Lecture 07: Linear Approximations of Neural Networks

Geoff Pleiss

Today we return to neural networks.

In the past lectures, we saw numerous phenomena of overparameterized (ridgeless) linear regression.

- When $d > n$, ridgeless linear regression **interpolates** the training data—the signal as well as the noise.
- Though the least squares optimization problem is ill-posed in this overparameterized regime, the implicit bias of (stochastic) gradient descent magically chooses the solution with the **smallest function norm** (which we argued should be good for generalization).
- Using random matrix theory and a high-dimensional asymptotic analysis of the linear regression risk, we showed a phenomenon of **implicit regularization** that increases as the ratio d/n increases, explaining the **double descent** curve.
- The generalization depends on the spectrum of the data covariance $\boldsymbol{\Sigma} = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]$. Only the most rapidly-decaying spectra produce interpolators with unbounded risk—the boogeyman from classical statistics. Power-law spectra yield interpolators that will be a multiplicative factor away from the Bayes risk, and special slow-decaying spectra will produce *consistent interpolators* that **benignly overfit** the data.

How will we relate these facts to neural networks? Many of the phenomena we analyzed in high-dimensional linear regression occur empirically for neural networks as well, suggesting there may be some overlap in the theory. We will explore this connection over the next four lectures.

1. In this lecture, we will explore different ways to approximate neural networks with linear models. As these models increase in size, the corresponding linear model approximations will approach infinite dimensionality, and so we leverage our understanding of RKHS to analyze these models in closed form.
2. The next lecture will explore how good these approximations are in practice. We will show that, when the parameters of a neural network are close to a random initialization, this approximation is essentially exact.
3. Optimization will be the focus of the third lecture, with one of the most surprising results of this course. Despite the non-convexity of the neural network optimization problem, we will show that large neural networks will remain arbitrarily close to this linear approximation through the course of training; yielding a trained neural network that obtains the global minimum with a functional form that can be analyzed analytically.
4. In the fourth lecture (part 2 of "advanced topics"), we will explore limitations of explaining neural networks with linear approximations.

---

1) Linear Approximation 1: The Random Feature Model

Recall the function form for a 1-hidden layer neural network $f(\cdot) : \mathbb{R}^p \to \mathbb{R}$ with d hidden features:

$$f(\boldsymbol{x};\boldsymbol{\theta}) = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\beta}, \qquad \boldsymbol{\phi}(\boldsymbol{x})^\top = \frac{1}{\sqrt{d}} \left[\sigma(\boldsymbol{w}_1^\top \boldsymbol{x}) \quad \cdots \quad \sigma(\boldsymbol{w}_D^\top \boldsymbol{x}) \right], \qquad \boldsymbol{\theta} = \Big[\underbrace{\boldsymbol{\beta}}_{\in \mathbb{R}^d} \quad \underbrace{\boldsymbol{w}_1}_{\in \mathbb{R}^p} \quad \cdots \quad \underbrace{\boldsymbol{w}_d}_{\in \mathbb{R}^p} \Big]. \quad (1)$$

$\boldsymbol{\beta}, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_d$ are the learnable parameters of the neural network. $\sigma(\cdot) : \mathbb{R} \to \mathbb{R}$ is the non-linear activation function, which is typically the ReLU i.e. $\sigma(z) = \max(0, z)$. Note that we have added a factor of $1/\sqrt{d}$ to the $\boldsymbol{\phi}(\boldsymbol{x})$ function; we will see it's importance very soon! Intuitively it decreases the relative importance of each hidden feature as d increases.

What would happen if we chose to only learn the $\boldsymbol{\beta}$ parameters, fixing the $\boldsymbol{w}_i$ parameters to some randomly initialized values? Then the $\boldsymbol{\phi}(\boldsymbol{x})$ function in Eq. (1) is fixed function of $\boldsymbol{x}$, and so $f(\boldsymbol{x}; \boldsymbol{\theta})$ is linear with respect to its learnable parameters $\boldsymbol{\beta}$!

This model is a crude approximation of a neural network, and so we will soon develop a more sophisticated linear approximation. Nevertheless, it's worthwhile analyzing its properties in a bit of detail.

