9

Effective Theory of the NTK at Initialization

> *In short, we believe that we have answered Minsky and Papert's challenge and* have *found a learning result sufficiently powerful to demonstrate that their pessimism about learning in multilayer machines was misplaced.*
>
> Rumelhart, Hinton, and Williams [15],
> acausally rising to meet the criticism from the epigraph of §5.

Since the last chapter was a tempest of equations, algebra, and integration, let's take some moments to value our expectations.

Our goal in §8 was to determine the NTK–preactivation joint distribution for a given layer ℓ at initialization: $p\big(z^{(\ell)}, \widehat{H}^{(\ell)}\big|\mathcal{D}\big)$. The data-dependent couplings and the connected correlators of this distribution *run* with depth according to the recursions that we just laboriously derived – (8.63) for the NTK mean, (8.77) and (8.79) for the NTK–preactivation cross correlations, and (8.89) and (8.97) for the NTK variance – in addition to the recursions derived in §4 for the kernel (4.118) and for the four-point vertex (4.119). This *RG-flow* analysis taught us that the NTK is a deterministic object in the first layer (§8.1), stochastically fluctuates and cross-correlates in the second layer (§8.2), and then further accumulates fluctuations and cross correlations in deeper layers (§8.3).

Now that we've thoroughly discussed the math, in this chapter we'll finally be able to consider the physics of this joint distribution. Building on our discussion of *criticality* and *universality* in §5, we'll first lay the groundwork for a similar analysis of the NTK while highlighting the relevant results from the last chapter (§9.1). In particular, our focus will be on understanding how the initialization hyperparameters and the training hyperparameters affect gradient descent at finite width. We'll once again find that the depth-to-width ratio L/n plays a starring role in controlling finite-width effects, first for the scale-invariant universality class (§9.2) and then for the $K^{\star} = 0$ universality class (§9.3). For both cases, the growing importance of NTK fluctuations and cross correlations with depth makes the finite-width interaction *relevant* under RG flow of the NTK.

https://doi.org/10.1017/9781009023405.011 Published online by Cambridge University Press

Finally, we'll introduce the infamous *exploding and vanishing gradient problem* of deep learning and see how our notion of criticality completely mitigates this problem (§9.4). In this context, we also explain how the bias and weight learning rates should each be scaled with the network depth.

9.1 Criticality Analysis of the NTK

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

$$C_b^{(\ell)} = C_b \,, \qquad C_W^{(\ell)} = C_W \,. \tag{9.1}$$

Further paralleling §5.4, we will consider MLPs with uniform hidden layer widths

$$n_1 = \cdots = n_{L-1} \equiv n, \tag{9.2}$$

which is a sensible choice in practice as well as notationally simplifying.

For the training hyperparameters, however, we'll preserve the layer dependence of the bias learning rate $\lambda_b^{(\ell)}$ and weight learning rate $\lambda_W^{(\ell)}$ for now, as different universality classes will require different treatments. We'll explore the general principle behind these hyperparameter choices in §9.4.

Going forward, we'll only focus on the leading contributions from the $1/n$ expansion to the single-input statistics, neglecting the subleading corrections at next-to-leading order and reserving the multi-input analysis for your private amusement.

Leading-Order NTK Recursions for a Single Input

Let's start with the NTK mean recursion. Analogously to all other observables, the $1/n$ expansion induces a series expansion on the NTK mean of the form

$$H^{(\ell)}_{\alpha_1\alpha_2} = H^{\{0\}(\ell)}_{\alpha_1\alpha_2} + \frac{1}{n_{\ell-1}} H^{\{1\}(\ell)}_{\alpha_1\alpha_2} + \frac{1}{n_{\ell-1}^2} H^{\{2\}(\ell)}_{\alpha_1\alpha_2} + O\left(\frac{1}{n^3}\right) . \tag{9.3}$$

Just as we defined the kernel $K^{(\ell)}_{\alpha_1\alpha_2}$ as the infinite-width limit of the mean metric $G^{(\ell)}_{\alpha_1\alpha_2}$ (4.106), let us give the leading $O(1)$ piece of the NTK mean a special symbol,

$$\Theta^{(\ell)}_{\alpha_1\alpha_2} \equiv H^{\{0\}(\ell)}_{\alpha_1\alpha_2} , \tag{9.4}$$

and a special name: the **frozen NTK**. The frozen NTK controls the training dynamics in the infinite-width limit, which we will investigate in detail next chapter.[1]

[1] Typically in the literature, the *neural tangent kernel* or NTK refers to this deterministic infinite-width NTK mean $\Theta^{(\ell)}_{\alpha_1\alpha_2}$. Since we are principally concerned with understanding finite-width networks, we instead chose to define and refer to the stochastic object $\widehat{H}^{(\ell)}_{i_1 i_2;\alpha_1\alpha_2}$ as the NTK. As a concession to the literature, we've used the customary symbol for the NTK, Θ, to represent the frozen NTK. (As a helpful mnemonic, note that there is an H frozen inside the Θ.) Unfortunately, you'll have to wait until §11 to understand the reason why we call the infinite-width NTK *frozen*; there we'll see how finite-width effects *defrost* the training process and make the NTK move. Here, in this chapter, you can at least see how it gets *agitated* by finite-width fluctuations.

https://doi.org/10.1017/9781009023405.011 Published online by Cambridge University Press

Now, taking the leading piece of the NTK mean recursion (8.63), we get a recursion solely for the frozen NTK:

$$\Theta^{(\ell+1)}_{\alpha_1\alpha_2} = \lambda_b^{(\ell+1)} + \lambda_W^{(\ell+1)} \langle \sigma_{\alpha_1}\sigma_{\alpha_2}\rangle_{K^{(\ell)}} + C_W \langle \sigma'_{\alpha_1}\sigma'_{\alpha_2}\rangle_{K^{(\ell)}} \Theta^{(\ell)}_{\alpha_1\alpha_2} . \tag{9.5}$$

Concurrently, as we are neglecting subleading contributions, we have exchanged the Gaussian expectations over the mean metric $G^{(\ell)}$ for ones over the kernel $K^{(\ell)}$. Finally, specializing to a single input, we simply drop the sample indices to get this recursion's final form,

$$\Theta^{(\ell+1)} = \lambda_b^{(\ell+1)} + \lambda_W^{(\ell+1)} g\Big(K^{(\ell)}\Big) + \chi_\perp\Big(K^{(\ell)}\Big)\,\Theta^{(\ell)}, \tag{9.6}$$

with the initial condition coming directly from our first-layer NTK analysis (8.23),

$$\Theta^{(1)} = \lambda_b^{(1)} + \lambda_W^{(1)} \left(\frac{1}{n_0} \sum_{j=1}^{n_0} x_j^2 \right) . \tag{9.7}$$

Note that here we have also made use of a helper function and susceptibility from §5.

For your convenience, let us also recall and reprint the full set of helper functions – (5.5) and (5.52) – and susceptibilities – (5.50) and (5.51) – that we first made popular in §5:

$$g(K) = \langle \sigma(z)\,\sigma(z)\rangle_K \,, \tag{9.8}$$

$$h(K) \equiv \frac{C_W}{4K^2} \Big\langle \sigma'(z)\,\sigma'(z)\,\Big(z^2 - K\Big)\Big\rangle_K = \frac{1}{2}\frac{d}{dK}\chi_\perp(K)\,, \tag{9.9}$$

$$\chi_\parallel(K) = C_W g'(K) = \frac{C_W}{2K^2} \Big\langle \sigma(z)\,\sigma(z)\,\Big(z^2 - K\Big)\Big\rangle_K = \frac{C_W}{K} \langle z\,\sigma'(z)\,\sigma(z)\rangle_K \,, \tag{9.10}$$

$$\chi_\perp(K) = C_W \langle \sigma'(z)\,\sigma'(z)\rangle_K \,. \tag{9.11}$$

As a reminder, to go between the middle and right-hand expression in (9.10), you should integrate by parts.

