Lecture 09: Large Optimized Neural Networks Are Equivalent to NTK Least Norm Interpolators

Geoff Pleiss

In the last lecture we examined a first-order Taylor approximation of neural networks:

$$f(\boldsymbol{x};\boldsymbol{\theta}) \approx \tilde{f}(\boldsymbol{x};\boldsymbol{\theta}) := f(\boldsymbol{x};\boldsymbol{\theta}^{(0)}) + \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}^{(0)})^\top(\boldsymbol{\theta}-\boldsymbol{\theta}^{(0)}), \tag{1}$$

where $\boldsymbol{\theta}^{(0)}$ is the random variable representing the parameters at initialization. This approximation, known as the *neural tangent kernel (NTK)* approximation, is linear with respect to the learnable parameters $\boldsymbol{\theta}$. Moreover, we saw that, when $\|\boldsymbol{\theta}-\boldsymbol{\theta}^{(0)}\|$ does not grow to quickly with the number of hidden units d, then this Taylor approximation becomes more exact as $d \to \infty$.

We finally arrive at a major result of this course: under certain optimization conditions, the NTK approximation is arbitrarily good for large neural networks *even after optimization.* More specifically

Theorem 1 (Jacot et al. [2018], simplified)**.** *Let $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$ be a training dataset, and let $f(\boldsymbol{x};\boldsymbol{\theta})$ be a (multiple layer) neural network with d parameters in each hidden layer. If*

- *the entries of $\boldsymbol{\theta}^{(1)}$ are i.i.d. $\mathcal{N}(0,1)$,*
- *the non-linearity σ has bounded second derivative, and*
- *we run gradient descent with an infinitesimally small learning rate on the squared error loss*

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= \frac{1}{2n}\sum_{i=1}^{n}(f(\boldsymbol{x}_i;\boldsymbol{\theta}) - y_i)^2 \\ &= \frac{1}{2}\|\boldsymbol{f}_{\boldsymbol{\theta}} - \boldsymbol{y}\|_2^2,\end{aligned} \qquad \boldsymbol{f}_{\boldsymbol{\theta}} = \begin{bmatrix} f(\boldsymbol{x}_1;\boldsymbol{\theta}) \\ \vdots \\ f(\boldsymbol{x}_n;\boldsymbol{\theta})\end{bmatrix},$$

then as $d \to \infty$ the neural network almost surely converges to the interpolator:

$$\hat{f}(\boldsymbol{x}) = \begin{bmatrix} k(\boldsymbol{x},\boldsymbol{x}_1)\cdot k(\boldsymbol{x},\boldsymbol{x}_n)\end{bmatrix} \begin{bmatrix} k(\boldsymbol{x}_1,\boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_1,\boldsymbol{x}_n) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_n,\boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_n,\boldsymbol{x}_n)\end{bmatrix}^{-1} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix},$$
$$k(\boldsymbol{x},\boldsymbol{x}') = \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}^{(0)})^\top \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}';\boldsymbol{\theta}^{(0)}),$$

A few remarks are in order about Theorem 1:

- $\hat{f}(\boldsymbol{x})$ is an interpolator, and thus has zero training error. We thus achieve a *global optimum* to our least squares optimization problem, despite optimizing a function that is non-convex w.r.t. the learnable parameters!
- $k(\boldsymbol{x},\boldsymbol{x}')$ is the **Neural Tangent Kernel (NTK)**. As discussed in the last lectures, we can compute $k(\boldsymbol{x},\boldsymbol{x}')$ in closed form any neural network architecture. For our simple neural network, the spectrum of $k(\boldsymbol{x},\boldsymbol{x}')$ decays polynomially [Geifman et al., 2020].
- $\hat{f}(\boldsymbol{x})$ is the *least norm interpolator* in the RKHS defined by the NTK $k(\boldsymbol{x},\boldsymbol{x}')$. In other words, we would arrive at $\hat{f}(\boldsymbol{x})$ if we had performed gradient descent on $\tilde{f}(\boldsymbol{x};\boldsymbol{\theta})$ rather than on $f(\boldsymbol{x};\boldsymbol{\theta})$. (or rather the least norm interpolator in some RKHS)

- We can analyze the risk of least norm linear models (and the least norm interpolator in RKHS) using the high-dimensional asymptotic limits we discussed in previous lectures.
- This convergence happens as $d \to \infty$, so, perhaps counterintuitively, neural networks become *better behaved* and *easier to characterize* as they get larger.