1.1 A Kernel Machine as $d \to \infty$

Before we begin training this neural network, we initialize all of its parameters randomly. Typically the $\boldsymbol{\beta}$ and the $\boldsymbol{w}_i$ parameters are drawn from a standard normal distribution $\mathcal{N}(0, 1)$.[1] The $\boldsymbol{\beta}$ parameters will of course move from this random initialization to their global optimum during training, but the $\boldsymbol{w}_i$ parameters will remain fixed at their random initialization.

Using our understanding of the space of linear models as RKHS, we can also say that the neural network $f(\boldsymbol{x}; \boldsymbol{\theta})$ with fixed hidden layer is a function from the RKHS defined by the kernel $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{x}')$. This kernel is not particularly useful since it depends on the random initialization of the $\boldsymbol{w}_i$ parameters. But, by the law of large numbers, this kernel becomes deterministic as $d \to \infty$:

$$\lim_{d\to\infty} k(\boldsymbol{x}, \boldsymbol{x}') = \lim_{d\to\infty} \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{x}') = \lim_{d\to\infty} \frac{1}{d} \sum_{i=1}^{d} \sigma(\boldsymbol{w}_i^\top \boldsymbol{x}) \sigma(\boldsymbol{w}_i^\top \boldsymbol{x}') \overset{\text{a.s.}}{=} \mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(0,1)} \left[\sigma(\boldsymbol{w}^\top \boldsymbol{x}) \sigma(\boldsymbol{w}^\top \boldsymbol{x}') \right].$$

Moreover, if $\sigma(z) = \max(0, z)$, we can derive this limiting kernel in closed form! The following is a simplified derivation of [Cho and Saul, 2009]:

Theorem 1.

$$\lim_{d\to\infty} k(\boldsymbol{x}, \boldsymbol{x}') \overset{\text{a.s.}}{=} \frac{1}{2\pi} \|\boldsymbol{x}\| \, \|\boldsymbol{x}'\| \left[\sin(\varphi) + (\pi - \varphi)\cos(\varphi) \right], \qquad \varphi = \cos^{-1}\left(\frac{\boldsymbol{x}^\top \boldsymbol{x}'}{\|\boldsymbol{x}\| \, \|\boldsymbol{x}'\|} \right). \tag{2}$$

Proof. We begin by noting that $\sigma(\alpha z) = \max(0, \alpha z) = \alpha \sigma(z)$ for $\alpha \geq 0$. In mathematical parlance, we say that σ is a **non-negative homogeneous function**. Thus, we have that

$$\lim_{d\to\infty} k(\boldsymbol{x}, \boldsymbol{x}') \quad \overset{\text{a.s.}}{=} \quad \mathbb{E}[\sigma(\boldsymbol{w}^\top \boldsymbol{x}) \sigma(\boldsymbol{w}^\top \boldsymbol{x})] \quad = \quad \|\boldsymbol{x}\| \, \|\boldsymbol{x}'\| \, \mathbb{E}\left[\sigma\left(\boldsymbol{w}^\top \tfrac{\boldsymbol{x}}{\|\boldsymbol{x}\|} \right) \sigma\left(\boldsymbol{w}^\top \tfrac{\boldsymbol{x}'}{\|\boldsymbol{x}'\|} \right) \right]. \tag{3}$$

We will therefore assume that $\boldsymbol{x}$ and $\boldsymbol{x}'$ are unit vectors without loss of generality, recognizing that the norms of these vectors will factor outside as multiplicative factors applied to the kernel. Next, by linear Gaussian identities, for any two unit vectors $\boldsymbol{x}, \boldsymbol{x}'$ and any $\boldsymbol{w} \sim \mathcal{N}(0, 1)$, $\boldsymbol{x}^\top \boldsymbol{w}$ and $\boldsymbol{x}'^\top \boldsymbol{w}$ are jointly Gaussian with

$$\mathbb{E}[\boldsymbol{x}^\top \boldsymbol{w}] = \boldsymbol{x}^\top \mathbb{E}[\boldsymbol{w}] = 0, \qquad \mathbb{E}[\boldsymbol{x}^\top \boldsymbol{w}\boldsymbol{w}^\top \boldsymbol{x}] = \boldsymbol{x}^\top \mathbb{E}[\boldsymbol{w}\boldsymbol{w}^\top] \boldsymbol{x} = 1,$$
$$\mathbb{E}[\boldsymbol{x}^\top \boldsymbol{w}\boldsymbol{w}^\top \boldsymbol{x}'] = \boldsymbol{x}^\top \mathbb{E}[\boldsymbol{w}\boldsymbol{w}^\top] \boldsymbol{x}' = \boldsymbol{x}^\top \boldsymbol{x}' = \cos(\varphi),$$