For the remaining recursions, we're going to fast-forward the process as the procedure for converting the multi-input recursions to leading-order single-input recursions surely requires your attention but is somewhat mindless: *(i)* drop the layer dependence of the initialization hyperparameters as (9.1) and uniformize the layer widths as (9.2); *(ii)* drop sample indices everywhere; *(iii)* replace the mean metric $G^{(\ell)}$ and the NTK mean $H^{(\ell)}$ with the kernel $K^{(\ell)}$ and the frozen NTK $\Theta^{(\ell)}$, respectively;[2] and *(iv)* substitute in for

[2] Picking nits, we should really make $1/n$ expansions – similar to (4.106) for the mean metric $G^{(\ell)}$ and (9.3) for the NTK mean $H^{(\ell)}$ – for the finite-width tensors $A^{(\ell)}$, $B^{(\ell)}$, $D^{(\ell)}$, $F^{(\ell)}$, and also properly make use of the one that we made for $V^{(\ell)}$ (4.105), denoting the leading-order pieces as $A^{\{0\}(\ell)}$ and such, and dropping the subleading pieces. In the interest of notational sanity we won't impose this on you, though our recursions for these tensors should all be understood as referring to these leading-order pieces. (The kernel and the frozen NTK are special in that these infinite-width objects have already been well-studied by the community, and so in this case it's important to differentiate between the finite-width object and the infinite-width piece.)

https://doi.org/10.1017/9781009023405.011 Published online by Cambridge University Press

helper functions and susceptibilities (9.8)–(9.11). In particular, this last step has the benefit of letting us recycle our results from §5 on the deep asymptotic behavior of these functions.

It will also be necessary to recall the single-input leading-order expression for the auxiliary stochastic variable (8.74),

$$\widehat{\Omega}^{(\ell+1)} \equiv \lambda_W^{(\ell+1)} \sigma(z)\sigma(z) + C_W\, \Theta^{(\ell)}\, \sigma'(z)\sigma'(z), \tag{9.12}$$

which appears in the recursions for $D^{(\ell)}$ (8.77) and $A^{(\ell)}$ (8.97); we'll make this substitution the penultimate step *(iii-b)*, if you will. In making these substitutions, please keep in mind that the frozen NTK $\Theta^{(\ell)}$ multiplying the second term is not a random variable and hence can be escorted out of any Gaussian expectations.

At this point, you should grab another roll of parchment, jot down expressions (9.8)–(9.12), flip back a few pages to locate recursions (8.77), (8.79), (8.89), and (8.97), for $D^{(\ell)}$, $F^{(\ell)}$, $B^{(\ell)}$, and $A^{(\ell)}$, respectively (or perhaps you kiddos can simply click the equation references in your eBook and copy over the equations to your tablet), and simplify them according to the four-(though-sometimes-secretly-five-)step process *(i)*–*(iv)* above. When you're finished, make sure you agree with us:

$$\begin{aligned} D^{(\ell+1)} = {} & \chi_\perp^{(\ell)} \chi_\parallel^{(\ell)} D^{(\ell)} + \left(\frac{\lambda_W^{(\ell+1)}}{C_W}\right)\left[C_W^2 \left\langle \sigma(z)\sigma(z)\sigma(z)\sigma(z)\right\rangle_{K^{(\ell)}} - \left(C_W g^{(\ell)}\right)^2 + \left(\chi_\parallel^{(\ell)}\right)^2 V^{(\ell)}\right] \\ & + \Theta^{(\ell)} \left[C_W^2 \left\langle \sigma(z)\sigma(z)\sigma'(z)\sigma'(z)\right\rangle_{K^{(\ell)}} - C_W g^{(\ell)} \chi_\perp^{(\ell)} + 2h^{(\ell)} \chi_\parallel^{(\ell)}\, V^{(\ell)}\right], \end{aligned} \tag{9.13}$$

$$F^{(\ell+1)} = \left(\chi_\parallel^{(\ell)}\right)^2 F^{(\ell)} + C_W^2 \left\langle \sigma(z)\sigma(z)\sigma'(z)\sigma'(z)\right\rangle_{K^{(\ell)}} \Theta^{(\ell)}\,, \tag{9.14}$$

$$B^{(\ell+1)} = \left(\chi_\perp^{(\ell)}\right)^2 B^{(\ell)} + C_W^2 \left\langle \sigma'(z)\sigma'(z)\sigma'(z)\sigma'(z)\right\rangle_{K^{(\ell)}} \left(\Theta^{(\ell)}\right)^2\,, \tag{9.15}$$

$$\begin{aligned} A^{(\ell+1)} = {} & \left(\chi_\perp^{(\ell)}\right)^2 A^{(\ell)} + \left(\frac{\lambda_W^{(\ell+1)}}{C_W}\right)^2 \left[C_W^2 \left\langle \sigma(z)\sigma(z)\sigma(z)\sigma(z)\right\rangle_{K^{(\ell)}} - \left(C_W g^{(\ell)}\right)^2 + \left(\chi_\parallel^{(\ell)}\right)^2 V^{(\ell)}\right] \\ & + 2\left(\frac{\lambda_W^{(\ell+1)}}{C_W}\right) \Theta^{(\ell)} \left[C_W^2 \left\langle \sigma(z)\sigma(z)\sigma'(z)\sigma'(z)\right\rangle_{K^{(\ell)}} - C_W g^{(\ell)} \chi_\perp^{(\ell)} + 2h^{(\ell)} \chi_\parallel^{(\ell)}\, V^{(\ell)}\right] \\ & + 2\left(\frac{\lambda_W^{(\ell+1)}}{C_W}\right) \chi_\perp^{(\ell)} \chi_\parallel^{(\ell)} D^{(\ell)} + 4h^{(\ell)} \chi_\perp^{(\ell)} \Theta^{(\ell)} D^{(\ell)} \\ & + \left(\Theta^{(\ell)}\right)^2 \left[C_W^2 \left\langle \sigma'(z)\sigma'(z)\sigma'(z)\sigma'(z)\right\rangle_{K^{(\ell)}} - \left(\chi_\perp^{(\ell)}\right)^2 + \left(2h^{(\ell)}\right)^2 V^{(\ell)}\right]. \end{aligned} \tag{9.16}$$

For these recursions, the initial conditions (recalling that the first-layer NTK is fully deterministic) all vanish identically as

$$A^{(1)} = B^{(1)} = D^{(1)} = F^{(1)} = 0. \tag{9.17}$$

https://doi.org/10.1017/9781009023405.011 Published online by Cambridge University Press

Here also, for helper functions and susceptibilities, we used the following simplifying notation:

$$g^{(\ell)} \equiv g\!\left(K^{(\ell)}\right), \qquad h^{(\ell)} \equiv h\!\left(K^{(\ell)}\right), \qquad \chi_{\parallel}^{(\ell)} \equiv \chi_{\parallel}\!\left(K^{(\ell)}\right), \qquad \chi_{\perp}^{(\ell)} \equiv \chi_{\perp}\!\left(K^{(\ell)}\right), \tag{9.18}$$

making the kernel dependence implicit.

