Lecture 03: Implicit Bias of (Stochastic) Gradient Descent, Double Descent

Geoff Pleiss

Back in the first lecture, we discussed the "classical" risk bound for an ERM-learned model $\hat{f} \in \mathcal{H}$:

$$\mathcal{R}_{\text{emp}}(\hat{f}) - \mathcal{R}(\hat{f}) \leq \tilde{O}\left(\sqrt{\text{cap}(\mathcal{H})/n}\right), \qquad \mathcal{R}_{\text{emp}}(\hat{f}) - \min_{f \in \mathcal{H}} \mathcal{R}(f) \leq \tilde{O}\left(\sqrt{\text{cap}(\mathcal{H})/n}\right),$$

where $\text{cap}(\mathcal{H})$ is some notion of capacity of the hypothesis space $\mathcal{H}$. If $\mathcal{H}$ is too restrictive of a hypothesis class, then $\min_{f \in \mathcal{H}} \mathcal{R}(f)$ will be too small; conversely if $\mathcal{H}$ is too expressive, then $\text{cap}(\mathcal{H})$ and thus the gap between the empirical and true risk will be too large. From the perspective of the bias-variance tradeoff, the former case is indicative of high bias (underfitting) and the latter is indicative of high variance (overfitting). Both scenarios result in poor generalization, as depicted in this toy diagram below:

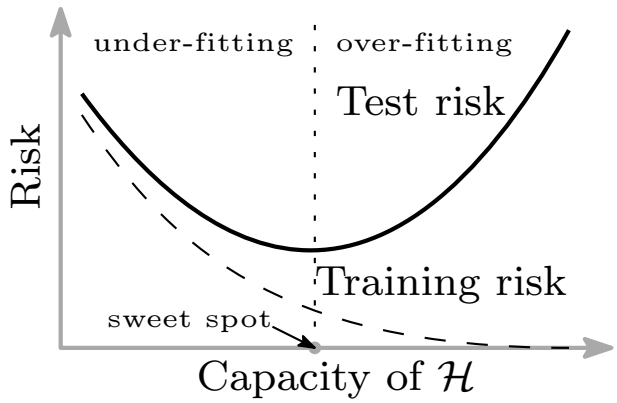

Figure 1: The classical bias-variance tradeoff. Reproduced from [Belkin et al., 2019].

If we consider neural networks with d parameters, increasing p will increase the capacity of the hypothesis class (as measured by any standard notion). We would thus expect large neural networks, which have many more parameters p than training data points n, to be in the extremely high variance regime where we are overfitting like crazy. Instead, something weird happens as we vary p:

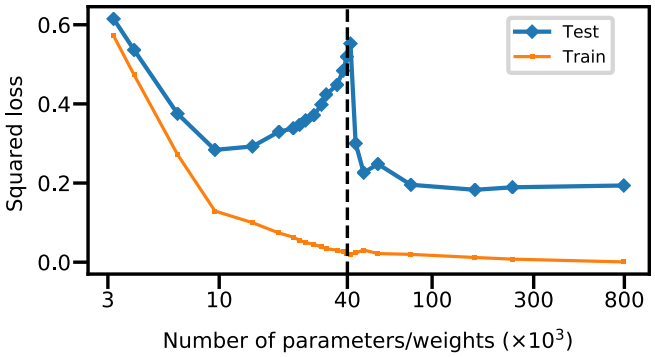

Figure 2: Training and test loss for a 2-layer neural network trained on MNIST as a function of p (the number of learnable parameters). Reproduced from [Belkin et al., 2019].

On the left side of the dashed line, the training and test error seem to follow the classical bias-variance tradeoff we see in Fig. 1. However, passed the dashed line the test error decreases as we increase p! It

appears that we can increase p to infinity and somehow we would still get better generalization than what we would have achieved on the left side of the dashed line. Moreover, we are obtaining this decrease in test error even though we're in the regime where the model (nearly) **interpolates** the training data—i.e. it has (nearly) zero training error.