---

1) High-Level Intuition

Theorem 1 appeared simultaneously in many publications [Jacot et al., 2018, Allen-Zhu et al., 2019, Du et al., 2019] all of which used fairly complex arguments that relied on specific neural network conditions.

[Chizat et al., 2019] provided what I consider to be the simplest proof, which demonstrates a much more general phenomenon about first-order Taylor approximations. At a high level, for large neural networks, the loss decreases rapidly without the neural network leaving the region where $\tilde{f}(\boldsymbol{x}; \boldsymbol{\theta})$ is a good approximation.

Consider a gradient descent step $\boldsymbol{\theta}^{(1)} = \boldsymbol{\theta}^{(0)} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(0)})$.

- Assuming we take a small enough step, we have that

  $$\frac{\mathcal{L}(\boldsymbol{\theta}^{(0)}) - \mathcal{L}(\boldsymbol{\theta}^{(1)})}{\boldsymbol{\theta}^{(1)} - \boldsymbol{\theta}^{(0)}} \approx \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(0)}).$$

  and thus

  $$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}^{(0)}) - \mathcal{L}(\boldsymbol{\theta}^{(1)}) &\approx \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(0)})^\top \left(\boldsymbol{\theta}^{(1)} - \boldsymbol{\theta}^{(0)} \right) \\ &= \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(0)})^\top \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(0)}) = \eta \left\| \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(0)}) \right\|^2 . \end{aligned}$$

- At the same time, we want to see how much the *Jacobian* of the neural network changes. If $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}^{(1)}) \approx \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}^{(0)})$ and η is small enough, then the neural network is close to its first order Taylor approximation about $\boldsymbol{\theta}^{(0)}$. Defining the **derivative operator norm** as

  $$\| \nabla_{\boldsymbol{\theta}} f(\cdot; \boldsymbol{\theta}_0) \| := \sup_{\boldsymbol{x}} \| \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}_0) \|,$$

  (i.e. the largest norm that the Jacobian can take under any input $\boldsymbol{x}$), we have that

  $$\begin{aligned} \| \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}^{(1)}) - \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}^{(0)}) \| &\approx \left\| \nabla^2_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}^{(0)}) \left(\boldsymbol{\theta}^{(1)} - \boldsymbol{\theta}^{(0)} \right) \right\| \\ &= \left\| \eta \nabla^2_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}^{(0)}) \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(0)}) \right\| \\ &\leq \eta \| \nabla^2_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}^{(0)}) \| \| \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(0)}) \| \\ &\leq \eta \| \nabla^2_{\boldsymbol{\theta}} f(\cdot; \boldsymbol{\theta}^{(0)}) \| \| \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(0)}) \|, \end{aligned}$$

  where $\| \nabla_{\boldsymbol{\theta}} f(\cdot; \boldsymbol{\theta}^{(0)}) \|$ is the **Hessian operator norm**, defined as $\| \nabla_{\boldsymbol{\theta}} f(\cdot; \boldsymbol{\theta}^{(0)}) \| := \sup_{\boldsymbol{x}} \| \nabla^2_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}^{(0)}) \|$. The penultimate inequality is a standard matrix norm bound; the last inequality follows from the definition of the operator norm.

- If the *relative change* of the loss is much faster than the *relative change of the Jacobian* (i.e. we can optimize the loss faster than the neural network loses its first order approximation), then:

$$\frac{1}{\eta} \frac{\mathcal{L}(\boldsymbol{\theta}^{(0)}) - \mathcal{L}(\boldsymbol{\theta}^{(1)})}{\mathcal{L}(\boldsymbol{\theta}^{(0)})} \approx \frac{\left\| \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(0)}) \right\|^{\cancel{2}}}{\mathcal{L}(\boldsymbol{\theta}^{(0)})} \gg \frac{\| \nabla^2_{\boldsymbol{\theta}} f(\cdot; \boldsymbol{\theta}^{(0)}) \| \cancel{\| \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(0)}) \|}}{\| \nabla_{\boldsymbol{\theta}} f(\cdot; \boldsymbol{\theta}^{(0)}) \|} \gtrapprox \frac{1}{\eta} \frac{\| \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}^{(1)}) - \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}^{(0)}) \|}{\| \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}^{(0)}) \|}. \tag{2}$$