We can further simplify (9.13) and (9.16) by recalling the single-input recursion for the four-point vertex (5.109),

$$V^{(\ell+1)} = \left(\chi_{\parallel}^{(\ell)}\right)^2 V^{(\ell)} + C_W^2 \left[\langle\sigma(z)\sigma(z)\sigma(z)\sigma(z)\rangle_{K^{(\ell)}} - \left(g^{(\ell)}\right)^2\right]. \tag{9.19}$$

Keep staring at these equations, and you'll see slightly more compact expressions emerge:

$$\begin{aligned} D^{(\ell+1)} &= \chi_{\perp}^{(\ell)}\chi_{\parallel}^{(\ell)} D^{(\ell)} + \left(\frac{\lambda_W^{(\ell+1)}}{C_W}\right) V^{(\ell+1)} \\ &\quad + \Theta^{(\ell)}\left[C_W^2 \left\langle\sigma(z)\sigma(z)\sigma'(z)\sigma'(z)\right\rangle_{K^{(\ell)}} - C_W g^{(\ell)}\chi_{\perp}^{(\ell)} + 2h^{(\ell)}\chi_{\parallel}^{(\ell)} V^{(\ell)}\right], \end{aligned} \tag{9.20}$$

$$\begin{aligned} A^{(\ell+1)} &= \left(\chi_{\perp}^{(\ell)}\right)^2 A^{(\ell)} - \left(\frac{\lambda_W^{(\ell+1)}}{C_W}\right)^2 V^{(\ell+1)} + 2\left(\frac{\lambda_W^{(\ell+1)}}{C_W}\right) D^{(\ell+1)} + 4h^{(\ell)}\chi_{\perp}^{(\ell)}\Theta^{(\ell)} D^{(\ell)} \\ &\quad + \left(\Theta^{(\ell)}\right)^2\left[C_W^2 \left\langle\sigma'(z)\sigma'(z)\sigma'(z)\sigma'(z)\right\rangle_{K^{(\ell)}} - \left(\chi_{\perp}^{(\ell)}\right)^2 + \left(2h^{(\ell)}\right)^2 V^{(\ell)}\right], \end{aligned} \tag{9.21}$$

which you may find makes things simpler when solving these recursions. However, please use these formulae with caution as both ℓ-th-layer and $(\ell+1)$-th-layer objects appear on their right-hand sides.

The Relevance of Scaling Laws

For the rest of this chapter, we will work through solving the five leading-order single-input NTK recursions (9.6) and (9.13)–(9.16). (Remember that we already solved single-input recursions for the kernel and four-point vertex way back in §5.) In solving these recursions, we will find that each observable obeys our *scaling ansatz* (5.93):

$$\begin{aligned} \mathcal{O}^{(\ell)} &= \left(\frac{1}{\ell}\right)^{p_{\mathcal{O}}}\left[c_{0,0} + c_{1,1}\left(\frac{\log \ell}{\ell}\right) + c_{1,0}\left(\frac{1}{\ell}\right) + c_{2,2}\left(\frac{\log^2 \ell}{\ell^2}\right) + \cdots\right] \\ &= \left(\frac{1}{\ell}\right)^{p_{\mathcal{O}}}\left[\sum_{s=0}^{\infty}\sum_{q=0}^{s} c_{s,q}\left(\frac{\log^q \ell}{\ell^s}\right)\right]. \end{aligned} \tag{9.22}$$

Recall that $p_{\mathcal{O}}$ is a *critical exponent*, which is universal for a given universality class of activation functions, while the constants $c_{s,q}$ depend on some of the details of the particular activation function under consideration.

https://doi.org/10.1017/9781009023405.011 Published online by Cambridge University Press

To properly understand the physics of these observables, recall from §5.4 that we need to consider dimensionless quantities. For the two tensors controlling the NTK variance, we should normalize by the square of the frozen NTK,

$$\frac{A^{(\ell)}}{n\left(\Theta^{(\ell)}\right)^2} \sim \frac{1}{n}\left(\frac{1}{\ell}\right)^{p_A-2p_\Theta} + \cdots, \qquad \frac{B^{(\ell)}}{n\left(\Theta^{(\ell)}\right)^2} \sim \frac{1}{n}\left(\frac{1}{\ell}\right)^{p_B-2p_\Theta} + \cdots, \tag{9.23}$$

while for the NTK–preactivation cross correlation, we should instead normalize by one factor of the frozen NTK and one factor of the kernel,

$$\frac{D^{(\ell)}}{nK^{(\ell)}\Theta^{(\ell)}} \sim \frac{1}{n}\left(\frac{1}{\ell}\right)^{p_D-p_\Theta-p_0} + \cdots, \qquad \frac{F^{(\ell)}}{nK^{(\ell)}\Theta^{(\ell)}} \sim \frac{1}{n}\left(\frac{1}{\ell}\right)^{p_F-p_\Theta-p_0} + \cdots, \tag{9.24}$$

where p_0 was the critical exponent for the single-input kernel $K^{(\ell)}$.

By looking at these dimensionless quantities, we'll find *scaling laws* that transcend universality classes. As a particular example, recall that the normalized four-point vertex (5.128),

$$\frac{V^{(\ell)}}{n\left(K^{(\ell)}\right)^2} \sim \frac{1}{n}\left(\frac{1}{\ell}\right)^{p_V-2p_0} + \cdots, \tag{9.25}$$

gave rise to a scaling law (5.129)

$$p_V - 2p_0 = -1, \tag{9.26}$$

for both scale-invariant and $K^\star = 0$ activation functions. This scaling law let us interpret the ratio ℓ/n as an *emergent scale* controlling the leading finite-width behavior of the preactivation distribution. *Spoiler alert:* in much the same way, we'll find scaling laws

$$p_A - 2p_\Theta = -1\,, \quad p_B - 2p_\Theta = -1\,, \quad p_D - p_\Theta - p_0 = -1\,, \quad p_F - p_\Theta - p_0 = -1 \tag{9.27}$$

that also hold for both the scale-invariant and $K^\star = 0$ universality classes. Thus, we'll be able to conclude that all the leading finite-width effects of the NTK–preactivation joint distribution are *relevant* and controlled by the same ℓ/n perturbative cutoff. This means that we can effectively describe the training of realistic deep networks of finite width and nonzero L/n.

Formalities: Perpendicular Perturbations and the Frozen NTK

Before explicitly analyzing universality classes, let us note that the frozen-NTK recursion (9.6) admits a formal solution

$$\Theta^{(\ell)} = \sum_{\ell'=1}^{\ell}\left\{\left[\lambda_b^{(\ell')} + \lambda_W^{(\ell')} g^{(\ell'-1)}\right]\left[\prod_{\ell''=\ell'}^{\ell-1}\chi_\perp^{(\ell'')}\right]\right\}. \tag{9.28}$$

https://doi.org/10.1017/9781009023405.011 Published online by Cambridge University Press

In words, we see that the solution involves a sum over all the previous layers $1, \ldots, \ell$, and that each term in the sum involves an additive contribution $\lambda_b^{(\ell')} + \lambda_W^{(\ell')} g^{(\ell'-1)}$. Such a contribution then gets recursively multiplied by perpendicular susceptibilities up to the $(\ell - 1)$-th layer, resulting in an overall multiplicative factor $\prod_{\ell''=\ell'}^{\ell-1} \chi_\perp^{(\ell'')}$. To avoid the exponential behavior that's generic with such a factor, we must set $\chi_\perp = 1$.

It is enlightening to tie this insight to the discussion we had in §5.1 where we performed our general criticality analysis of the kernel recursion. There, we first looked at the single-input kernel and set $\chi_\parallel = 1$ to avoid exponential behavior in the network outputs. Then, we looked at the two-input kernel and analyzed how the off-diagonal perpendicular perturbations $\delta\delta K_{[2]}^{(\ell)}$ flow. Turning off the odd perturbations $\delta K_{[1]}^{(\ell)}$, a brief inspection of the perpendicular recursion (5.48),

$$\delta\delta K_{[2]}^{(\ell+1)} = \chi_\perp^{(\ell)} \delta\delta K_{[2]}^{(\ell)}, \tag{9.29}$$

necessitated the criticality condition $\chi_\perp = 1$ so as to preserve the difference between nearby inputs as they propagate through the network. At the time, we presumed that such a condition would be useful for comparing nearby inputs when learning from data. Indeed, the same multiplicative factor that appeared in the formal solution for the frozen NTK (9.28),

$$\prod_{\ell''=\ell'}^{\ell-1} \chi_\perp^{(\ell'')} = \frac{\delta\delta K_{[2]}^{(\ell)}}{\delta\delta K_{[2]}^{(\ell')}}, \tag{9.30}$$

also appears in a formal solution for $\delta\delta K_{[2]}^{(\ell)}$. Thus, with both formal solutions (9.28) and (9.30), we have formalized the connection between preserving $\delta\delta K_{[2]}^{(\ell)}$ data and learning from data.