This **double descent** curve, first characterized by Belkin et al. [2019], is not unique to neural networks. We can reproduce it in linear regression.

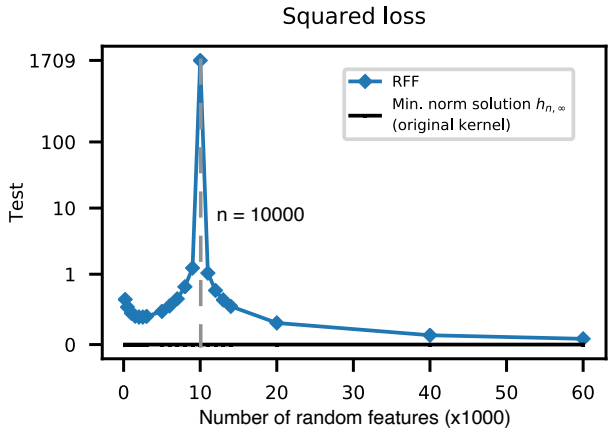

Figure 3: Training and test loss for a linear regression model trained on a basis expansion of some data as a function of p (the number of basis functions). These linear regressors are trained on a $n = 10{,}000$ set of training data. Reproduced from [Belkin et al., 2019].

Fig. 3 depicts the exact same test/train error curves for a linear regression model trained without any regularization. We are regressing on a basis expansion of some data, and as we move along the x axis we add more basis functions that represent our data. The number of parameters in p—equal to the number of basis functions we have in our representation. Moreover, the dashed-line inflection point perfectly corresponds with the point where $n = p$.

Why does this double descent phenomenon occur? Beyond the dashed line we have more than enough capacity to perfectly fit the training data, and—without regularization—we are not specifying which of the infinitely many interpolating models we should choose. The answer is subtle but profound. Even though the capacity of the neural network/linear regressor grows as p increases, the optimization algorithm we use happens to concentrate us on a magical solution with good generalization properties.

---

1) Overparameterized Ridgeless Linear Regression Has Infinitely Many Solutions

Consider ridge regression in the $p > n$ setting (where we ignore the $1/n$ factor in the loss):

$$\begin{aligned}&\min_{\boldsymbol{\theta}} \frac{1}{2}\sum_{i=1}^{n}\left(\boldsymbol{x}_i^\top\boldsymbol{\theta} - y_i\right)^2 + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2 \\ =&\min_{\boldsymbol{\theta}} \frac{1}{2}\|\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2\end{aligned} \qquad \boldsymbol{X} = \begin{bmatrix}\boldsymbol{x}_1^\top \\ \vdots \\ \boldsymbol{x}_n^\top\end{bmatrix} \in \mathbb{R}^{n\times p}, \quad \boldsymbol{y} = \begin{bmatrix}y_1 \\ \vdots \\ y_n\end{bmatrix} \in \mathbb{R}^n.$$

This optimization problem is strictly convex and so has a unique global minimum that can be found with (stochastic) gradient descent. It also happens to have a closed form solution, which can be found by setting

the gradient to zero and solving analytically:

$$\begin{aligned}\boldsymbol{\theta}_\lambda^* &= \left(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y} \\ &= \boldsymbol{X}^\top \left(\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{y}. \end{aligned} \tag{1}$$

What now happens in the ridgeless case, where instead we are just solving $\min_{\boldsymbol{\theta}} \frac{1}{2}\|\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}\|_2^2$? If $p > n$ and $\boldsymbol{X}$ is full rank, there are now infinitely many $\boldsymbol{\theta}$ that perfectly fit the data, achieving the minimum loss of zero. This optimization problem is still convex, but it is no longer strictly convex. The solution that we arrive at will depend on the *optimization algorithm*.

