Spectrum Dependent Learning Curves in Kernel Regression and Wide Neural Networks

Blake Bordelon [1] **Abdulkadir Canatar** [2] **Cengiz Pehlevan** [1 3]

Abstract

We derive analytical expressions for the generalization performance of kernel regression as a function of the number of training samples using theoretical methods from Gaussian processes and statistical physics. Our expressions apply to wide neural networks due to an equivalence between training them and kernel regression with the Neural Tangent Kernel (NTK). By computing the decomposition of the total generalization error due to different spectral components of the kernel, we identify a new spectral principle: as the size of the training set grows, kernel machines and neural networks fit successively higher spectral modes of the target function. When data are sampled from a uniform distribution on a high-dimensional hypersphere, dot product kernels, including NTK, exhibit learning stages where different frequency modes of the target function are learned. We verify our theory with simulations on synthetic data and MNIST dataset.

1. Introduction

Finding statistical patterns in data that generalize beyond a training set is a main goal of machine learning. Generalization performance depends on factors such as the number of training examples, the complexity of the learning task, and the nature of the learning machine. Identifying precisely how these factors impact the performance poses a theoretical challenge. Here, we present a theory of generalization in kernel machines (Schölkopf & Smola, 2001) and neural networks (LeCun et al., 2015) with wide hidden layers that addresses these questions.

The goal of our theory is not to provide worst case bounds on generalization performance in the sense of statistical learning theory (Vapnik, 1999), but to provide analytical expressions that explain the average or a typical performance in the spirit of statistical physics. The techniques we use are a continuous approximation to learning curves previously used in Gaussian processes (Sollich, 1999; 2002; Sollich & Halees, 2002) and the replica method of statistical physics (Sherrington & Kirkpatrick, 1975; Mézard et al., 1987).

We first develop an approximate theory of generalization in kernel regression that is applicable to any kernel. We then use our theory to gain insight into neural networks by using a correspondence between kernel regression and neural network training. When the hidden layers of a neural network are taken to infinite width with a certain initialization scheme, recent influential work (Jacot et al., 2018; Arora et al., 2019; Lee et al., 2019) showed that training a feedforward neural network with gradient descent to zero training loss is equivalent to kernel interpolation (or ridgeless kernel regression) with a kernel called the Neural Tangent Kernel (NTK) (Jacot et al., 2018). Our kernel regression theory contains kernel interpolation as a special limit (ridge parameter going to zero).

Our contributions and results are summarized below:

- Using a continuous approximation to learning curves adapted from Gaussian process literature (Sollich, 1999; 2002), we derive analytical expressions for learning curves for each spectral component of a target function learned through kernel regression.
- We present another way to arrive at the same analytical expressions using the replica method of statistical physics and a saddle-point approximation (Sherrington & Kirkpatrick, 1975; Mézard et al., 1987).
- Analysis of our theoretical expressions show that different spectral modes of a target function are learned with different rates. Modes corresponding to higher kernel eigenvalues are learned faster, in the sense that a marginal training data point causes a greater percent reduction in generalization error for higher eigenvalue modes than for lower eigenvalue modes.

[1]John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA [2]Department of Physics, Harvard University, Cambridge, MA, USA [3]Center for Brain Science, Harvard University, Cambridge, MA, USA. Correspondence to: Cengiz Pehlevan <cpehlevan@seas.harvard.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

Code: https://github.com/Pehlevan-Group/NTK_Learning_Curves

- When data is sampled from a uniform distribution on a hypersphere, dot product kernels, which include NTK, admit a degenerate Mercer decomposition in spherical harmonics, Y_{km}. In this case, our theory predicts that generalization error of lower frequency modes of the target function decrease more quickly than higher frequency modes as the dataset size grows. Different learning stages are visible in the sense described below.
- As the dimensions of data, d, go to infinity, learning curves exhibit different learning stages. For a training set of size $p \sim \mathcal{O}(d^l)$, modes with $k < l$ are perfectly learned, $k = l$ are being learned, and $k > l$ are not learned.
- We verify the predictions of our theory using numerical simulations for kernel regression and kernel interpolation with NTK, and wide and deep neural networks trained with gradient descent. Our theory fits experiments remarkably well on synthetic datasets and MNIST.

1.1. Related Work

Our main approximation technique comes from the literature on Gaussian processes, which is related to kernel regression in a certain limit. Total learning curves for Gaussian processes, but not their spectral decomposition as we do here, have been studied in a limited teacher-student setting where both student and teacher were described by the same Gaussian process and the same noise in (Opper & Vivarelli, 1998; Sollich, 1999). We allow arbitrary teacher distributions. Sollich also considered mismatched models where teacher and student kernels had different eigenspectra and different noise levels (Sollich, 2002). The total learning curve from this model is consistent with our results when the teacher noise is sent to zero, but we also consider, provide expressions for, and analyze generalization in spectral modes. We use an analogue of the "lower-continuous" approximation scheme introduced in (Sollich & Halees, 2002), the results of which we reproduce through the replica method (Mézard et al., 1987).

Generalization bounds for kernel ridge regression have been obtained in many contexts (Schölkopf & Smola, 2001; Cucker & Smale, 2002; Vapnik, 1999; Gyorfi et al., 2003), but the rates of convergence often crucially depend on the explicit ridge parameter λ and do not provide guarantees in the ridgeless case. Using a teacher-student setting, Spigler et al. (2019) showed that learning curves for kernel regression asymptotically decay with a power law determined by the decay rate of the teacher and the student. Such power law decays have been observed empirically on standard datasets (Hestness et al., 2017; Spigler et al., 2019). Recently, interest in explaining the phenomenon of interpolation has led to the study of generalization bounds on ridgeless regression (Belkin et al., 2018b;a; 2019b; Liang & Rakhlin, 2018). Here, our aim is to capture the average case performance of kernel regression, as opposed to a bound on it, that remains valid for the ridgeless case and finite sample sizes.

In statistical physics domain, Dietrich et al. (1999) calculated learning curves for support vector machines, but not kernel regression, in the limit of number of training samples going to infinity for dot product kernels with binary inputs using a replica method. Our theory applies to general kernels and finite size datasets. In the infinite training set limit, they observed several learning stages where each spectral mode is learned with a different rate. We observe similar phenomena in kernel regression. In a similar spirit, (Cohen et al., 2019) calculates learning curves for infinite-width neural networks using a path integral formulation and a replica analysis but does not discuss the spectral dependence of the generalization error.