With the formalities out of the way, let's now analyze our two eminent universality classes, the scale-invariant universality class and the $K^\star = 0$ universality class.

9.2 Scale-Invariant Universality Class

As a reminder, the canonical members of the scale-invariant universality class are the `ReLU` and `linear` activation functions. For a general activation function in this universality class,

$$\sigma(z) = \begin{cases} a_+ z\,, & z \geq 0, \\ a_- z\,, & z < 0, \end{cases} \tag{9.31}$$

recall from §5.2 that the helper functions and susceptibilities evaluate to

$$g^{(\ell)} = A_2 K^{(\ell)}, \tag{9.32}$$

$$h^{(\ell)} = 0, \tag{9.33}$$

$$\chi_\parallel^{(\ell)} = \chi, \tag{9.34}$$

$$\chi_\perp^{(\ell)} = \chi, \tag{9.35}$$

https://doi.org/10.1017/9781009023405.011 Published online by Cambridge University Press

with $\chi \equiv C_W A_2$. By substituting in (9.31) and performing the integrals, we can just as easily evaluate the three other Gaussian expectations that we'll need:

$$\langle \sigma(z)\sigma(z)\sigma(z)\sigma(z) \rangle_{K^{(\ell)}} = 3A_4 \left(K^{(\ell)}\right)^2 , \tag{9.36}$$

$$\langle \sigma(z)\sigma(z)\sigma'(z)\sigma'(z) \rangle_{K^{(\ell)}} = A_4 K^{(\ell)}, \tag{9.37}$$

$$\langle \sigma'(z)\sigma'(z)\sigma'(z)\sigma'(z) \rangle_{K^{(\ell)}} = A_4. \tag{9.38}$$

Here and right before, we've also made use of our previous definitions for the constants that naturally arise from these integrations:

$$A_2 \equiv \frac{a_+^2 + a_-^2}{2} , \qquad A_4 \equiv \frac{a_+^4 + a_-^4}{2}. \tag{9.39}$$

With these recollections, we are reminded of one of this class's principal characteristics: both susceptibilities are independent of the kernel and constant for all layers. With that in mind, we were able to easily satisfy criticality for the scale-invariant universality class by setting the initialization hyperparameters to

$$C_b = 0 , \qquad C_W = \frac{1}{A_2}. \tag{9.40}$$

With these tunings, both susceptibilities are set to unity, $\chi = 1$, the fixed-point value of the kernel is given in terms of the input by the expression (5.66),

$$K^\star \equiv \frac{1}{A_2} \left(\frac{1}{n_0} \sum_{j=1}^{n_0} x_j^2 \right) , \tag{9.41}$$

and the four-point vertex at criticality is given by (5.120),

$$V^{(\ell)} = (\ell - 1) \left(\frac{3A_4}{A_2^2} - 1 \right) (K^\star)^2 . \tag{9.42}$$

NTK Mean (Frozen NTK)

With the above expressions in mind, at criticality, the recursion (9.6) for the single-input frozen NTK simplifies to

$$\Theta^{(\ell+1)} = \Theta^{(\ell)} + \lambda_b^{(\ell+1)} + \lambda_W^{(\ell+1)} A_2 K^\star. \tag{9.43}$$

This recursion, together with the initial condition (9.7), is easy to solve for a given set of bias and weight learning rates.

For instance, assuming layer-independent learning rates $\lambda_b^{(\ell)} = \lambda_b$ and $\lambda_W^{(\ell)} = \lambda_W$, we find

$$\Theta^{(\ell)} = (\lambda_b + \lambda_W A_2 K^\star)\, \ell. \tag{9.44}$$

https://doi.org/10.1017/9781009023405.011 Published online by Cambridge University Press

With these uniform learning rates, we see that the frozen NTK for the scale-invariant universality class grows linearly with depth. Since the NTK involves a sum over all the previous layers, (9.28), linear growth implies that these contributions are uniform across the layers; this is in contrast to the noncritical cases, for which we would have had exponentially different contributions from the different layers of a deep network, as is clear from the formal solution (9.28). Finally, a comparison with the ansatz (9.22) implies that the critical exponent for the frozen NTK is given by $p_\Theta = -1$. We will interpret all these points further in §9.4.

NTK Variance and NTK–Preactivation Cross Correlation (Agitated NTK)

Now, let's evaluate our finite-width recursions (9.13)–(9.16) to find the NTK variance and the NTK–preactivation cross correlations.[3] First, we can simplify them by substituting in for the helper functions $g^{(\ell)} = A_2 K^{(\ell)}$ (9.32) and $h^{(\ell)} = 0$ (9.33) as well as making use of our formulae for the three other Gaussian expectations involving A_4, (9.36)–(9.38). Then, let us tune the initialization hyperparameters to criticality (9.40) by picking $C_W = 1/A_2$, which sets both susceptibilities to unity, $\chi_\parallel^{(\ell)} = \chi_\perp^{(\ell)} = 1$, and makes the kernel fixed, $K^{(\ell)} = K^\star$. With these manipulations, we get

$$D^{(\ell+1)} = D^{(\ell)} + \lambda_W A_2 \left[\left(\frac{3A_4}{A_2^2} - 1\right)(K^\star)^2 + V^{(\ell)}\right] + \left(\frac{A_4}{A_2^2} - 1\right) K^\star \Theta^{(\ell)}, \tag{9.45}$$

$$F^{(\ell+1)} = F^{(\ell)} + \frac{A_4}{A_2^2} K^\star \Theta^{(\ell)}, \tag{9.46}$$

$$B^{(\ell+1)} = B^{(\ell)} + \frac{A_4}{A_2^2}\left(\Theta^{(\ell)}\right)^2, \tag{9.47}$$

$$\begin{aligned} A^{(\ell+1)} = {} & A^{(\ell)} + (\lambda_W A_2)^2 \left[\left(\frac{3A_4}{A_2^2} - 1\right)(K^\star)^2 + V^{(\ell)}\right] \\ & + 2\lambda_W A_2 \left(\frac{A_4}{A_2^2} - 1\right) K^\star \Theta^{(\ell)} + 2\lambda_W A_2 D^{(\ell)} + \left(\frac{A_4}{A_2^2} - 1\right)\left(\Theta^{(\ell)}\right)^2 . \end{aligned} \tag{9.48}$$

Note that we have also assumed layer-independent learning rates as we did just before when working out the NTK mean.

Next, substituting in our solutions for $V^{(\ell)}$ (9.42) and $\Theta^{(\ell)}$ (9.44), we can easily solve the recursions for $D^{(\ell)}$, $F^{(\ell)}$, and $B^{(\ell)}$. Then, with our solution for $D^{(\ell)}$ in hand, we can also solve the recursion for $A^{(\ell)}$. All together, this gives the following solutions:

$$D^{(\ell)} = \frac{\ell(\ell-1)}{2}\left[\lambda_b \left(\frac{A_4}{A_2^2} - 1\right) K^\star + \lambda_W A_2 \left(\frac{4A_4}{A_2^2} - 2\right)(K^\star)^2\right], \tag{9.49}$$

$$F^{(\ell)} = \frac{\ell(\ell-1)}{2}\left[\frac{A_4}{A_2^2}\left(\lambda_b + \lambda_W A_2 K^\star\right) K^\star\right], \tag{9.50}$$

[3] You could also choose to evaluate (9.20) and then (9.21) for $D^{(\ell)}$ and $A^{(\ell)}$, respectively; it's about the same level of difficulty and obviously yields the same solution either way.

https://doi.org/10.1017/9781009023405.011 Published online by Cambridge University Press

$$B^{(\ell)} = \frac{\ell(\ell-1)(2\ell-1)}{6}\left(\frac{A_4}{A_2^2}\right)(\lambda_b + \lambda_W A_2 K^\star)^2\,, \tag{9.51}$$

$$A^{(\ell)} = \frac{\ell^3}{3}\left[\left(\frac{A_4}{A_2^2}-1\right)\lambda_b^2 + 3\left(\frac{A_4}{A_2^2}-1\right)\lambda_b\lambda_W A_2 K^\star + \left(5A_4 - 3A_2^2\right)\lambda_W^2\,(K^\star)^2\right] + \cdots\,, \tag{9.52}$$

where for $A^{(\ell)}$ we kept only the leading large-ℓ contribution.