---

2) The Minimum Norm Solution

Imagine we have an infinitesimal amount of regularization. By Eq. (1), the (nearly-)ridgeless regressor will arrive at the solution

$$\boldsymbol{\theta}_0^* := \lim_{\lambda \to 0} \boldsymbol{\theta}_\lambda^* = \boldsymbol{X}^\top \left(\boldsymbol{X}\boldsymbol{X}^\top\right)^{-1} \boldsymbol{y}.$$

It can easily be seen that this solution (nearly) interpolates the data. There is something very special about this solution: it is the **minimum norm solution** that interpolates the data. That is:

$$\boldsymbol{\theta}_0^* = \underset{\boldsymbol{X}\boldsymbol{\theta} = \boldsymbol{y}}{\arg\min} \|\boldsymbol{\theta}\|_2^2. \tag{2}$$

Using the functional perspective we developed last lecture, if $\mathcal{H}$ is the RKHS of linear functions of $\boldsymbol{x}$, then

$$f^*(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\theta}_0^* = \underset{f \in \mathbb{I}}{\arg\min} \|f\|_{\mathcal{H}}^2, \qquad \mathbb{I} := \{f \in \mathcal{H} : f(\boldsymbol{x}_i) = y_i \ \forall i \in [1, n]\}. \tag{3}$$

While we could prove the minimum norm nature of $\boldsymbol{\theta}_0^*$ from the Eq. (2) perspective, we will prove it from the Eq. (3) perspective to generalize to arbitrary RKHS.

Proof. Let $\mathcal{H}$ be some RKHS with associated kernel $k(\boldsymbol{x}, \boldsymbol{x}')$. Given the training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, we will split $\mathcal{H}$ into two orthogonal subspaces $\mathcal{H}_n$ and $\mathcal{H}_\perp$:

$$\mathcal{H}_n := \operatorname{span}\{k(\boldsymbol{x}_i, \cdot)\}_{i=1}^n, \qquad \mathcal{H}_\perp := \{g \in \mathcal{H} \ : \ \langle f_n, g\rangle_{\mathcal{H}} = 0 \ \ \forall f_n \in \mathcal{H}_n\}.$$

Any function $f \in \mathcal{H}$ can be decomposed as

$$f(\cdot) = \underbrace{f_n(\cdot)}_{\in \mathcal{H}_n} + \underbrace{f_\perp(\cdot)}_{\in \mathcal{H}_\perp}.$$

Note that

$$\|f\|_{\mathcal{H}}^2 = \langle f_n + g, f_n + g\rangle_{\mathcal{H}} = \|f_n\|_{\mathcal{H}}^2 + \|g\|_{\mathcal{H}}^2 + \cancelto{0}{2\langle f_n, g\rangle_{\mathcal{H}}}$$

If $f \in \mathbb{I}$ (as defined in Eq. (3)), then $f(\boldsymbol{x}_i) = f_n(\boldsymbol{x}_i) + g(\boldsymbol{x}_i) = y_i$ for all i. However, by the reproducing property of $\mathcal{H}$ and the definition of $\mathcal{H}_\perp$:

$$g(\boldsymbol{x}_i) = \langle g(\cdot), k(\boldsymbol{x}_i, \cdot)\rangle_{\mathcal{H}} = 0. \qquad (k(\boldsymbol{x}_i, \cdot) \in \mathcal{H}_n, \text{ definition of } \mathcal{H}_\perp)$$

Therefore, $f_n(\cdot) \in \mathbb{I}$. Since $f_n(\cdot) = \sum_{i=1}^n \alpha_i k(\boldsymbol{x}_i, \cdot)$ for some $\alpha_i \in \mathbb{R}$, we have that

$$\underbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}}_{\boldsymbol{y}} = \begin{bmatrix} f_n(\boldsymbol{x}_1) \\ \vdots \\ f_n(\boldsymbol{x}_n) \end{bmatrix} = \underbrace{\begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_n, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{bmatrix}}_{:=\boldsymbol{K}} \underbrace{\begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}}_{:=\boldsymbol{\alpha}},$$

which implies that $\boldsymbol{\alpha} = \boldsymbol{K}^{-1}\boldsymbol{y}$ parameterizes the unique interpolating function in $\mathcal{H}_n$.