In the infinite width limit, neural networks have many more parameters than training samples yet they do not overfit (Zhang et al., 2017). Some authors suggested that this is a consequence of the training procedure since stochastic gradient descent is implicitly biased towards choosing the simplest functions that interpolate the training data (Belkin et al., 2019a; 2018b; Xu et al., 2019a; Jacot et al., 2018). Other studies have shown that neural networks fit the low frequency components of the target before the high frequency components during training with gradient descent (Xu et al., 2019b; Rahaman et al., 2019; Zhang et al., 2019; Luo et al., 2019). In addition to training dynamics, recent works such as (Yang & Salman, 2019; Bietti & Mairal, 2019; Cao et al., 2019) have discussed how the spectrum of kernels impacts its smoothness and approximation properties. Here we explore similar ideas by explicitly calculating average case learning curves for kernel regression and studying its dependence on the kernel's eigenspectrum.

2. Kernel Regression Learning Curves

We start with a general theory of kernel regression. Implications of our theory for dot product kernels including NTK and trained neural networks are described in Section 3.

2.1. Notation and Problem Setup

We start by defining our notation and setting up our problem. Our initial goal is to derive a mathematical expression for generalization error in kernel regression, which we will analyze in the subsequent sections using techniques from the Gaussian process literature (Sollich, 1999; 2002; Sollich & Halees, 2002) and statistical physics (Sherrington & Kirkpatrick, 1975; Mézard et al., 1987).

The goal of kernel regression is to learn a function $f : \mathcal{X} \to \mathbb{R}^C$ from a finite number of observations (Wahba, 1990; Schölkopf & Smola, 2001). In developing our theory, we will first focus on the case where $C = 1$, and later extend

our results to $C > 1$ as we discuss in Section 2.5. Let $\{\mathbf{x}_i, y_i\} \in \mathcal{X} \times \mathbb{R}$, where $\mathcal{X} \subseteq \mathbb{R}^d$, be one of the p training examples and let $\mathcal{H}$ be a Reproducing Kernel Hilbert space (RKHS) with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. To avoid confusion with our notation for averaging, we will always decorate angular brackets for Hilbert inner product with $\mathcal{H}$ and a comma. Kernel ridge regression is defined as:

$$\min_{f \in \mathcal{H}} \sum_{i=1}^{p} (f(\mathbf{x}_i) - y_i)^2 + \lambda ||f||_{\mathcal{H}}^2. \tag{1}$$

The $\lambda \to 0$ limit is referred to as interpolating kernel regression, and, as we will discuss later, relevant to training wide neural networks. The unique minimum of the convex optimization problem is given by

$$f(\mathbf{x}) = \mathbf{y}^\top (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{k}(\mathbf{x}), \tag{2}$$

where $K(\cdot, \cdot)$ is the reproducing kernel for $\mathcal{H}$, $\mathbf{K}$ is the $p \times p$ kernel gram matrix $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, and $k(\mathbf{x})_i = K(\mathbf{x}, \mathbf{x}_i)$. Lastly, $\mathbf{y} \in \mathbb{R}^p$ is the vector of target values $y_i = f^*(\mathbf{x}_i)$. For interpolating kernel regression, when the kernel is invertible, the solution is the same except that $\lambda = 0$, meaning that training data is fit perfectly. The proof of this optimal solution is provided in the Supplementary Information (SI) Section 1.

Let $p(\mathbf{x})$ be the probability density function from which the input data are sampled. The generalization error is defined as the expected risk with expectation taken over new test points sampled from the same density $p(\mathbf{x})$. For a given dataset $\{\mathbf{x}_i\}$ and target function $f^*(\mathbf{x})$, let $f_K(\mathbf{x}; \{\mathbf{x}_i\}, f^*)$ represent the function learned with kernel regression. The generalization error for this dataset and target function is

$$E_g(\{\mathbf{x}_i\}, f^*) = \int d\mathbf{x}\, p(\mathbf{x}) \left(f_K(\mathbf{x}; \{\mathbf{x}_i\}, f^*) - f^*(\mathbf{x})\right)^2. \tag{3}$$

To calculate the *average case* performance of kernel regression, we average this generalization error over the possible datasets $\{\mathbf{x}_i\}$ and target functions f^*

$$E_g = \langle E_g(\{\mathbf{x}_i\}, f^*) \rangle_{\{\mathbf{x}_i\}, f^*}. \tag{4}$$

Our aim is to calculate E_g for a general kernel and a general distribution over teacher functions.

For our theory, we will find it convenient to work with the feature map defined by the Mercer decomposition. By Mercer's theorem (Mercer, 1909; Rasmussen & Williams, 2005) , the kernel admits a representation in terms of its M kernel eigenfunctions $\{\phi_\rho(\mathbf{x})\}$,

$$K(\mathbf{x}, \mathbf{x}') = \sum_{\rho=1}^{M} \lambda_\rho \phi_\rho(\mathbf{x}) \phi_\rho(\mathbf{x}') = \sum_{\rho=1}^{M} \psi_\rho(\mathbf{x}) \psi_\rho(\mathbf{x}_i), \tag{5}$$

where $\psi_\rho(\mathbf{x}) = \sqrt{\lambda_\rho}\phi(\mathbf{x})$ is the feature map we will work with. In our analysis, M will be taken to be infinite, but for the derivation of the learning curves, we will first consider M as a finite integer. The eigenfunctions and eigenvalues are defined with respect to the probability measure that generates the data $d\mu(\mathbf{x}) = p(\mathbf{x})d\mathbf{x}$

$$\int d\mathbf{x}'\, p(\mathbf{x}') K(\mathbf{x}, \mathbf{x}') \phi_\rho(\mathbf{x}') = \lambda_\rho \phi_\rho(\mathbf{x}). \tag{6}$$

We will also find it convenient to work with a vector representation of the RKHS functions in the feature space. Kernel eigenfunctions form a complete orthonormal basis, allowing the expansion of the target function f^* and learned function f in terms of features $\{\psi_\rho(\mathbf{x})\}$

$$f^*(\mathbf{x}) = \sum_\rho \overline{w}_\rho \psi_\rho(\mathbf{x}), \quad f(\mathbf{x}) = \sum_\rho w_\rho \psi_\rho(\mathbf{x}). \tag{7}$$

Hence, M-dimensional vectors $\mathbf{w}$ and $\overline{\mathbf{w}}$ constitute a representation of f and f^* respectively in the feature space.