From these four solutions, we can read off another four critical exponents for the scale-invariant universality class,

$$p_D = -2\,,\qquad p_F = -2\,,\qquad p_B = -3\,,\qquad p_A = -3, \tag{9.53}$$

which correspond to the quadratic growth of $D^{(\ell)}$ and $F^{(\ell)}$ and the cubic growth of $B^{(\ell)}$ and $A^{(\ell)}$. Combined with $p_0 = 0$ for the kernel and $p_\Theta = -1$ for the frozen NTK, we obtain the advertised ℓ/n-scaling (9.27) of the appropriately normalized quantities (9.23) and (9.24).

9.3 $K^\star = 0$ Universality Class

As a reminder, two notable members of this class are `tanh` and `sin`. More generally, the $K^\star = 0$ universality class contains any activation function with a corresponding kernel that has a nontrivial fixed point at $K^\star = 0$.

Specifically, recall from §5.3.3 that we used the following notation for the Taylor coefficients of an activation function:

$$\sigma(z) = \sum_{p=0}^{\infty}\frac{\sigma_p}{p!}z^p. \tag{9.54}$$

Then, from an analysis of the single-input kernel recursion, we learned that there's a nontrivial fixed point at $K^\star = 0$ if and only if the activation function vanishes at the origin with nonzero slope (5.89),

$$\sigma_0 = 0\,,\qquad \sigma_1 \neq 0, \tag{9.55}$$

for which we can satisfy the criticality conditions by tuning the initialization hyperparameters as (5.90),

$$C_b = 0\,,\qquad C_W = \frac{1}{\sigma_1^2}. \tag{9.56}$$

Going forward, we will assume that the bias variance and rescaled weight variance have been tuned to criticality as (9.56).

Unlike the scale-invariant universality class, the criticality analysis for the $K^\star = 0$ universality class was perturbative around $K = 0$. For this analysis, we expanded the

https://doi.org/10.1017/9781009023405.011 Published online by Cambridge University Press

helper function $g(K)$ and both susceptibilities as (5.83)–(5.85), which – now with (9.55) and (9.56) in mind – evaluate to

$$g(K) = \sigma_1^2 \left[K + a_1 K^2 + O\left(K^3\right)\right], \tag{9.57}$$

$$\chi_\parallel(K) = 1 + 2a_1 K + O\left(K^2\right), \tag{9.58}$$

$$\chi_\perp(K) = 1 + b_1 K + O\left(K^2\right), \tag{9.59}$$

where we've also recalled the following combinations of Taylor coefficients:

$$a_1 \equiv \left(\frac{\sigma_3}{\sigma_1}\right) + \frac{3}{4}\left(\frac{\sigma_2}{\sigma_1}\right)^2, \tag{9.60}$$

$$b_1 \equiv \left(\frac{\sigma_3}{\sigma_1}\right) + \left(\frac{\sigma_2}{\sigma_1}\right)^2. \tag{9.61}$$

As a reminder, to get these expressions we first Taylor-expanded their definitions (9.8), (9.10), and (9.11) in z, and then evaluated each series of Gaussian expectations to the desired order. Following the same method, we can evaluate the helper function $h(K)$ (9.9) as well as the two other Gaussian expectations needed to solve our NTK recursions:

$$h(K) = \frac{b_1}{2} + O\left(K^1\right), \tag{9.62}$$

$$\langle \sigma(z)\sigma(z)\sigma'(z)\sigma'(z)\rangle_K = \sigma_1^4 \left[K + O\left(K^2\right)\right], \tag{9.63}$$

$$\langle \sigma'(z)\sigma'(z)\sigma'(z)\sigma'(z)\rangle_K = \sigma_1^4 \left[1 + O\left(K^1\right)\right]. \tag{9.64}$$

Remembering that the parallel susceptibility characterizes the linear response of the kernel perturbations around the fixed point (5.10), we note from above that the parallel susceptibility at criticality (9.58) is close to one near the nontrivial fixed point at $K^\star = 0$. Consequently, we found a power-law large-ℓ asymptotic solution for the single-input kernel (5.94),

$$K^{(\ell)} = K^\star + \Delta K^{(\ell)} = \Delta K^{(\ell)} = \left[\frac{1}{(-a_1)}\right]\frac{1}{\ell} + O\left(\frac{\log \ell}{\ell^2}\right), \tag{9.65}$$

which slowly but surely approaches the $K^\star = 0$ nontrivial fixed point, justifying our perturbative approach to the deep asymptotics. As for the single-input four-point vertex, we previously found (5.127)

$$V^{(\ell)} = \left[\frac{2}{3a_1^2}\right]\frac{1}{\ell} + O\left(\frac{\log \ell}{\ell^2}\right), \tag{9.66}$$

and the appropriately normalized quantity (9.25) has an ℓ/n scaling:

$$\frac{V^{(\ell)}}{n\left(K^{(\ell)}\right)^2} = \left(\frac{2}{3}\right)\frac{\ell}{n} + O\left(\frac{\log(\ell)}{n}\right). \tag{9.67}$$

https://doi.org/10.1017/9781009023405.011 Published online by Cambridge University Press

NTK Mean (Frozen NTK)

Let's start with our generic formal solution (9.28) to the frozen NTK recursion (9.6). Note that for $K^\star = 0$ activation functions, the multiplicative factor (9.30) takes the form

$$\prod_{\ell''=\ell'}^{\ell-1} \chi_\perp^{(\ell'')} = \frac{\delta\delta K_{[2]}^{(\ell)}}{\delta\delta K_{[2]}^{(\ell')}} = \left(\frac{\ell'}{\ell}\right)^{p_\perp} + \cdots \tag{9.68}$$

when we plug in our large-ℓ asymptotic solution for $\delta\delta K_{[2]}^{(\ell)}$ (5.99). As a reminder, the critical exponent controlling the falloff was given in terms of the Taylor coefficient combinations, $p_\perp = b_1/a_1$, which evaluates to 1 for the `tanh` and `sin` activation functions. Plugging this multiplicative factor back into the formal solution (9.28) along with our expansion for $g(K)$ (9.57) evaluated on the asymptotic kernel (9.65), we find

$$\Theta^{(\ell)} = \sum_{\ell'=1}^{\ell} \left\{ \left[\lambda_b^{(\ell')} + \lambda_W^{(\ell')} \frac{\sigma_1^2}{(-a_1)} \left(\frac{1}{\ell'}\right) + \cdots \right] \left[\left(\frac{\ell'}{\ell}\right)^{p_\perp} + \cdots \right] \right\}. \tag{9.69}$$

Here, the factor in the first square bracket is an additive contribution picked up from the ℓ'-th layer, while the factor in the second square bracket is a multiplicative contribution from recursively passing from the ℓ'-th layer to the ℓ-th layer.

Effective theorists may take issue with two aspects of this solution (9.69) if we naively continue to choose layer-independent learning rates $\lambda_b^{(\ell)} = \lambda_b$ and $\lambda_W^{(\ell)} = \lambda_W$.

- First, notice in the first square bracket that the ℓ'-dependence of the bias term differs from the ℓ'-dependence of the weight term by a factor of $\sim (1/\ell')$. This means that the contribution of the weights to the NTK decreases with depth relative to the contribution of the biases.

- Second, notice in the second square bracket that the $\sim (\ell')^{p_\perp}$ behavior means that the NTK is dominated by contributions from deeper layers for $p_\perp > 0$ in comparison to the shallower layers. Remembering that the NTK controls the dynamics of observables (8.3), this in turn means that the training dynamics will be heavily influenced by the model parameters near the output layer.

Additionally, practical practitioners may now wonder whether these unnatural depth scalings also contribute to the empirical preference for `ReLU` over `tanh` in the deep learning community. (More on this in §9.4.)