Thus, the set of interpolants $\mathbb{I}$ is given by

$$\mathbb{I} = \left\{ \underbrace{f_n^*(\cdot)}_{\in \mathcal{H}_n} + \underbrace{g(\cdot)}_{\in \mathcal{H}_\perp} \;:\; f_n^*(\cdot) = \sum_{i=1}^n \alpha_i k(\boldsymbol{x}_i, \cdot) \right\}.$$

And since $\|f\|_{\mathcal{H}}^2 = \|f_n^*\|_{\mathcal{H}}^2 + \|g\|_{\mathcal{H}}^2$, for all $f \in \mathbb{I}$, we obtain the minimum norm solution by setting $g = 0$.

Finally, note that in the linear case where $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^\top \boldsymbol{x}'$, we have:

$$f_n^*(\boldsymbol{x}) = \begin{bmatrix} k(\boldsymbol{x}, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}, \boldsymbol{x}_n) \end{bmatrix} \boldsymbol{K}^{-1}\boldsymbol{y} = \boldsymbol{x}\left(\boldsymbol{X}^\top \left(\boldsymbol{X}\boldsymbol{X}^\top\right)^{-1} \boldsymbol{y}\right) = \boldsymbol{x}^\top \boldsymbol{\theta}_0^*.$$

□

Why is a minimum norm solution desirable? If we think of $\|f\|_{\mathcal{H}}$ as a measure of the complexity of f, then the minimum norm interpolator is the "least complex" function that interpolates the data [Belkin et al., 2018].

Put a different way, imagine we had two training sets: $\mathcal{D}_1$ and $\mathcal{D}_2$. Ideally we would not want the functions learned from these two training sets to be too different (i.e. we want low variance). Let $\mathbb{I}_1 \subset \mathcal{H}$ and $\mathbb{I}_2 \subset \mathcal{H}$ be the sets of interpolating functions for each training set. By the previous proof, we know that both subsets are affine subspaces of $\mathcal{H}$. Changing the data will change the "angle" of the subspace. If we are far away from the origin, even a slight change in "angle" could produce very different functions. However, the minimum norm solutions in $\mathbb{I}_1$ and $\mathbb{I}_2$ are "close" to the origin, and so the "slight difference in angle" between $\mathbb{I}_1$ and $\mathbb{I}_2$ will not produce very different functions. See Fig. 4 for an illustration.

---

3) The Implicit Bias of Gradient Descent

Magically, gradient descent will generally find the minimum norm solution for overparameterized linear regression. It's worth reflecting on this fact for a moment. We have a problem with infinitely many global optima, and gradient descent happens to choose the one that is "simplest" [Zhang et al., 2017].

Theorem 1. *Consider the ridgeless linear regression problem* $\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \frac{1}{2}\|\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}\|_2^2$ *with* $p > n$. *If we initialize optimization from any* $\boldsymbol{\theta}^{(0)} \in \text{collspace}\{\boldsymbol{X}^\top\}$, *and run gradient descent with step size* $\gamma < \sigma_{\max}(\boldsymbol{X})^2$ *(where* $\sigma_{\max}(\boldsymbol{X})$ *is the largest singular value of* $\boldsymbol{X}$*), then the limit of the iterates is the minimum norm solution* $\boldsymbol{\theta}_0^*$.

Proof. Let $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top = \boldsymbol{X}$ be the singular value decomposition of $\boldsymbol{X}$, with $\boldsymbol{U}, \boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{V} \in \mathbb{R}^{p \times n}$.