We can also obtain a feature space expression for the optimal kernel regression function (2). Let $\mathbf{\Psi} \in \mathbb{R}^{M \times p}$ be feature matrix for the sample so that $\mathbf{\Psi}_{\rho,i} = \psi_\rho(\mathbf{x}_i)$. With this representation, kernel ridge regression (1) can be recast as the optimization problem $\min_{\mathbf{w} \in \mathbb{R}^M,\, \|\mathbf{w}\|_2 < \infty} \|\mathbf{\Psi}^\top \mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|^2$, whose solution is

$$\mathbf{w} = (\mathbf{\Psi}\mathbf{\Psi}^\top + \lambda \mathbf{I})^{-1} \mathbf{\Psi} \mathbf{y}. \tag{8}$$

Another novelty of our theory is the decomposition of the generalization error into its contributions from different eigenmodes. The feature space expression of the generalization error after averaging over the data distribution can be written as:

$$E_g = \sum_\rho E_\rho, \quad E_\rho \equiv \lambda_\rho \left\langle (w_\rho - \overline{w}_\rho)^2 \right\rangle_{\{\mathbf{x}_i\}, \overline{\mathbf{w}}}, \tag{9}$$

where we identify E_ρ as the generalization error in mode ρ.

Proof.

$$\begin{aligned} E_g &= \left\langle (f(x) - f^*(x))^2 \right\rangle_{\mathbf{x}, \{\mathbf{x}_i\}, f*} \\ &= \sum_{\rho,\gamma} \left\langle (w_\rho - \overline{w}_\rho)(w_\gamma - \overline{w}_\gamma) \right\rangle_{\{\mathbf{x}_i\}, f*} \left\langle \psi_\rho(\mathbf{x}) \psi_\gamma(\mathbf{x}) \right\rangle_{\mathbf{x}} \\ &= \sum_\rho \lambda_\rho \left\langle (w_\rho - \overline{w}_\rho)^2 \right\rangle_{\{\mathbf{x}_i\}, \overline{\mathbf{w}}} = \sum_\rho E_\rho. \end{aligned} \tag{10}$$

□

We introduce a matrix notation for RKHS eigenvalues $\mathbf{\Lambda}_{\rho,\gamma} \equiv \delta_{\rho,\gamma} \lambda_\rho$ for convenience. Finally, with our notation set up, we can present our first result about generalization error.

Proposition 1. *For the* $\mathbf{w}$ *that minimizes the training error (eq. (8)), the generalization error (eq. (4)) is given by*

$$E_g = \mathrm{Tr}\left(\mathbf{D}\left\langle \mathbf{G}^2 \right\rangle_{\{\mathbf{x}_i\}}\right), \tag{11}$$

which can be decomposed into modal generalization errors

$$E_\rho = \sum_\gamma \mathbf{D}_{\rho,\gamma}\left\langle \mathbf{G}^2_{\gamma,\rho}\right\rangle_{\{\mathbf{x}_i\}}, \tag{12}$$

where

$$\mathbf{G} = \left(\frac{1}{\lambda}\mathbf{\Phi}\mathbf{\Phi}^\top + \mathbf{\Lambda}^{-1}\right)^{-1}, \quad \mathbf{\Phi} = \mathbf{\Lambda}^{-1/2}\mathbf{\Psi}. \tag{13}$$

and

$$\mathbf{D} = \mathbf{\Lambda}^{-1/2}\left\langle \overline{\mathbf{w}}\overline{\mathbf{w}}^\top\right\rangle_{\overline{\mathbf{w}}}\mathbf{\Lambda}^{-1/2}. \tag{14}$$

We leave the proof to SI Section 2 but provide a few cursory observations of this result. First, note that all of the dependence on the teacher function comes in the matrix $\mathbf{D}$ whereas all of the dependence on the empirical samples is in $\mathbf{G}$. In the rest of the paper, we will develop multiple theoretical methods to calculate the generalization error given by expression (11).

Averaging over the target weights in the expression for $\mathbf{D}$ is easily done for generic weight distributions. The case of a fixed target is included by choosing a delta-function distribution over $\overline{\mathbf{w}}$.

We present two methods for computing the nontrivial average of the matrix $\mathbf{G}^2$ over the training samples $\{\mathbf{x}_i\}$. First, we consider the effect of adding a single new sample to $\mathbf{G}$ to derive a recurrence relation for $\mathbf{G}$ at different number of data points. This method generates a partial differential equation that must be solved to compute the generalization error. Second, we use a replica method and a saddle point approximation to calculate the matrix elements of $\mathbf{G}$. These approaches give identical predictions for the learning curves of kernel machines.

For notational simplicity, in the rest of the paper, we will use $\langle \ldots \rangle$ to mean $\langle \ldots \rangle_{\{\mathbf{x}_i\},\overline{\mathbf{w}}}$ unless stated otherwise. In all cases, the quantity inside the brackets will depend either on the data distribution or the distribution of target weights, but not both.

2.2. Continuous Approximation to Learning Curves

First, we adopt a method following Sollich (1999; 2002) and Sollich & Halees (2002) to calculate the generalization error. We generalize the definition of $\mathbf{G}$ by introducing an auxiliary parameter v, and make explicit its dataset size, p, dependence:

$$\tilde{\mathbf{G}}(p,v) = \left(\frac{1}{\lambda}\mathbf{\Phi}\mathbf{\Phi}^\top + \mathbf{\Lambda}^{-1} + v\mathbf{I}\right)^{-1}. \tag{15}$$

Note that the quantity we want to calculate is given by

$$\left\langle \mathbf{G}^2(p)\right\rangle = -\frac{\partial}{\partial v}\left\langle \tilde{\mathbf{G}}(p,v)\right\rangle\Big|_{v=0}. \tag{16}$$

By considering the effect of adding a single randomly sampled input $\mathbf{x}'$, and treating p as a continuous parameter, we can derive an approximate quasi-linear partial differential equation (PDE) for the average elements of $\mathbf{G}$ as a function of the number of data points p (see below for a derivation):

$$\frac{\partial\left\langle \tilde{\mathbf{G}}(p,v)\right\rangle}{\partial p} = \frac{1}{\lambda + \mathrm{Tr}\left\langle \tilde{\mathbf{G}}(p,v)\right\rangle}\frac{\partial}{\partial v}\left\langle \tilde{\mathbf{G}}(p,v)\right\rangle, \tag{17}$$

with the initial condition $\tilde{\mathbf{G}}(0,v) = (\mathbf{\Lambda}^{-1} + v\mathbf{I})^{-1}$, which follows from $\mathbf{\Phi}\mathbf{\Phi}^\top = \mathbf{0}$ when there is no data. Since $\tilde{\mathbf{G}}$ is initialized as a diagonal matrix, the off-diagonal elements will not vary under the dynamics and $\left\langle \tilde{\mathbf{G}}(p,v)\right\rangle$ will remain diagonal for all (p,v). We will use the solutions to this PDE and relation (16) to arrive at an approximate expression for the generalization error E_g and the mode errors E_ρ.