Having said all that, we can rectify this imbalance by scaling out the layer dependence as

$$\lambda_b^{(\ell)} \equiv \widetilde{\lambda}_b \left(\frac{1}{\ell}\right)^{p_\perp}, \qquad \lambda_W^{(\ell)} \equiv \widetilde{\lambda}_W \left(\frac{1}{\ell}\right)^{p_\perp - 1}, \tag{9.70}$$

where $\widetilde{\lambda}_b$ and $\widetilde{\lambda}_W$ are layer-independent constants. Substituting this ansatz into our solution (9.69), we find

$$\Theta^{(\ell)} = \left[\widetilde{\lambda}_b + \frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)} \right] \left(\frac{1}{\ell}\right)^{p_\perp - 1} + \cdots, \tag{9.71}$$

https://doi.org/10.1017/9781009023405.011 Published online by Cambridge University Press

which manifestly balances the weight and bias contributions. Thus, we see for the $K^\star = 0$ universality class that the critical exponent for the frozen NTK is given by

$$p_\Theta = p_\perp - 1. \tag{9.72}$$

In particular, for both the `tanh` and `sin` activation functions, $p_\Theta = 0$.

NTK Variance and NTK–Preactivation Cross Correlation (Agitated NTK)

Now, let's finally deal with the agitated NTK statistics for the $K^\star = 0$ universality class. To aid our computation at criticality, let us make use of the asymptotic behavior of the kernel (9.65) and record the leading large-ℓ asymptotics of the helper functions (9.57) and (9.62), the susceptibilities (9.58) and (9.59), and the two other needed Gaussian expectations (9.63) and (9.64):

$$C_W g^{(\ell)} = \left[\frac{1}{(-a_1)}\right]\frac{1}{\ell} + \cdots, \tag{9.73}$$

$$h^{(\ell)} = \frac{b_1}{2} + \cdots, \tag{9.74}$$

$$\chi_\parallel^{(\ell)} = 1 - \frac{2}{\ell} + \cdots, \tag{9.75}$$

$$\chi_\perp^{(\ell)} = 1 - \frac{p_\perp}{\ell} + \cdots, \tag{9.76}$$

$$C_W^2 \left\langle \sigma(z)\sigma(z)\sigma'(z)\sigma'(z)\right\rangle_{K^{(\ell)}} = \left[\frac{1}{(-a_1)}\right]\frac{1}{\ell} + \cdots, \tag{9.77}$$

$$C_W^2 \left\langle \sigma'(z)\sigma'(z)\sigma'(z)\sigma'(z)\right\rangle_{K^{(\ell)}} = 1 + \cdots. \tag{9.78}$$

Additionally, going forward we will assume that the bias and weight learning rates have the layer dependence (9.70) motivated by equal per-layer NTK contribution.

With all this out of the way, it's straightforward to evaluate the large-ℓ asymptotics of $F^{(\ell)}$ and $B^{(\ell)}$. Plugging in the above expressions and the frozen NTK asymptotic solution (9.71) into their recursions (9.14) and (9.15), we get

$$F^{(\ell+1)} = \left[1 - \frac{4}{\ell} + \cdots\right] F^{(\ell)} + \left\{\left[\frac{1}{(-a_1)}\right]\left[\widetilde{\lambda}_b + \frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)}\right]\left(\frac{1}{\ell}\right)^{p_\perp} + \cdots\right\}, \tag{9.79}$$

$$B^{(\ell+1)} = \left[1 - \frac{2p_\perp}{\ell} + \cdots\right] B^{(\ell)} + \left\{\left[\widetilde{\lambda}_b + \frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)}\right]^2\left(\frac{1}{\ell}\right)^{2p_\perp - 2} + \cdots\right\}. \tag{9.80}$$

Substituting in our scaling ansatz (9.22), they have the following asymptotic solutions at large ℓ:

$$F^{(\ell)} = \frac{1}{(5-p_\perp)}\left[\frac{1}{(-a_1)}\right]\left[\widetilde{\lambda}_b + \frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)}\right]\left(\frac{1}{\ell}\right)^{p_\perp - 1} + \cdots, \tag{9.81}$$

$$B^{(\ell)} = \frac{1}{3}\left[\widetilde{\lambda}_b + \frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)}\right]^2\left(\frac{1}{\ell}\right)^{2p_\perp - 3} + \cdots. \tag{9.82}$$

https://doi.org/10.1017/9781009023405.011 Published online by Cambridge University Press

Next, for the $D^{(\ell)}$ recursion, let's start with the slightly more compact expression (9.20). Plugging in the expressions (9.73)–(9.78) along with the learning rates (9.70) and the asymptotic solutions for the four-point vertex (9.66) and the frozen NTK (9.71), we get a recursion

$$D^{(\ell+1)} = \left[1 - \frac{(p_\perp + 2)}{\ell} + \cdots\right] D^{(\ell)} + \left\{ \left[\frac{2}{3(-a_1)}\right] \left[-p_\perp \widetilde{\lambda}_b - (p_\perp - 1)\frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)} \right] \left(\frac{1}{\ell}\right)^{p_\perp} + \cdots \right\}. \tag{9.83}$$

This recursion can also be easily solved by using our scaling ansatz (9.22), giving

$$D^{(\ell)} = \frac{-2}{9(-a_1)} \left[p_\perp \widetilde{\lambda}_b + (p_\perp - 1)\frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)} \right] \left(\frac{1}{\ell}\right)^{p_\perp - 1} + \cdots . \tag{9.84}$$

Finally, for $A^{(\ell)}$ recursion (9.21), the by-now-familiar routine of flipping back and forth in your book and plugging in the large-ℓ asymptotic expressions (9.73)–(9.78), the learning rates (9.70), and the asymptotic solutions for the four-point vertex (9.66), for the frozen NTK (9.71), and for $D^{(\ell)}$ (9.84), gives

$$A^{(\ell+1)} = \left[1 - \frac{2p_\perp}{\ell} + \cdots\right] A^{(\ell)} + \left\{ \frac{4}{9} \left[p_\perp \widetilde{\lambda}_b + (p_\perp - 1)\frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)} \right]^2 \left(\frac{1}{\ell}\right)^{2p_\perp - 2} + \cdots \right\}, \tag{9.85}$$

which can be solved using the same large-ℓ scaling ansatz (9.22), giving

$$A^{(\ell)} = \frac{4}{27} \left[p_\perp \widetilde{\lambda}_b + (p_\perp - 1)\frac{\widetilde{\lambda}_W \sigma_1^2}{(-a_1)} \right]^2 \left(\frac{1}{\ell}\right)^{2p_\perp - 3} + \cdots . \tag{9.86}$$

With this, we complete our evaluation of the agitated NTK statistics for the $K^\star = 0$ universality class.[4]

Having solved all the recursions we have, let's collect and recollect the critical exponents. From (9.81) and (9.84) we collect $p_F = p_D = p_\perp - 1$, while from (9.82) and (9.86) we collect $p_B = p_A = 2p_\perp - 3$. Recollecting $p_0 = 1$ for the kernel and $p_\Theta = p_\perp - 1$ for the frozen NTK, these critical exponents for the $K^\star = 0$ universality class again obey ℓ/n scaling (9.27) for the normalized quantities defined in (9.23) and (9.24). Together with our scale-invariant results (9.53), this means the posited relations (9.27) do indeed persist across universality classes as scaling laws.

In summary, we have found that the leading finite-width behavior of the NTK–preactivation joint distribution – as measured by the NTK variance and NTK–preacitvation cross correlation – has a *relevant* ℓ/n scaling regardless of activation function, as is natural according to the principles of our effective theory.