[1]Sometimes this distribution is scaled with d; however the $1/\sqrt{d}$ factor in $\boldsymbol{\phi}(\boldsymbol{x})$ accomplishes this scaling for us.

where the first two rules also apply to $\boldsymbol{x}'$. Thus we have that

$$\begin{bmatrix} \boldsymbol{x}^\top \boldsymbol{w} \\ \boldsymbol{x}'^\top \boldsymbol{w} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \underbrace{\begin{bmatrix} 1 & \cos(\varphi) \\ \cos(\varphi) & 1 \end{bmatrix}}_{:=\boldsymbol{A}} \right).$$

Defining $z := \boldsymbol{x}^\top \boldsymbol{w}$ and $z' := \boldsymbol{x}'^\top \boldsymbol{w}$, we can write the expectation in Eq. (3) as

$$\mathbb{E}\left[\sigma\left(\boldsymbol{w}^\top \boldsymbol{x}\right)\sigma\left(\boldsymbol{w}^\top \boldsymbol{x}'\right)\right] = \tau(\varphi), \qquad \begin{aligned} \tau(\varphi) &= \int\int \sigma(z)\sigma(z')\mathcal{N}\left(\begin{bmatrix} z \\ z' \end{bmatrix}; \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \boldsymbol{A}\right) dz\, dz' \\ &= \frac{1}{2\pi|\boldsymbol{A}|^{1/2}} \int_0^\infty \int_0^\infty zz' \exp\left(-\frac{1}{2}\begin{bmatrix} z & z' \end{bmatrix} \boldsymbol{A}^{-1} \begin{bmatrix} z \\ z' \end{bmatrix}\right) dz\, dz'. \end{aligned}$$

Recognizing that

$$\boldsymbol{A}^{-1} = \frac{1}{\sin^2(\varphi)} \begin{bmatrix} 1 & -\cos(\varphi) \\ -\cos(\varphi) & 1 \end{bmatrix}, \qquad |\boldsymbol{A}| = 1 - \cos^2(\varphi) = \sin^2(\varphi),$$

converting to polar coordinates $z = r\cos(\psi)$ and $z' = r\sin(\psi)$, and plugging in trigonometric identities completes the proof. (See [Cho and Saul, 2009, Appx. A] for a details.) □

We refer to the limiting kernel in Eq. (2) as the **arc-cosine kernel**.

1.2 Universal Approximation

Based on Eq. (2), we see that infinite-width neural networks with randomly-initialized hidden layer weights are (almost surely) universal approximators.

Theorem 2. *Let $\mathcal{H}$ be the RKHS defined by the limiting kernel in Eq. (2). For any compact $\mathcal{X} \subset \mathbb{R}^d$ and any continuous function $\ell : \mathcal{X} \to \mathbb{R}$, there exists some $h \in \mathcal{H}$ with $\sup_{\boldsymbol{x}\in\mathcal{X}} |\ell(\boldsymbol{x}) - h(\boldsymbol{x})| < \epsilon$ for any $\epsilon > 0$.*

The proof requires a bit of analysis, so here's a proof sketch based on Micchelli et al. [2006, Corr. 8].

Proof sketch. Assume that $\mathcal{X}$ is the shell of the unit sphere $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| = 1\}$. Then the limiting kernel in Eq. (2) is a function of $\boldsymbol{x}^\top \boldsymbol{x}' = \cos(\varphi)$, i.e.

$$\lim_{d\to\infty} k(\boldsymbol{x}, \boldsymbol{x}') \overset{\text{a.s.}}{=} g(\boldsymbol{x}^\top \boldsymbol{x}'), \qquad g(\cos(\varphi)) = \frac{1}{2\pi}\Big[\sin(\varphi) + (\pi - \varphi)\cos(\varphi)\Big].$$

Note that $g : [-1, 1] \to \mathbb{R}$ admits a Taylor expansion with infinitely many non-zero coefficients.