Derivation of the PDE approximation (17). Let $\phi \in \mathbb{R}^M$ represent the new feature to be added to $\mathbf{G}^{-1}$ so that $\phi_\rho = \phi_\rho(\mathbf{x}')$ where $\mathbf{x}' \sim p(\mathbf{x}')$ is a random sample from the data distribution. Let $\left\langle \tilde{\mathbf{G}}(p,v)\right\rangle_{\mathbf{\Phi}}$ denote the matrix $\tilde{\mathbf{G}}$ averaged over it's p-sample design matrix $\mathbf{\Phi}$. By the Woodbury matrix inversion formula

$$\begin{aligned}\left\langle \tilde{\mathbf{G}}(p+1,v)\right\rangle_{\mathbf{\Phi},\phi} &= \left\langle \left(\tilde{\mathbf{G}}(p,v)^{-1} + \frac{1}{\lambda}\phi\phi^\top\right)^{-1}\right\rangle_{\mathbf{\Phi},\phi} \\ &= \left\langle \tilde{\mathbf{G}}(p,v)\right\rangle_{\mathbf{\Phi}} - \left\langle \frac{\tilde{\mathbf{G}}(p,v)\phi\phi^\top\tilde{\mathbf{G}}(p,v)}{\lambda + \phi^\top\tilde{\mathbf{G}}(p,v)\phi}\right\rangle_{\mathbf{\Phi},\phi}. \end{aligned} \tag{18}$$

Performing the average of the last term on the right hand side is difficult so we resort to an approximation, where the numerator and denominator are averaged separately.

$$\left\langle \tilde{\mathbf{G}}(p+1,v)\right\rangle_{\mathbf{\Phi},\phi} \approx \left\langle \tilde{\mathbf{G}}(p,v)\right\rangle_{\mathbf{\Phi}} - \frac{\left\langle \tilde{\mathbf{G}}(p,v)^2\right\rangle_{\mathbf{\Phi}}}{\lambda + \mathrm{Tr}\left\langle \tilde{\mathbf{G}}(p,v)\right\rangle_{\mathbf{\Phi}}}, \tag{19}$$

where we used the fact that $\langle \phi_\rho(\mathbf{x}')\phi_\gamma(\mathbf{x}')\rangle_{\mathbf{x}'\sim p(\mathbf{x}')} = \delta_{\rho,\gamma}$.

Treating p as a continuous variable and taking a continuum limit of the finite differences given above, we arrive at (17). □

Next, we present the solution to the PDE (17) and the resulting generalization error.

Proposition 2. *Let* $g_\rho(p,v) = \left\langle \tilde{\mathbf{G}}(p,v)_{\rho\rho} \right\rangle$ *represent the diagonal elements of the average matrix* $\left\langle \tilde{\mathbf{G}}(p,v) \right\rangle$*. These matrix elements satisfy the implicit relationship*

$$g_\rho(p,v) = \left(\frac{1}{\lambda_\rho} + v + \frac{p}{\lambda + \sum_{\gamma=1}^{M} g_\gamma(p,v)} \right)^{-1}. \quad (20)$$

This implicit solution is obtained from the method of characteristics which we provide in Section 3 of the SI.

Proposition 3. *Under the PDE approximation* (17)*, the average error* E_ρ *associated with mode* ρ *is*

$$E_\rho(p) = \frac{\langle \overline{w}_\rho^2 \rangle}{\lambda_\rho} \left(\frac{1}{\lambda_\rho} + \frac{p}{\lambda + t(p)} \right)^{-2} \left(1 - \frac{p\gamma(p)}{(\lambda + t(p))^2} \right)^{-1}, \quad (21)$$

where $t(p) \equiv \sum_\rho g_\rho(p,0)$ *is the solution to the implicit equation*

$$t(p) = \sum_\rho \left(\frac{1}{\lambda_\rho} + \frac{p}{\lambda + t(p)} \right)^{-1}, \quad (22)$$

and $\gamma(p)$ *is defined as*

$$\gamma(p) = \sum_\rho \left(\frac{1}{\lambda_\rho} + \frac{p}{\lambda + t(p)} \right)^{-2}. \quad (23)$$

The full proof of this proposition is provided in Section 3 of the SI. We show the steps required to compute theoretical learning curves numerically in Algorithm 1.

Algorithm 1 Computing Theoretical Learning Curves

Input: RKHS spectrum $\{\lambda_\rho\}$, target function weights $\{\overline{w}_\rho\}$, regularizer λ, sample sizes $\{p_i\}$, $i = 1, ..., m$;
for $i = 1$ **to** m **do**
  Solve numerically $t_i = \sum_\rho \left(\frac{1}{\lambda_\rho} + \frac{p_i}{\lambda + t_i} \right)^{-1}$
  Compute $\gamma_i = \sum_\rho \left(\frac{1}{\lambda_\rho} + \frac{p_i}{\lambda + t_i} \right)^{-2}$
  $E_{\rho,i} = \frac{\langle \overline{w}_\rho^2 \rangle}{\lambda_\rho} \left(\frac{1}{\lambda_\rho} + \frac{p_i}{\lambda + t_i} \right)^{-2} \left(1 - \frac{p_i \gamma_i}{(\lambda + t_i)^2} \right)^{-1}$
end for

In eq. (21), the target function sets the overall scale of E_ρ. That E_ρ depends only on $\bar{w}_\rho$, but not other target modes, is an artifact of our approximation scheme, and in a full treatment may not necessarily hold. The spectrum of the kernel affects all modes in a nontrivial way. When we apply this theory to neural networks in Section 3, the information about the architecture of the network will be in the spectrum $\{\lambda_\rho\}$. The dependence on number of samples p is also nontrivial, but we will consider various informative limits below.

We note that though the mode errors fall asymptotically like p^{-2} (SI Section 4), the total generalization error E_g can scale with p in a nontrivial manner. For instance, if $\overline{w}_\rho^2 \lambda_\rho \sim \rho^{-a}$ and $\lambda_\rho \sim \rho^{-b}$ then a simple computation (SI Section 4) shows that $E_g \sim p^{-\min\{1-a,2b\}}$ as $p \to \infty$ for ridgeless regression and $E_g \sim p^{-\min\{1-a,2b\}/b}$ for explicitly regularized regression. This is consistent with recent observations that total generalization error for neural networks and kernel regression falls in a power law $E_g \sim p^{-\beta}$ with β dependent on kernel and target function (Hestness et al., 2017; Spigler et al., 2019).