[4] Curiously, for activation functions with $p_\perp = 1$, such as `tanh` and `sin`, the single-input tensors $D^{(\ell)}$ and $A^{(\ell)}$ are independent of the weight learning rate at leading order.

https://doi.org/10.1017/9781009023405.011 Published online by Cambridge University Press

9.4 Criticality, Exploding and Vanishing Problems, and None of That

Having now analyzed the NTK statistics of deep networks, let us culminate our discussion by revisiting our original motivation for criticality: *exploding and vanishing problems*. In particular, let us finally introduce – and then immediately abolish – the exploding and vanishing gradient problem.[5]

Traditional View on the Exploding and Vanishing Gradient Problem

Traditionally, the exploding and vanishing gradient problem is manifested by considering the behavior of the gradient of the loss for a deep network. Using the chain rule twice, the derivative of the loss with respect to a model parameter $\theta_\mu^{(\ell)}$ in the ℓ-th layer – either a bias $\theta_\mu^{(\ell)} \equiv b_j^{(\ell)}$ or a weight $\theta_\mu^{(\ell)} \equiv W_{jk}^{(\ell)}$ – takes the form

$$\frac{d\mathcal{L}_\mathcal{A}}{d\theta_\mu^{(\ell)}} = \sum_{\alpha\in\mathcal{D}} \sum_{i_L=1}^{n_L} \sum_{i_\ell=1}^{n_\ell} \epsilon_{i_L;\alpha} \frac{dz_{i_L;\alpha}^{(L)}}{dz_{i_\ell;\alpha}^{(\ell)}} \frac{dz_{i_\ell;\alpha}^{(\ell)}}{d\theta_\mu^{(\ell)}} . \tag{9.87}$$

In this gradient, the first factor is the *error factor* (8.14)

$$\epsilon_{i;\alpha} \equiv \frac{\partial\mathcal{L}_\mathcal{A}}{\partial z_{i;\alpha}^{(L)}} , \tag{9.88}$$

the final factor is a *trivial factor* (8.11)

$$\frac{dz_{i;\alpha}^{(\ell)}}{db_j^{(\ell)}} = \delta_{ij} , \qquad \frac{dz_{i;\alpha}^{(\ell)}}{dW_{jk}^{(\ell)}} = \delta_{ij}\, \sigma_{k;\alpha}^{(\ell-1)} , \tag{9.89}$$

and the middle factor is the *chain-rule factor*

$$\frac{dz_{i_L;\alpha}^{(L)}}{dz_{i_\ell;\alpha}^{(\ell)}} = \sum_{i_{\ell+1},\dots,i_{L-1}} \frac{dz_{i_L;\alpha}^{(L)}}{dz_{i_{L-1};\alpha}^{(L-1)}} \cdots \frac{dz_{i_{\ell+1};\alpha}^{(\ell+1)}}{dz_{i_\ell;\alpha}^{(\ell)}} = \sum_{i_{\ell+1},\dots,i_{L-1}} \prod_{\ell'=\ell}^{L-1} \left[W_{i_{\ell'+1}i_{\ell'}}^{(\ell'+1)} \sigma'^{(\ell')}_{i_{\ell'};\alpha} \right] , \tag{9.90}$$

which can be derived by iterating the backward equation (8.17) or equivalently by repeatedly using the chain rule in conjunction with the MLP forward equation (8.9). If this text causes you to experience a large error factor yourself, please flip backward to §8.0 and review our discussion of the *backpropagation* algorithm.

The point is that without any fine-tuning, the product of matrices from layer ℓ to layer L in the chain-rule factor (9.90) will generically lead to exponential behavior. Even

[5] This problem was first noticed [58, 59] in the context of training (the now somewhat deprecated) *recurrent neural networks* (RNNs), during the era when neural networks were still *neural networks* and not yet *deep learning*, that is, at a time when MLPs weren't yet deep enough for this to have been an obvious issue.

https://doi.org/10.1017/9781009023405.011 Published online by Cambridge University Press

for networks of moderate depth, this makes it extremely difficult for the shallower-layer parameters to receive a well-behaved gradient and consequentially be properly trained: a vanishing gradient means that such parameters receive no training signal from the data and loss, while an exploding gradient is indicative of an instability in which the loss may increase or even blow up. This is the **exploding and vanishing gradient problem**. In a sense, this is a backward iteration dual of the already familiar *exploding and vanishing kernel problem* that arises from the forward iteration equation.[6]

Of course, not only does the chain-rule factor (9.90) need to be well behaved for stable training, but the error factor (9.88) and trivial factor (9.89) must be as well. As we'll explain next, these latter factors are directly tied to the exploding and vanishing kernel problem. However, we'll also see that our well-understood notion of criticality is already sufficient to mitigate both exploding and vanishing problems together.

Critical View on the Exploding and Vanishing Gradient Problem

Critically, let us recall from §3.2 and then §5.1 our discussion of the exploding and vanishing kernel problem. In those sections, we first motivated criticality as remedying exponential behavior in the kernel $K^{(\ell)}_{\alpha_1\alpha_2}$. As the L-th-layer kernel controls the *typical* values of the network output – and as the dataset's labels are generically order-one numbers – we suggested that such an exploding or vanishing kernel would be problematic for training. Now that we know a little about gradient descent, we can actually see a more direct manifestation of this instability by considering all the factors that make up the network's gradient, (9.87).

First, let's see how the error factor (9.88) is tied to the *exploding* kernel problem. For example, the error factor for the MSE loss (7.2) is given by (7.15)

$$\epsilon_{i;\alpha} = z^{(L)}_{i;\alpha} - y_{i;\alpha} . \tag{9.91}$$

As you can clearly see, if the kernel explodes, then the typical output – and hence typical values of the error factor – will explode as well.[7] To ensure this does not happen, we must set $\chi_{\parallel}(K^\star) \leq 1$.

Second, notice that the trivial factor (9.89) for the weights is proportional to the activation. For activation functions contained in either of the scale-invariant or $K^\star = 0$ universality classes, if the kernel – and consequently the typical preactivation – is exponentially small, then the activation – and consequently the trivial factor – will be exponentially suppressed. Subsequently, the weights in the deeper layers of the network

[6] If you'd like, you can see this duality concretely by considering a deep linear network, for which the statistics of such a product of weights can be worked out exactly as in §3.

[7] Getting ahead of ourselves, a precocious reader might wonder whether this matters for the *cross-entropy loss*, since for that loss the error factor will stay of order one even if the network output explodes. However, in this case the model would then be (exponentially) overconfident on its predictions, and such an inductive bias would be difficult to correct via training.

https://doi.org/10.1017/9781009023405.011 Published online by Cambridge University Press

would struggle to train as they only receive an exponentially small update. Thus, in order to avoid this *vanishing* kernel problem, we demand $\chi_{\|}(K^\star) \geq 1$.[8]

Combining these two observations, we see that the exploding and vanishing kernel problem is directly manifested as a subproblem of the exploding and vanishing gradient problem and further see how our criticality condition imposed on the parallel susceptibility, $\chi_{\|}(K^\star) = 1$, serves to mitigate it.

Moreover, we can shed further light on the vanishing of the trivial factor by considering its embodiment in the NTK. Considering our formal solution for the frozen NTK (9.28) and recalling the original definition (8.7), we can track the contribution of the weight derivatives as leading to the additive term $\lambda_W^{(\ell')} g^{(\ell'-1)}$. For an exponentially vanishing kernel, this factor is exponentially suppressed as

$$\lambda_W^{(\ell')} g^{(\ell'-1)} \propto K^{(\ell'-1)} \lll 1, \tag{9.92}$$

since $g(K) = A_2 K$ (9.32) for the scale-invariant universality class, and $g(K) = \sigma_1^2 K + O(K^2)$ (9.57) for the $K^\star = 0$ universality class. This is another way of seeing that such deeper-layer weights are not contributing to the training dynamics and, in particular, also implies that such weights will have a minimal effect on the updates to *other* parameters.