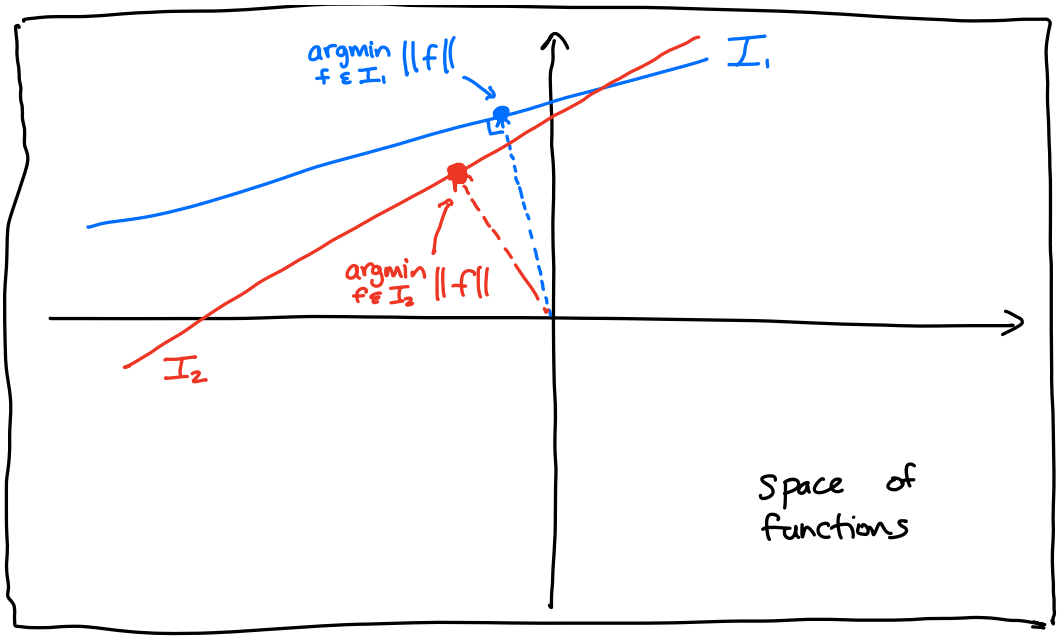

Figure 4: The sets of interpolating functions $\mathbb{I}_1$ and $\mathbb{I}_2$ for two different training datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ are affine subspaces of $\mathcal{H}$. The minimum norm solutions $\boldsymbol{\theta}_0^1$ and $\boldsymbol{\theta}_0^2$ are "close" to the origin, and so the "slight difference in angle" between $\mathbb{I}_1$ and $\mathbb{I}_2$ will not produce very different functions.

Since $\boldsymbol{\theta}^{(0)} \in \text{collspace}\{\boldsymbol{X}^\top\}$, we have $\boldsymbol{\theta}_0 = \boldsymbol{V}\boldsymbol{\beta}_0$ for some $\boldsymbol{\beta}_0 \in \mathbb{R}^n$. Now assume that it also holds that $\boldsymbol{\theta}^{(i)} = \boldsymbol{V}\boldsymbol{\beta}^{(i)}$ for some $\boldsymbol{\beta}^{(i)} \in \mathbb{R}^n$. Gradient descent performs the iteration

$$\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} - \gamma \nabla \mathcal{L}(\boldsymbol{\theta}^{(i)}) = \underbrace{\boldsymbol{\theta}^{(i)}}_{\boldsymbol{V}\boldsymbol{\beta}^{(i)}} - \underbrace{\gamma \boldsymbol{X}^\top \left(\boldsymbol{X}\boldsymbol{\theta}^{(i)} - \boldsymbol{y}\right)}_{\boldsymbol{V}\left(\gamma \boldsymbol{\Sigma}\boldsymbol{U}^\top\right)\left(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \boldsymbol{V}\boldsymbol{\beta}^{(i)} - \boldsymbol{y}\right)},$$