Plugging in $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}\|\hat{\boldsymbol{f}}_{\boldsymbol{\theta}} - \boldsymbol{y}\|_2^2$, we have that

$$\begin{aligned} \|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})\| &= \left\|\nabla_{\boldsymbol{\theta}}\boldsymbol{f}_{\boldsymbol{\theta}}^\top \left(\boldsymbol{f}_{\boldsymbol{\theta}} - \boldsymbol{y}\right)\right\| \\ &\leq \|\nabla_{\boldsymbol{\theta}}\boldsymbol{f}_{\boldsymbol{\theta}}^\top\|\|\boldsymbol{f}_{\boldsymbol{\theta}} - \boldsymbol{y}\|, \end{aligned} \qquad \nabla_{\boldsymbol{\theta}}\boldsymbol{f}_{\boldsymbol{\theta}} = \begin{bmatrix}\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_1;\boldsymbol{\theta}) & \vdots & \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_n;\boldsymbol{\theta})\end{bmatrix} \in \mathbb{R}^{n\times d}.$$

Since each column in $\nabla_{\boldsymbol{\theta}}\boldsymbol{f}_{\boldsymbol{\theta}}$ has norm at most $\|\nabla_{\boldsymbol{\theta}} f(\cdot;\boldsymbol{\theta})\|$ we have that $\|\nabla_{\boldsymbol{\theta}}\boldsymbol{f}_{\boldsymbol{\theta}}\| \leq \sqrt{n}\|\nabla_{\boldsymbol{\theta}} f(\cdot;\boldsymbol{\theta})\|$. Thus, the condition of Eq. (2) is satisfied if

$$\frac{\sqrt{n}\|\nabla_{\boldsymbol{\theta}} f(\cdot;\boldsymbol{\theta}^{(0)})\|\|\boldsymbol{f}_{\boldsymbol{\theta}} - \boldsymbol{y}\|}{\frac{1}{2}\|\boldsymbol{f}_{\boldsymbol{\theta}} - \boldsymbol{y}\|^2} \gg \frac{\|\nabla^2_{\boldsymbol{\theta}} f(\cdot;\boldsymbol{\theta}^{(0)})\|}{\|\nabla_{\boldsymbol{\theta}} f(\cdot;\boldsymbol{\theta}^{(0)})\|},$$

or, after rearranging terms (and recognizing that $2\sqrt{n}$ is a constant factor that doesn't matter), if

$$1 \gg \frac{\|\nabla^2_{\boldsymbol{\theta}} f(\cdot;\boldsymbol{\theta}^{(0)})\|}{\|\nabla_{\boldsymbol{\theta}} f(\cdot;\boldsymbol{\theta}^{(0)})\|^2}\|\boldsymbol{f}_{\boldsymbol{\theta}} - \boldsymbol{y}\| := \kappa \tag{3}$$

Several fortuitous things happen with wide neural networks with one hidden layer that use our $1/\sqrt{d}$ scaling:

$$f(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{\sqrt{d}}\sum_{i=1}^{d}\beta_i\sigma(\boldsymbol{w}_i^\top\boldsymbol{x}), \qquad \boldsymbol{\theta} = \begin{bmatrix}\boldsymbol{w}_1 & \cdots & \boldsymbol{w}_d & \beta_1 & \cdots & \beta_d\end{bmatrix}^\top.$$

- Each term in the Jacobian $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0)$ inherits a factor of $1/\sqrt{d}$, and there are $O(d)$ elements in the term. Thus, $\|\nabla_{\boldsymbol{\theta}} f(\cdot;\boldsymbol{\theta}^{(0)})\|_2^2 \asymp O(1)$.
- The Hessian $\nabla^2_{\boldsymbol{\theta}} f(\cdot;\boldsymbol{\theta}^{(0)})$ also inherits the factor of $1/\sqrt{d}$. By inspection, the only non-zero terms in the Hessian will be β_i, w_i pairs and w_i, w_i diagonal entries. If $\ddot{\sigma}$ is bounded, then all non-zero terms in the Hessian are $O(1/\sqrt{d})$, where the $1/\sqrt{d}$ factor simply comes from the scaling. Each row/column of the Hessian has at most 2 non-zero terms, so $\|\nabla^2_{\boldsymbol{\theta}} f(\cdot;\boldsymbol{\theta}^{(0)})\| \asymp O(1/d)$.
- Putting these two facts together, we have that κ in Eq. (3) is $O(1/\sqrt{d})$.