2.3. Computing Learning Curves with Replica Method

The result of the continuous approximation can be obtained using another approximation method, which we outline here and detail in SI Section 5. We perform the average of matrix $\mathbf{G}(p,v)$ over the training data, using the replica method (Sherrington & Kirkpatrick, 1975; Mézard et al., 1987) from statistical physics and a finite size saddle-point approximation, and obtain identical learning curves to Proposition 3. Our starting point is a Gaussian integral representation of the matrix inverse

$$\langle \mathbf{G}(p,v)_{\rho,\gamma} \rangle = \frac{\partial^2}{\partial h_\rho \partial h_\gamma} R(p,v,\mathbf{h})|_{\mathbf{h}=0},$$
$$R(p,v,\mathbf{h}) \equiv \left\langle \frac{1}{Z} \int d\mathbf{u}\, e^{-\frac{1}{2}\mathbf{u}^\top (\frac{1}{\lambda}\mathbf{\Phi}\mathbf{\Phi}^\top + \mathbf{\Lambda}^{-1} + v\mathbf{I})\mathbf{u} + \mathbf{h}\cdot\mathbf{u}} \right\rangle, \quad (24)$$

where $Z = \int d\mathbf{u}\, e^{-\frac{1}{2}\mathbf{u}^\top (\frac{1}{\lambda}\mathbf{\Phi}\mathbf{\Phi}^\top + \mathbf{\Lambda}^{-1} + v\mathbf{I})\mathbf{u}}$. Since Z also depends on the dataset (quenched disorder) $\mathbf{\Phi}$, to make the average over $\mathbf{\Phi}$ tractable, we use the following limiting procedure: $Z^{-1} = \lim_{n\to 0} Z^{n-1}$. As is common in the physics of disordered systems (Mézard et al., 1987), we compute $R(p,v,\mathbf{h})$ for integer n and analytically continue the expressions in the $n \to 0$ limit under a symmetry ansatz. This procedure produces the same average matrix elements as the continuous approximation discussed in Proposition 2, and therefore the same generalization error given in Proposition 3. Further detail is provided in SI Section 5.

2.4. Spectral Dependency of Learning Curves

We can get insight about the behavior of learning curves by considering ratios between errors in different modes:

$$\frac{E_\rho}{E_\gamma} = \frac{\langle \overline{w}_\rho^2 \rangle}{\langle \overline{w}_\gamma^2 \rangle} \frac{\lambda_\gamma}{\lambda_\rho} \frac{(\frac{1}{\lambda_\gamma} + \frac{p}{\lambda + t})^2}{(\frac{1}{\lambda_\rho} + \frac{p}{\lambda + t})^2}. \quad (25)$$

For small p this ratio approaches $\frac{E_\rho}{E_\gamma} \sim \frac{\lambda_\rho \langle \overline{w}_\rho^2 \rangle}{\lambda_\gamma \langle \overline{w}_\gamma^2 \rangle}$. For large p, $\frac{E_\rho}{E_\gamma} \sim \frac{\langle \overline{w}_\rho^2 \rangle / \lambda_\rho}{\langle \overline{w}_\gamma^2 \rangle / \lambda_\gamma}$, indicating that asymptotically ($p \to \infty$), the amount of relative error in mode ρ grows with the ratio $\langle \overline{w}_\rho^2 \rangle / \lambda_\rho$, showing that the asymptotic mode error is

relatively large if the teacher function places large amounts of power in modes that have small RKHS eigenvalues λ_ρ.

We can also examine how the RKHS spectrum affects the evolution of the error ratios with p. Without loss of generality, we take $\lambda_\gamma > \lambda_\rho$ and show in SI Section 6 that

$$\frac{d}{dp}\log E_\rho > \frac{d}{dp}\log E_\gamma. \tag{26}$$

In this sense, the marginal training data point causes a greater percent reduction in generalization error for modes with larger RKHS eigenvalues.

2.5. Multiple Outputs

The learning curves we derive for a scalar function can be straightforwardly extended to the case where the function outputs are multivariate: $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^C$. For least squares regression, this case is equivalent to solving C separate learning problems for each component functions $f_c(\mathbf{x})$, $c = 1, ..., C$. Let $\mathbf{y}_c \in \mathbb{R}^p$ be the corresponding vectors of target values possibly generated by different target functions, f_c^*. The learning problem in this case is

$$\min_{f\in\mathcal{H}^c} \sum_{c=1}^{C}\left[\sum_{i=1}^{p}(f_c(\mathbf{x}_i) - y_{c,i})^2 + \lambda||f_c||_{\mathcal{H}}^2\right]. \tag{27}$$

The solution to the learning problem depends on the same kernel but different targets for each function:

$$f_c(\mathbf{x}) = \mathbf{y}_c^\top(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{k}(\mathbf{x}),\ c = 1, \ldots, C. \tag{28}$$

Our theory can be used to generate predictions for the generalization error of each of the C learned functions, $f_c(\mathbf{x})$, and then summed to obtain the total error.

3. Dot Product Kernels on $\mathbb{S}^{d-1}$ and NTK

For the remainder of the paper, we specialize to the case where our inputs are drawn uniformly on $\mathcal{X} = \mathbb{S}^{d-1}$, a $(d-1)$-dimensional unit hyper-sphere. In addition, we will assume that the kernel is a dot product kernel ($K(\mathbf{x}, \mathbf{x}') = \kappa(\mathbf{x}^\top\mathbf{x}')$), as is the case for NTK. In this setting, the kernel eigenfunctions are spherical harmonics $\{Y_{km}\}$ (Bietti & Mairal, 2019; Efthimiou & Frye, 2014), and the Mercer decomposition is given by

$$K(\mathbf{x}, \mathbf{x}') = \sum_{k=0}^{\infty}\lambda_k\sum_{m=1}^{N(d,k)} Y_{km}(\mathbf{x})Y_{km}(\mathbf{x}'). \tag{29}$$

Here, $N(d, k)$ is the dimension of the subspace spanned by d-dimensional spherical harmonics of degree k. Rotation invariance renders the eigenspectrum degenerate since each of the $N(d, k)$ modes of frequency k share the same eigenvalue λ_k. A review of these topics is given in SI Sections 7 and 8.

