{\delta_1}\sigma'_{\delta_2}\right\rangle_{G^{(\ell)}}\right)\left(C_W^{(\ell+1)}\left\langle\sigma'_{\delta_3}\sigma'_{\delta_4}\right\rangle_{G^{(\ell)}}\right)S^{(\ell)}_{\delta_1\delta_2\delta_3\delta_4} + O\left(\frac{1}{n}\right).
\end{aligned} \tag{∞.179}$$

The tensor $T^{(\ell)}$ satisfies the following layer-to-layer recursion:

$$\begin{aligned}
&T^{(\ell+1)}_{\delta_1\delta_3\delta_4\delta_2} \\
&=\left(\lambda_W^{(\ell+1)}\right)^2\left\langle\sigma'_{\delta_1}\sigma'_{\delta_2}\sigma_{\delta_3}\sigma_{\delta_4}\right\rangle_{G^{(\ell)}} H^{(\ell)}_{\delta_1\delta_2} \\
&\quad+\left(\frac{n_\ell}{n_{\ell-1}}\right)\left(\lambda_W^{(\ell+1)}\right)^2\sum_{\delta_5,\ldots,\delta_8\in\mathcal{D}}\left\langle z_{\delta_5}\sigma'_{\delta_1}\sigma_{\delta_3}\right\rangle_{G^{(\ell)}}\left\langle z_{\delta_6}\sigma'_{\delta_2}\sigma_{\delta_4}\right\rangle_{G^{(\ell)}} G_{(\ell)}^{\delta_5\delta_7}G_{(\ell)}^{\delta_6\delta_8}F^{(\ell)}_{\delta_7\delta_1\delta_8\delta_2} \\
&\quad+C_W^{(\ell+1)}\lambda_W^{(\ell+1)}\left\langle\sigma'_{\delta_1}\sigma''_{\delta_2}\sigma_{\delta_3}\sigma'_{\delta_4}\right\rangle_{G^{(\ell)}} H^{(\ell)}_{\delta_2\delta_1}H^{(\ell)}_{\delta_2\delta_4} \\
&\quad+\left(\frac{n_\ell}{n_{\ell-1}}\right)C_W^{(\ell+1)}\lambda_W^{(\ell+1)}H^{(\ell)}_{\delta_2\delta_4}\sum_{\delta_5,\ldots,\delta_8\in\mathcal{D}}\left\langle z_{\delta_5}\sigma'_{\delta_1}\sigma_{\delta_3}\right\rangle_{G^{(\ell)}}\left\langle z_{\delta_6}\sigma''_{\delta_2}\sigma'_{\delta_4}\right\rangle_{G^{(\ell)}} G_{(\ell)}^{\delta_5\delta_7}G_{(\ell)}^{\delta_6\delta_8}F^{(\ell)}_{\delta_7\delta_1\delta_8\delta_2} \\
&\quad+\left(\frac{n_\ell}{n_{\ell-1}}\right)C_W^{(\ell+1)}\lambda_W^{(\ell+1)}\left\langle\sigma'_{\delta_2}\sigma'_{\delta_4}\right\rangle_{G^{(\ell)}}\left(\left\langle\sigma''_{\delta_1}\sigma_{\delta_3}\right\rangle_{G^{(\ell)}}Q^{(\ell)}_{\delta_2\delta_4\delta_1\delta_1} + \left\langle\sigma'_{\delta_1}\sigma'_{\delta_3}\right\rangle_{G^{(\ell)}}Q^{(\ell)}_{\delta_2\delta_4\delta_1\delta_3}\right) \\
&\quad+C_W^{(\ell+1)}\lambda_W^{(\ell+1)}\left\langle\sigma'_{\delta_2}\sigma''_{\delta_1}\sigma_{\delta_4}\sigma'_{\delta_3}\right\rangle_{G^{(\ell)}} H^{(\ell)}_{\delta_1\delta_2}H^{(\ell)}_{\delta_1\delta_3} \\
&\quad+\left(\frac{n_\ell}{n_{\ell-1}}\right)C_W^{(\ell+1)}\lambda_W^{(\ell+1)}H^{(\ell)}_{\delta_1\delta_3}\sum_{\delta_5,\ldots,\delta_8\in\mathcal{D}}\left\langle z_{\delta_5}\sigma'_{\delta_2}\sigma_{\delta_4}\right\rangle_{G^{(\ell)}}\left\langle z_{\delta_6}\sigma''_{\delta_1}\sigma'_{\delta_3}\right\rangle_{G^{(\ell)}} G_{(\ell)}^{\delta_5\delta_7}G_{(\ell)}^{\delta_6\delta_8}F^{(\ell)}_{\delta_7\delta_2\delta_8\delta_1} \\
&\quad+\left(\frac{n_\ell}{n_{\ell-1}}\right)C_W^{(\ell+1)}\lambda_W^{(\ell+1)}\left\langle\sigma'_{\delta_1}\sigma'_{\delta_3}\right\rangle_{G^{(\ell)}}\left(\left\langle\sigma''_{\delta_2}\sigma_{\delta_4}\right\rangle_{G^{(\ell)}}Q^{(\ell)}_{\delta_1\delta_3\delta_2\delta_2} + \left\langle\sigma'_{\delta_2}\sigma'_{\delta_4}\right\rangle_{G^{(\ell)}}Q^{(\ell)}_{\delta_1\delta_3\delta_2\delta_4}\right)
\end{aligned} \tag{∞.180}$$