Zooming out. To summarize

- We've defined a κ ratio that characterizes how fast the loss of our neural network decreases relative to the change in the Jacobian (i.e. deviation from the first order Taylor approximation) during any small gradient step.
- Because of the $1/\sqrt{d}$ scaling in wide neural networks, we have that $\kappa \asymp O(1/\sqrt{d})$.
- Thus, as we run gradient descent, we will reduce the loss of our neural network while remaining nearly linear with respect to the learnable parameters!
- Gradient descent is biased towards minimum norm solutions in linear models; thus we know that the neural network will converge to the least norm interpolator in the RKHS defined by the NTK.

---

2) Proof Strategy

Here are the steps we would need to prove this result more rigorously (see [Chizat et al., 2019] for details):

- We define a *very small radius* $B = \sigma_{\min}/2\gamma$, where $\sigma_{\min}$ is the minimum singular value of the initial Jacobian and γ is the maximum value of $\ddot{\sigma}$. We study neural network optimization under the conditions that $\|\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)}\| \leq B$.
- Because we are running gradient descent with an infinitesimally small learning rate, we can rewrite gradient descent as differential equation:

$$\frac{d\boldsymbol{\theta}^{(t)}}{dt} = -\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(t)}).$$

- We then prove, using standard differential equation theory, that $\mathcal{L}(\boldsymbol{\theta}^{(t)})$ decays exponentially quickly as long as $\|\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^{(0)}\| \leq B$.
- We then prove, again using standard differential equation theory, that $\boldsymbol{\theta}^{(t)}$ remains close to $\boldsymbol{\theta}^{(0)}$, and—if d is large enough—$\|\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^{(0)}\| \leq B$ for all $t \in [0, \infty)$.
- Finally, we show that if we had run gradient descent on the first order Taylor approximation $\tilde{f}(\boldsymbol{x}; \boldsymbol{\theta})$ to obtain parameters $\tilde{\boldsymbol{\theta}}^{(t)}$, then the difference in parameters $\|\tilde{\boldsymbol{\theta}}^{(t)} - \boldsymbol{\theta}^{(t)}\|$ decays with d because the Jacobian does not change too much.

3) Limitations of NTK Analysis

There are a couple of limitations to the NTK analysis:

1. We require wide neural networks. Many neural networks used in practice are not too wide relative to their depth. A high-dimensional asymptotic approach (scaling width and depth to infinity simultaneously) can reveal different asymptotic properties Li et al. [2021].
2. This setup isn't totally realistic. It requires that we run gradient descent with an infinitesimally small learning rate, a regression loss, and full-batch gradient descent.
3. Reducing neural networks to kernel machines loses the "magic" of neural networks. Recall that kernel machines are linear models, and linear models have been around for forever. Surely if neural networks were just linear models, then we would have been able to replicate their successes long ago.

In an upcoming lecture we will study neural networks that deviate from this linearity, and we will see that this analysis can explain some empirical properties that are not captured by the NTK approximation. Nevertheless, the NTK analysis provides a significant amount of insight. It tells us that we do not have to be "scared" of large neural networks (in terms of overfitting or poor optimization), and it gives us a useful tool to approximate the inductive bias of neural networks.

References

Z. Allen-Zhu, Y. Li, and Z. Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252, 2019.

L. Chizat, E. Oyallon, and F. Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

S. Du, J. Lee, H. Li, L. Wang, and X. Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685, 2019.

A. Geifman, A. Yadav, Y. Kasten, M. Galun, D. Jacobs, and B. Ronen. On the similarity between the laplace and neural tangent kernels. *Advances in Neural Information Processing Systems*, 33:1451–1461, 2020.

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, pages 8571–8580, 2018.

M. Li, M. Nica, and D. Roy. The future is log-Gaussian: ResNets and their infinite-depth-and-width limit at initialization. In *Advances in Neural Information Processing Systems*, volume 34, pages 7852–7864, 2021.