and so $\boldsymbol{\theta}^{(i+1)}$ can also be written as $\boldsymbol{V}\boldsymbol{\beta}^{(i+1)}$ for some $\boldsymbol{\beta}^{(i+1)} \in \mathbb{R}^n$. We can thus rewrite the update rule in the projection onto $\text{collspace}\{\boldsymbol{X}^\top\}$:

$$\boldsymbol{\beta}^{(i+1)} = \boldsymbol{\beta}^{(i)} - \gamma \boldsymbol{\Sigma}^2 \boldsymbol{\beta}^{(i)} + \gamma \boldsymbol{U}\boldsymbol{y} = \left(\boldsymbol{I} - \gamma \boldsymbol{\Sigma}^2\right)\boldsymbol{\beta}^{(i)} + \gamma \boldsymbol{\Sigma}\boldsymbol{U}\boldsymbol{y}, \qquad \boldsymbol{\theta}^{(i+1)} = \boldsymbol{V}\boldsymbol{\beta}^{(i+1)}.$$

A simple induction will show that this recursion admits the closed form:

$$\boldsymbol{\beta}^{(i)} = \left(\boldsymbol{I} - \gamma \boldsymbol{\Sigma}^2\right)^i \boldsymbol{\beta}^{(0)} + \sum_{j=1}^{i-1} \left(\boldsymbol{I} - \gamma \boldsymbol{\Sigma}^2\right)^j \gamma \boldsymbol{\Sigma}\boldsymbol{U}\boldsymbol{y}.$$

Note that—by the assumption on γ—the matrix $\boldsymbol{I} - \gamma\boldsymbol{\Sigma}^2$ is positive definite with eigenvalues strictly less than 1. As $i \to \infty$, the matrix $\left(\boldsymbol{I} - \gamma\boldsymbol{\Sigma}^2\right)^i$ thus converges to zero and the summation becomes the Taylor series for a matrix inverse:

$$\lim_{i\to\infty} \sum_{j=1}^{i-1} \left(\boldsymbol{I} - \gamma \boldsymbol{\Sigma}^2\right)^j = \left(\boldsymbol{I} - \left(\boldsymbol{I} - \gamma \boldsymbol{\Sigma}^2\right)\right)^{-1} = \frac{1}{\gamma}\boldsymbol{\Sigma}^{-2}$$

Thus, $\lim_{i\to\infty} \boldsymbol{\beta}^{(i)} = \boldsymbol{\Sigma}^{-1}\boldsymbol{U}\boldsymbol{y}$, and so

$$\lim_{i\to\infty} \boldsymbol{\theta}^{(i)} = \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}\boldsymbol{y} = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^\top \boldsymbol{U}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{U}\boldsymbol{y} = \boldsymbol{X}^\top \left(\boldsymbol{X}\boldsymbol{X}^\top\right)^{-1} \boldsymbol{y} = \boldsymbol{\theta}_0^*.$$

□

Note that stochastic gradient descent also converges to the minimum norm solution, even if gradient descent is initialized outsized the column space of $\boldsymbol{X}$. However the proof is much more complicated. We refer to this minimum-norm-seeking behaviour as the **implicit bias** of (stochastic) gradient descent.

---

4) Putting it All Together

We now have a clearer picture of what's going on with double descent, at least in the case of ridgeless linear regression. Let's re-examine Fig. 3. The left side of the dotted line represents the **underparameterized** regime where $p < n$ and linear regression does not interpolate data. In this regime the test error follows a classical bias-variance tradeoff: increasing the number of parameters decreases bias but increases variance, resulting in an initial decline in test error followed by an increase as $p \to n$.