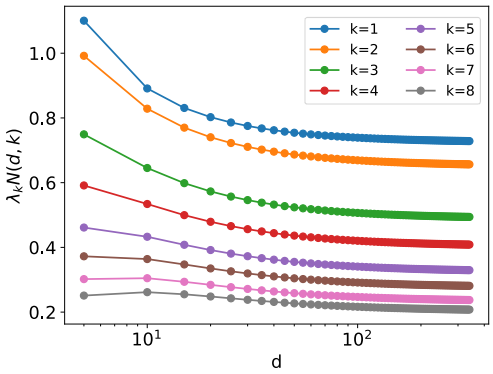


Figure 1. Spectrum of 10-layer NTK multiplied by degeneracy as a function of dimension for various k, calculated by numerical integration (SI Section 8). $\lambda_k N(d, k)$ stays constant as input dimension increases, confirming that $\lambda_k N(d, k)^{-1} \sim \mathcal{O}_d(1)$ at large d.

We briefly comment on another fact that will later be used in our numerical simulations. Dot product kernels admit an expansion in terms of Gegenbauer polynomials $\{Q_k\}$, which form a complete and orthonormal basis for the uniform measure on the sphere (Dai & Xu, 2013): $\kappa(z) = \sum_{k=0}^{\infty}\lambda_k N(d, k)Q_k(z)$. The Gegenbauer polynomials are related to spherical harmonics $\{Y_{km}\}$ through $Q_k(\mathbf{x}^\top\mathbf{x}') = \frac{1}{N(d,k)}\sum_{m=1}^{N(d,k)} Y_{km}(\mathbf{x})Y_{km}(\mathbf{x}')$ (Dai & Xu, 2013) (see SI Sections 7 and 8 for a review).

3.1. Frequency Dependence of Learning Curves

In the special case of dot product kernels with monotonically decaying spectra, results given in Section 2.4 indicate that the marginal training data point causes greater reduction in relative error for low frequency modes than for high frequency modes. Monotonic RKHS spectra represent an inductive bias that preferentially favors fitting lower frequencies as more data becomes available. More rapid decay in the spectrum yields a stronger bias to fit low frequencies first.

To make this intuition more precise, we now discuss an informative limit $d \to \infty$ where the degeneracy factor approaches to $N(d, k) \sim d^k/k!$. In the following, we replace eigenfunction index ρ with index pair (k, m). Eigenvalues of the kernel scales with d as $\lambda_k \sim N(d, k)^{-1}$ (Smola et al., 2001) in the $d \to \infty$ limit, as we verify numerically in Figure 1 for NTK. If we take $p = \alpha d^\ell$ for some integer degree ℓ, then E_{km} exhibits three distinct learning stages. Leaving the details to SI Section 9, we find that in this limit, for large α:

$$\frac{E_{km}(\alpha)}{E_{km}(0)} \approx \begin{cases} 1, & k > \ell \\ \frac{\text{const.}}{\alpha^2}, & k = \ell \\ 0, & k < \ell \end{cases}, \tag{30}$$

where the constant is given in SI Section 9. In other words, $k < l$ modes are perfectly learned, $k = l$ are being learned with an asymptotic $1/\alpha^2$ rate, and $k > l$ are not learned.

This simple calculation demonstrates that the lower modes are learned earlier with increasing sample complexity since the higher modes stays stationary until p reaches to the degeneracy of that mode.

3.2. Neural Tangent Kernel and its Spectrum

For fully connected architectures, the NTK is a rotation invariant kernel that describes how the predictions of infinitely wide neural networks evolve under gradient flow (Jacot et al., 2018). Let θ_i index all of the parameters of the neural network and let $f_\theta(\mathbf{x})$ be the output of the network. Here, we focus on scalar network outputs for simplicity, but generalization to multiple outputs is straightforward, as discussed in Section 2.5. Then the neural tangent kernel is defined as

$$K_{\text{NTK}}(\mathbf{x}, \mathbf{x}') = \sum_i \left\langle \frac{\partial f_\theta(\mathbf{x})}{\partial \theta_i} \frac{\partial f_\theta(\mathbf{x}')}{\partial \theta_i} \right\rangle_\theta . \tag{31}$$

Let $\mathbf{u}_\theta \in \mathbb{R}^p$ be the current predictions of f_θ on the training data. If the parameters of the model are updated via gradient flow on a quadratic loss, $\frac{d\theta}{dt} = -\nabla_\theta \mathbf{u}_\theta \cdot (\mathbf{u}_\theta - \mathbf{y})$, then the predictions on the training data evolve with the following dynamics (Pehlevan et al., 2018; Jacot et al., 2018; Arora et al., 2019; Lee et al., 2019)

$$\frac{d\mathbf{u}_\theta}{dt} = -\mathbf{K}_{\text{NTK}} \cdot (\mathbf{u}_\theta - \mathbf{y}) . \tag{32}$$

When the width of the neural network is taken to infinity with proper initialization, where the weights at layer ℓ are sampled $W^{(\ell)} \sim \mathcal{N}\left(0, 1/n^{(\ell)}\right)$ where $n^{(\ell)}$ is the number of hidden units in layer ℓ, the NTK becomes independent of the particular realization of parameters and approaches a deterministic function of the inputs and the nonlinear activation function (Jacot et al., 2018). Further, the kernel is approximately fixed throughout gradient descent (Jacot et al., 2018; Arora et al., 2019). If we assume that $\mathbf{u}_\theta = 0$ at $t = 0$, then the final learned function is

$$f(\mathbf{x}) = \mathbf{y}^\top \mathbf{K}_{\text{NTK}}^{-1} \mathbf{k}(\mathbf{x}). \tag{33}$$

Note that this corresponds to ridgeless, interpolating regression where $\lambda = 0$. We will use this correspondence and our kernel regression theory to explain neural network learning curves in the next section. For more information about NTK for fully connected architectures see SI Sections 10 and 11.

To generate theoretical learning curves, we need the eigenspectrum of the kernels involved. For $\mathcal{X} = \mathbb{S}^{d-1}$, it suffices to calculate the projections of the kernel on the Gegenbauer basis $\langle K_{\text{NTK}}(\mathbf{x}), Q_k(\mathbf{x}) \rangle_{\mathbf{x}}$, which we evaluate numerically with Gauss-Gegenbauer quadrature (SI Section 8). Further details on NTK spectrum is presented in SI Section 11.

4. Experiments

In this section, we test our theoretical results for kernel regression, kernel interpolation and wide networks for various kernels and datasets.