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press

$$
\begin{aligned}
&+\left(\frac{n_\ell}{n_{\ell-1}}\right)\left(C_W^{(\ell+1)}\right)^2\left\langle\sigma'_{\delta_1}\sigma'_{\delta_3}\right\rangle_{G^{(\ell)}}\left\langle\sigma'_{\delta_2}\sigma'_{\delta_4}\right\rangle_{G^{(\ell)}}T^{(\ell)}_{\delta_1\delta_3\delta_4\delta_2}\\
&+\left(C_W^{(\ell+1)}\right)^2\left\langle\sigma''_{\delta_1}\sigma''_{\delta_2}\sigma'_{\delta_3}\sigma'_{\delta_4}\right\rangle_{G^{(\ell)}}H^{(\ell)}_{\delta_1\delta_2}H^{(\ell)}_{\delta_1\delta_3}H^{(\ell)}_{\delta_2\delta_4}\\
&+\left(\frac{n_\ell}{n_{\ell-1}}\right)\left(C_W^{(\ell+1)}\right)^2H^{(\ell)}_{\delta_1\delta_3}H^{(\ell)}_{\delta_2\delta_4}\\
&\qquad\qquad\sum_{\delta_5,\ldots,\delta_8\in\mathcal{D}}\left\langle z_{\delta_5}\sigma''_{\delta_1}\sigma'_{\delta_3}\right\rangle_{G^{(\ell)}}\left\langle z_{\delta_6}\sigma''_{\delta_2}\sigma'_{\delta_4}\right\rangle_{G^{(\ell)}}G^{\delta_5\delta_7}_{(\ell)}G^{\delta_6\delta_8}_{(\ell)}F^{(\ell)}_{\delta_7\delta_1\delta_8\delta_2}\\
&+\left(\frac{n_\ell}{n_{\ell-1}}\right)\left(C_W^{(\ell+1)}\right)^2H^{(\ell)}_{\delta_2\delta_4}\left\langle\sigma'_{\delta_1}\sigma'_{\delta_3}\right\rangle_{G^{(\ell)}}\left(\left\langle\sigma'''_{\delta_2}\sigma'_{\delta_4}\right\rangle_{G^{(\ell)}}Q^{(\ell)}_{\delta_1\delta_3\delta_2\delta_2}+\left\langle\sigma''_{\delta_2}\sigma''_{\delta_4}\right\rangle_{G^{(\ell)}}Q^{(\ell)}_{\delta_1\delta_3\delta_2\delta_4}\right)\\
&+\left(\frac{n_\ell}{n_{\ell-1}}\right)\left(C_W^{(\ell+1)}\right)^2H^{(\ell)}_{\delta_1\delta_3}\left\langle\sigma'_{\delta_2}\sigma'_{\delta_4}\right\rangle_{G^{(\ell)}}\left(\left\langle\sigma'''_{\delta_1}\sigma'_{\delta_3}\right\rangle_{G^{(\ell)}}Q^{(\ell)}_{\delta_2\delta_4\delta_1\delta_1}+\left\langle\sigma''_{\delta_1}\sigma''_{\delta_3}\right\rangle_{G^{(\ell)}}Q^{(\ell)}_{\delta_2\delta_4\delta_1\delta_3}\right)\\
&+O\left(\frac{1}{n}\right).
\end{aligned}
$$

The tensor $U^{(\ell)}$ satisfies the following layer-to-layer recursion:

$$
\begin{aligned}
U^{(\ell+1)}_{\delta_1\delta_4\delta_2\delta_3}&=\left(C_W^{(\ell+1)}\right)^2\left\langle\sigma''_{\delta_1}\sigma''_{\delta_2}\sigma'_{\delta_3}\sigma'_{\delta_4}\right\rangle_{G^{(\ell)}}H^{(\ell)}_{\delta_1\delta_2}H^{(\ell)}_{\delta_1\delta_3}H^{(\ell)}_{\delta_2\delta_4} \qquad (\infty.181)\\
&\quad+\left(\frac{n_\ell}{n_{\ell-1}}\right)\left(C_W^{(\ell+1)}\left\langle\sigma'_{\delta_1}\sigma'_{\delta_4}\right\rangle_{G^{(\ell)}}\right)\left(C_W^{(\ell+1)}\left\langle\sigma'_{\delta_2}\sigma'_{\delta_3}\right\rangle_{G^{(\ell)}}\right)U^{(\ell)}_{\delta_1\delta_4\delta_2\delta_3}+O\left(\frac{1}{n}\right).
\end{aligned}
$$

https://doi.org/10.1017/9781009023405.014 Published online by Cambridge University Press