Now consider the point on the right where $p = 60{,}000$. Let $\mathcal{H}_{60,000}$ be the space of linear functions over these $p = 60{,}000$. All of the linear models represented by this plot are members of $\mathcal{H}_{60,000}$. (For example, the regressor with $p = 100$ features could be written as a function in $\mathcal{H}_{60,000}$ where 59,900 of the coefficients are zero.) All of the models on the right side of the dotted line $p > n$ are thus interpolating functions in $\mathcal{H}_{60,000}$. However, the rightmost model is—by definition—the minimum norm interpolator in $\mathcal{H}_{60,000}$, and so all other interpolating functions will have a larger norm, and thus more likely to be higher variance. (See Fig. 5 for an illustration.) In the next lecture we will try to formalize this intuition.

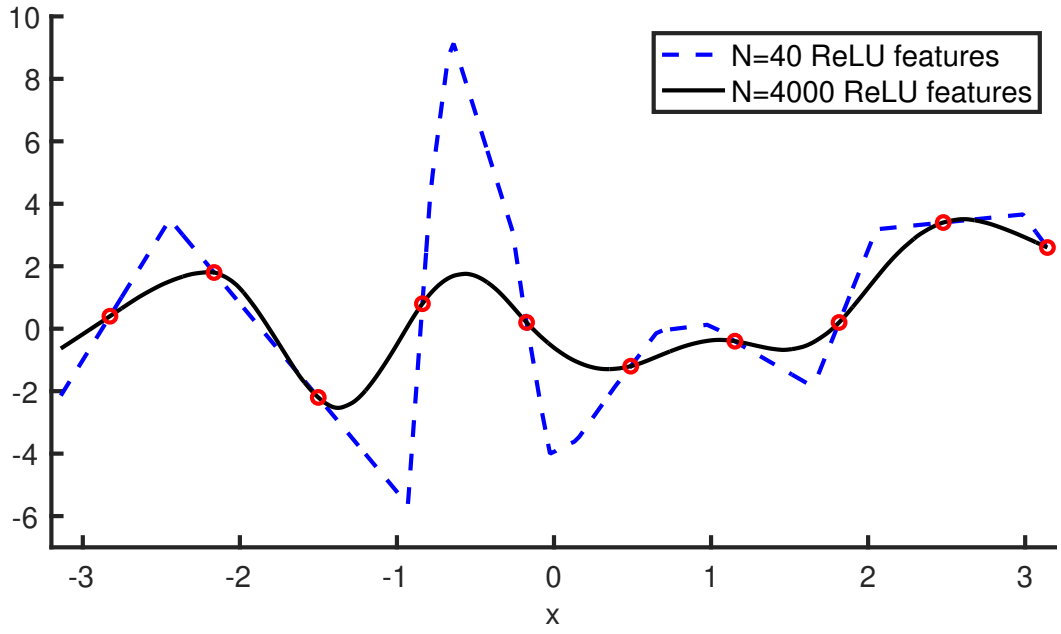

Figure 5: Two interpolating solutions that are linear functions of basis expansions of $\mathbb{R}$. The interpolator that uses 40 features is a much worse and less smooth fit than the one that uses 4000 features. Both live within the same RKHS, but the 4000 feature interpolator has a lower function norm. Figure reproduced from [Belkin et al., 2019].

It is worth emphasizing that this behaviour hinges on the fact that we use gradient descent to optimize our models. If we were instead to use some other optimization algorithm that was not implicitly biased towards minimum norm solutions, we instead might choose very poor interpolating functions, and thus we'd be more likely to see the classical bias-variance tradeoff as $p \to \infty$. In other words, the implicit bias of gradient descent allows us to achieve good generalization in the overparameterized regime.

There is still one open question: though larger p is better once we are in the overparamterized/interpolation regime, why does the overparamterized regime sometimes produce lower test error than the underparamterized regime? Can we characterize when overparameterization is likely to be beneficial? We will address this question in a future lecture.

References

M. Belkin, S. Ma, and S. Mandal. To understand deep learning we need to understand kernel learning. In *International Conference on Machine Learning*, pages 541–549. PMLR, 2018.

M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.