4.1. NTK Regression and Interpolation

We first test our theory in a kernel regression task with NTK demonstrating the spectral decomposition. In this experiment, the target function is a linear combination of a kernel evaluated at randomly sampled points $\{\overline{\mathbf{x}}_i\}$:

$$f^*(\mathbf{x}) = \sum_{i=1}^{p'} \overline{\alpha}_i K(\mathbf{x}, \overline{\mathbf{x}}_i), \tag{34}$$

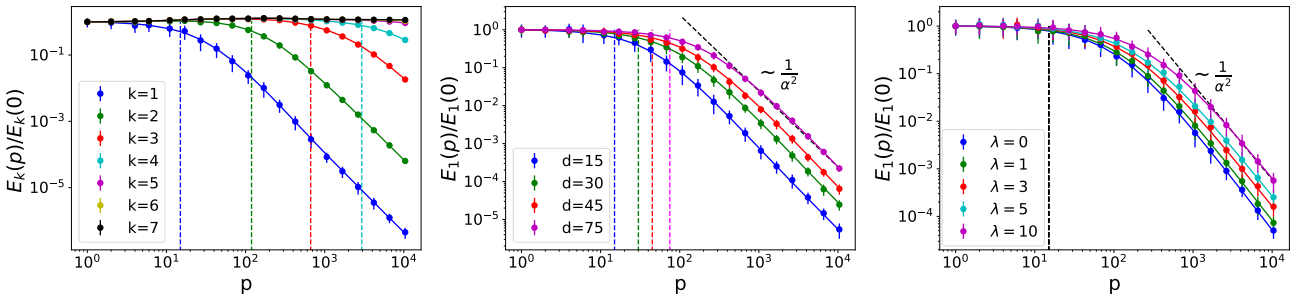
where $\overline{\alpha}_i \sim \mathcal{B}(1/2)$ are sampled randomly from a Bernoulli distribution on $\{\pm 1\}$ and $\overline{\mathbf{x}}_i$ are sampled uniformly from $\mathbb{S}^{d-1}$. The points $\overline{\mathbf{x}}_i$ are independent samples from $\mathbb{S}^{d-1}$ and are different than the training set $\{\mathbf{x}_i\}$. The student function is learned with kernel regression by inverting the Gram matrix $\mathbf{K}$ defined on the training samples $\{\mathbf{x}_i\}$ according to eq. (2). With this choice of target function, exact computation of the mode wise errors $E_k = \sum_m E_{km}$ in terms of Gegenbauer polynomials is possible; the formula and its derivation are provided in Section 12.2 of the SI. We compare these experimental mode-errors to those predicted by our theory and find perfect agreement. For these experiments, both the target and student kernels are taken to be NTK of a 4-layer fully connected ReLU without bias.

Figure 2 shows the errors for each frequency k as a function of sample size p. In Figure 2(a), we show that the mode errors sequentially start falling when $p \sim N(d, k)$. Figure 2(b) shows the mode error corresponding to $k = 1$ for kernel regression with 3-layer NTK across different dimensions. Higher input dimension causes the frequency modes to be learned at larger p. We observe an asymptotic $\sim 1/\alpha^2$ decay in modal errors. Finally, we show the effect of regularization on mode errors with a 10-layer NTK in Figure 2(c). With increasing λ, learning begins at larger p values.

4.2. Learning Curves for Finite Width Neural Networks

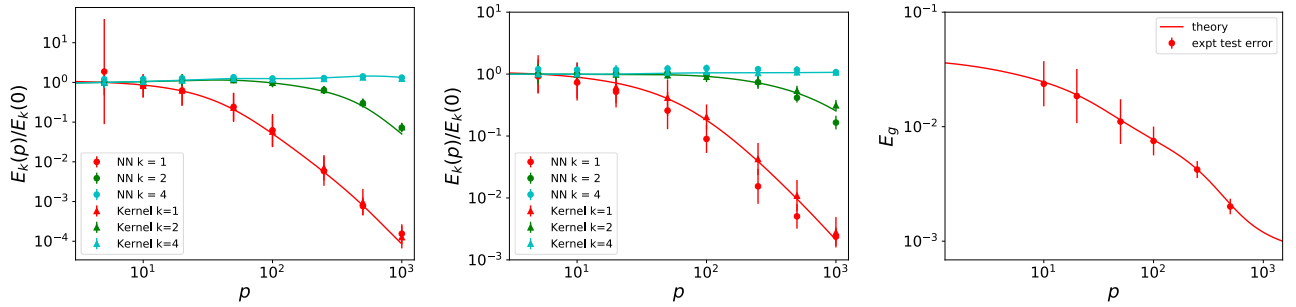
Having established that our theory accurately predicts the generalization error of kernel regression with NTK, we now compare the generalization error of finite width neural networks trained on a quadratic loss with the theoretical learning curves for NTK. For these experiments, we use the Neural-Tangents Library (Novak et al., 2020) which supports training and inference for both finite and infinite width neural networks.

First, we use "pure mode" teacher functions, meaning the teacher is composed only of spherical harmonics of the same



(a) 3-layer NTK $d = 15$ $\lambda = 0$ (b) 3-layer NTK $k = 1$ $\lambda = 1$ (c) 10-layer NTK $d = 15$ $k = 1$

Figure 2. Learning curves for kernel regression with NTK averaged over 50 trials compared to theory. Error bars are standard deviation. Solid lines are theoretical curves calculated using eq. (21). Dashed vertical lines indicate the degeneracy $N(d, k)$. (a) Normalized learning curves for different spectral modes. Sequential fitting of mode errors is visible. (b) Normalized learning curves for varying data dimension, d. (c) Learning curves for varying regularization parameter, λ.



(a) 2-layer NN $N = 10000$ (b) 4-layer NN $N = 500$ (c) 2-Layer NN Student-Teacher; $N = 8000$

Figure 3. (a) and (b) Learning curves for neural networks (NNs) on "pure modes" as defined in eq. (35). (c) Learning curve for the student teacher setup defined in (36). The theory curves shown as solid lines are again computed with eq. (21). The test error for the finite width neural networks and NTK are shown with dots and triangles respectively. The generalization error was estimated by taking a random test sample of 1000 data points. The average was taken over 25 trials and the standard deviations are shown with errorbars. The networks were initialized with the default Gaussian NTK parameterization (Jacot et al., 2018) and trained with stochastic gradient descent (details in SI Section 13).

degree. For "pure mode" k, the teacher is constructed with the following rule:

$$f^*(\mathbf{x}) = \sum_{i=1}^{p'} \overline{\alpha}_i Q_k(\mathbf{x}^\top \overline{\mathbf{x}}_i), \tag{35}$$