The RKHS $\mathcal{H}$ is defined as the (completion of the) space of functions

$$\mathcal{H} := \left\{ h(\boldsymbol{x}) = \sum_{i=1}^n \alpha_i g(\boldsymbol{x}^\top \boldsymbol{x}_i) \;:\; n \in \mathbb{N}, \boldsymbol{x}_i \in \mathcal{X} \right\}.$$

Each term inside the summation of the sum thus admits an infinite polynomial expansion. By choosing α_i and $\boldsymbol{x}_i$ appropriately, we can model any analytic infinite polynomial and thus, by the Stone-Weierstrass theorem, we can model any ℓ that is continuous on $\mathcal{X}$ up to any arbitrary precision.

We can repeat this process for shells of various radii, and "glue" them together to approximate any continuous function on any compact $\mathcal{X} \subset \mathbb{R}^d$. □

1.3 Observations About the Arc-Cosine Kernel

With Eq. (2) in hand, we can make the following observations:

1. For any $\boldsymbol{x}, \boldsymbol{x}'$ with norm 1, the arc-cosine kernel is a function of $\boldsymbol{x}^\top \boldsymbol{x}' = \cos(\varphi)$. The linear kernel $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^\top \boldsymbol{x}'$ also is a function of $\boldsymbol{x}^\top \boldsymbol{x}'$. However, the arc-cosine RKHS can learn any arbitrary continuous function while the linear RKSH can only learn linear functions.

2. For most data distributions, the spectrum of the kernel in Eq. (2) will decay at a rate of $\lambda_i = i^{-\alpha}$ for some $\alpha > 1$ [Geifman et al., 2020].

3. For neural networks with multiple hidden layers, each of which is infinitely wide[2] and with fixed random initialization, we can similarly compute the limiting kernel of a neural network. The formula for this limiting kernel is a recursive application of Eq. (2) [Lee et al., 2018], and the kernel function is depicted in Fig. 1.

4. The RKHS associated with randomly-initialized infinite-width deep neural networks are also universal approximators.

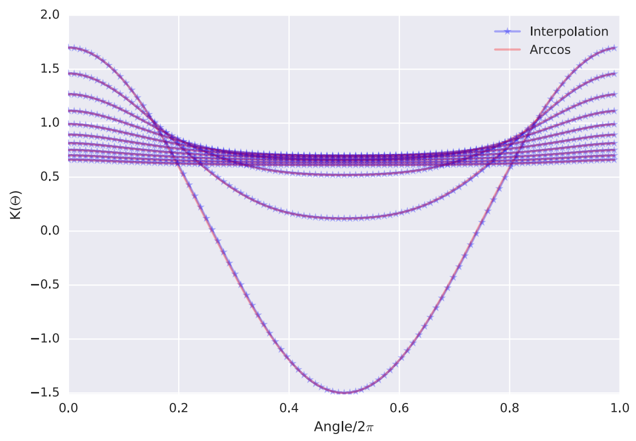

Figure 1: The limiting kernel of randomly initialized infinite-width neural networks with ReLU activations, as a function of the angle between two vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$. Each line represents a network of a different depth. Deeper neural networks have "flatter" limiting kernels. The line with the most variation represents a 1-hidden layer neural network.

However, we also note the following limitations:

1. This approximation disregards the learning of the hidden layer parameters.

2. The capacity of a neural network under this approximation depends on its width but not its depth. A neural network with L hidden layers of width d has only d learnable parameters, so width would have to match n before the network starts interpolating data. In practice, we would expect that depth has some impact on capacity.

In the next lecture we will look at a more sophisticated linear approximation of neural networks.

[2]For simplicity, we assume that we take each hidden layer to infinite-width sequentially rather than simultaneously.

References

Y. Cho and L. K. Saul. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems*, 2009.

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.

J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018.

C. A. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(95):2651–2667, 2006.