Similarly, we see that the chain-rule factor (9.90) is also encoded in the NTK in the multiplicative factor $\prod_{\ell''=\ell'}^{\ell-1} \chi_{\perp}^{(\ell'')}$ (9.30). In fact, such a factor was secretly always lurking in the NTK forward equation (8.8) or (8.12) as the coefficient of the recursive term. To disclose that secret, note that in the infinite-width limit, the expectation of the chain-rule factor factorizes, and we see from (8.8) or (8.12) that

$$\mathbb{E}\left[\sum_{j_1,j_2} \frac{dz_i^{(\ell+1)}}{dz_{j_1}^{(\ell)}} \frac{dz_i^{(\ell+1)}}{dz_{j_2}^{(\ell)}}\right] = \mathbb{E}\left[\sum_{j_1,j_2} W_{ij_1}^{(\ell+1)} W_{ij_2}^{(\ell+1)} \sigma'^{(\ell)}_{j_1} \sigma'^{(\ell)}_{j_2}\right] = C_W \langle \sigma'(z)\sigma'(z)\rangle_{K^{(\ell)}} = \chi_{\perp}^{(\ell)}, \tag{9.93}$$

thus making this connection explicit.

As we discussed in §9.1 under the heading *Formalities: Perpendicular Perturbations and the Frozen NTK*, we need to ensure that these multiplicative chain-rule factors are under control for training to be well behaved, on average. In particular, if $\chi_{\perp}(K^\star) > 1$, then the deeper-layer NTKs will exponentially explode, and the training dynamics will be unstable, potentially leading to growing losses. If instead $\chi_{\perp}(K^\star) < 1$, then the contribution of the biases and weights in the shallower layers to the deeper-layer NTKs will be exponentially diminished. This means both that such parameters will struggle to move via gradient-descent updates and *also* that they will struggle to influence the evolution of the parameters in the deeper layers.

All in all, we see that contribution of the chain-rule factor to the exploding and vanishing gradient problem is directly connected to an exponentially growing or decaying

[8] Incidentally, for the scale-invariant universality class, this same logic provides an additional justification for avoiding the *exploding* kernel problem. That is, since scale-invariant activation functions don't *saturate*, if $\chi_{\|}(K^\star) > 1$, then the activation – and consequentially the trivial factor – would explode.

https://doi.org/10.1017/9781009023405.011 Published online by Cambridge University Press

multiplicative factor in the NTK and further see how our criticality condition imposed on the perpendicular susceptibility, $\chi_{\perp}(K^{\star}) = 1$, serves to mitigate it.[9]

An Equivalence Principle for Learning Rates

Although originally derived by considering the behavior of the kernel recursion, we just recovered both of our criticality conditions $\chi_{\parallel}(K^{\star}) = 1$ and $\chi_{\perp}(K^{\star}) = 1$ by a direct analysis of gradient-descent updates and of the NTK forward equation. Importantly, the guiding principle we found was that each layer should not make an exponentially different contribution to the NTK.

However, there's really no reason for us to stop at the exponential level. In fact, we can further demand at the polynomial level that no type of model parameter and no layer dominate over another for training. In other words, rather than requiring *more or less equal* contributions from the parameters in different layers, we demand parametrically *equal* contributions to the NTK for each parameter group from every layer. This gives an **equivalence principle** for setting the training hyperparameters, i.e., the bias and weight learning rates. In fact, en route to this section, we already found a way to satisfy this equivalence principle for each of our universality classes.

As we retroactively saw in §9.2, the equivalence principle was easily met for activation functions contained in the scale-invariant universality class by setting the bias and weight learning rates to be layer-independent:

$$\eta\lambda_b^{(\ell)} = \frac{\eta\widetilde{\lambda}_b}{L}\,, \qquad \frac{\eta\lambda_W^{(\ell)}}{n_{\ell-1}} = \frac{\eta\widetilde{\lambda}_W}{L n_{\ell-1}}\,. \tag{9.94}$$

Here, we have also re-emphasized the rescaling of the weight learning rates by width of the previous layer as we discussed multiple times in §8.0. In particular, you should understand the *parametrically equal contributions* provision of equivalence principle as requiring appropriate depth *and* width scalings. Also here – with our discussion in §8.0 under the heading *Scaling in the Effective Theory* in mind – note that we rescaled the learning rates by the overall depth L of the network so as to have an order-one NTK in the output layer. This ensures that we can naturally compare these rescaled learning

[9] Having now explained why *criticality* is a complete solution to the exploding and vanishing gradient problem, let us discuss remedies of *traditionality*.

One of the first heuristic solutions – first discussed in the context of recurrent neural networks – is gradient clipping [60], in which the norm of the gradient is reduced whenever it exceeds a certain threshold. As should be apparent given our discussion, such an ad hoc, distortionary, and hard-to-tune heuristic is completely unnecessary – and potentially even destructive – in networks that are at criticality.

A second heuristic solution was the adoption of the `ReLU`. Recall that activation functions such as the `tanh` and the `sigmoid` *saturate* when $|z| \to \infty$. This implies that the derivative of the activation vanishes upon saturation as $\sigma'(z) = 0$. Such saturation naturally leads to vanishing gradients, as we can easily see from the right-hand side of (9.90). It is partially within this context that practitioners adopted the `ReLU` over saturating activation functions such as the `tanh` (see, e.g., [19]). However, with our deeper and more critical understanding, we now appreciate that criticality is sufficient to mitigate this vanishing gradient problem for any activation function that admits critical initialization hyperparameters, even for saturating ones like `tanh`.

https://doi.org/10.1017/9781009023405.011 Published online by Cambridge University Press

rates $\eta\widetilde{\lambda}_b$ and $\eta\widetilde{\lambda}_W$ across models of different depths as well as that we have properly scaled order-one changes in observables, e.g., the loss, after any gradient-descent update.

Meanwhile, for the $K^\star = 0$ universality class, we retroactively saw in §9.3 that the equivalence principle requires

$$\eta\lambda_b^{(\ell)} = \eta\widetilde{\lambda}_b \left(\frac{1}{\ell}\right)^{p_\perp} L^{p_\perp - 1}\,, \qquad \frac{\eta\lambda_W^{(\ell)}}{n_{\ell-1}} = \frac{\eta\widetilde{\lambda}_W}{n_{\ell-1}} \left(\frac{L}{\ell}\right)^{p_\perp - 1}\,, \tag{9.95}$$

where here we recall our discussion of having separate depth scalings for the bias and weight learning rates (9.70) as a way to ensure both uniform contributions in parameter groups and in layers to the asymptotic frozen NTK solution (9.71).[10] In particular, for odd smooth activation functions such as `tanh` and `sin`, the critical exponent for perpendicular perturbations is given by $p_\perp = 1$, and we have a simpler prescription:

$$\eta\lambda_b^{(\ell)} = \frac{\eta\widetilde{\lambda}_b}{\ell}\,, \qquad \frac{\eta\lambda_W^{(\ell)}}{n_{\ell-1}} = \frac{\eta\widetilde{\lambda}_W}{n_{\ell-1}}. \tag{9.96}$$

With this, we further wonder whether the empirical preference for `ReLU` over `tanh` is at least partially due to the fact that the `ReLU` learning rates are naturally ℓ-independent (9.94), while the `tanh` learning rates require nontrivial ℓ-rescalings (9.96).

In conclusion, while the optimal values of the order-one training hyperparameters $\eta\widetilde{\lambda}_b$ and $\eta\widetilde{\lambda}_W$ surely depend on the specifics of a task, we expect that the layer ℓ, depth L, and width $n_{\ell-1}$ scalings dictated by the equivalence principle will lead to the least variation across network architectures with differing widths n_ℓ and depths L.

[10] Please do not confuse these ℓ-rescalings with L-rescalings. The former modifies the learning rates of a given layer ℓ by a factor of ℓ, and the latter modifies the learning rates of any layer in the network by the overall network depth L.

Also, with the recent critical discussion of the exploding and vanishing kernel problem in mind, you should see that this relative ℓ-scaling between the bias and weight learning rates is just a polynomial version of the vanishing kernel problem: the equivalence principle ensures that deeper-layer weights receive polynomially nonvanishing gradients as well as contribute polynomially-equally to the training dynamics of other parameters.

https://doi.org/10.1017/9781009023405.011 Published online by Cambridge University Press