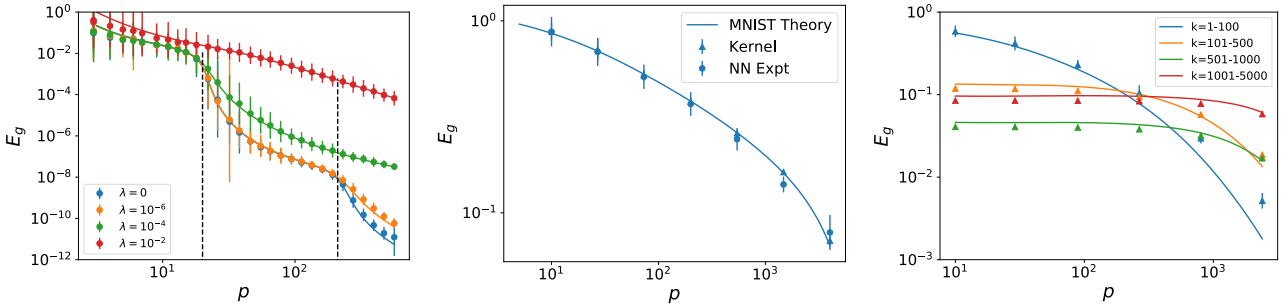
where again $\overline{\alpha}_i \sim \mathcal{B}(1/2)$ and $\overline{\mathbf{x}}_i \sim p(\mathbf{x})$ are sampled randomly. Figure 3(a) shows the learning curve for a fully connected 2-layer ReLU network with width $N = 10000$, input dimension $d = 30$ and $p' = 10000$. As before, we see that the lower k pure modes require less data to be fit. Experimental test errors for kernel regression with NTK on the same synthetic datasets are plotted as triangles. Our theory perfectly fits the experiments.

Results from a 4-layer NN simulation are provided in Figure 3(b). Each hidden layer had $N = 500$ hidden units. We again see that the $k = 2$ mode is only learned for $p > 200$. $k = 4$ mode is not learned at all in this range. Our theory again perfectly fits the experiments.

Lastly, we show that our theory also works for composite functions that contain many different degree spherical harmonics. In this setup, we randomly initialize a two layer teacher neural network and train a student neural network

$$f^*(\mathbf{x}) = \overline{\mathbf{r}}^\top \sigma(\overline{\mathbf{\Theta}}\mathbf{x}), \quad f(\mathbf{x}) = \mathbf{r}^\top \sigma(\mathbf{\Theta}\mathbf{x}), \tag{36}$$

where $\mathbf{\Theta}, \overline{\mathbf{\Theta}} \in \mathbb{R}^{M \times d}$ are the feedforward weights for the student and teacher respectively, σ is an activation function and $\mathbf{r}, \overline{\mathbf{r}} \in \mathbb{R}^M$ are the student and teacher readout weights. Chosen in this way with ReLU activations, the teacher is composed of spherical harmonics of many different degrees (Section 13 in SI). The total generalization error for this teacher student setup as well as the theoretical prediction of our theory is provided in Figure 3(c) for $d = 25$, $N = 8000$. They agree excellently. Results from additional neural net-



(a) Gaussian kernel and measure in $d = 20$ (b) 3-Layer NN on MNIST, $N = 800$ (c) NTK regression on MNIST, $\lambda = 0$.

Figure 4. (a) Learning curves for Gaussian kernel in $d = 20$ dimensions with varying λ. For small λ, learning stages are visible at $p = N(d, k)$ for $k = 1, 2$ ($p = 20, 210$, vertical dashed lines) but the stages are obscured for non-negligible λ. (b) Learning curve for 3-layer NTK regression and a neural network (NN) on a subset of 8000 randomly sampled images of handwritten digits from MNIST. (c) Aggregated NTK regression mode errors for the setup in (b). Eigenmodes of MNIST with larger eigenvalues are learned more rapidly with increasing p.

work simulations are provided in Section 13 of the SI.

4.3. Gaussian Kernel Regression and Interpolation

We next test our theory on another widely-used kernel. The setting where the probability measure and kernel are Gaussian, $K(\mathbf{x}, \mathbf{x}') = e^{-\frac{1}{2\ell^2}||\mathbf{x}-\mathbf{x}'||^2}$, allows analytical computation of the eigenspectrum, $\{\lambda_k\}$ (Rasmussen & Williams, 2005). In d dimensions, the k-th distinct eigenvalue corresponds to a set of $N(d, k) = \binom{d+k-1}{k} \sim d^k/k!$ degenerate eigenmodes. The spectrum itself decays exponentially.

In Figure 4(a), experimental learning curves for $d = 20$ dimensional standard normal random vector data and a Gaussian kernel with $\ell = 50$ are compared to our theoretical predictions for varying ridge parameters λ. The target function $f^*(\mathbf{x})$ is constructed with the same rule we used for the NTK experiments, shown in eq. 34. When λ is small, sharp drops in the generalization error occur when $p \approx N(d, k)$ for $k = 1, 2$. These drops are suppressed by the explicit regularization λ.

4.4. MNIST: Discrete Data Measure and Kernel PCA

We can also test our theory for finite datasets by defining a probability measure with equal point mass on each of the data points $\{\mathbf{x}_i\}_{i=1}^{\tilde{p}}$ in the dataset (including training and test sets):

$$p(\mathbf{x}) = \frac{1}{\tilde{p}} \sum_{i=1}^{\tilde{p}} \delta(\mathbf{x} - \mathbf{x}_i). \tag{37}$$

With this measure, the eigenvalue problem (6) becomes a $\tilde{p} \times \tilde{p}$ kernel PCA problem (see SI 14)

$$\mathbf{K}\mathbf{\Phi}^\top = \tilde{p}\mathbf{\Phi}^\top \mathbf{\Lambda}. \tag{38}$$

Once the eigenvalues $\mathbf{\Lambda}$ and eigenvectors $\mathbf{\Phi}^\top$ have been identified, we compute the target function coefficients by projecting the target data $\mathbf{y}_c$ onto these principal components $\overline{\mathbf{w}}_\mathbf{c} = \mathbf{\Lambda}^{-1/2}\mathbf{\Phi}\mathbf{y}_c$ for each target $c = 1, \ldots, C$. Once all of these ingredients are obtained, theoretical learning curves can be computed using Algorithm 1 and multiple class formalism described in Section 2.5, providing estimates of the error on the entire dataset incurred when training with a subsample of $p < \tilde{p}$ data points. An example where the discrete measure is taken as $\tilde{p} = 8000$ images of handwritten digits from MNIST (Lecun et al., 1998) and the kernel is NTK with 3 layers is provided in Figures 4(c)and 4(b). For total generalization error, we find perfect agreement between kernel regression and neural network experiments, and our theory.

5. Conclusion

In this paper, we presented an approximate theory of the average generalization performance for kernel regression. We studied our theory in the ridgeless limit to explain the behavior of trained neural networks in the infinite width limit (Jacot et al., 2018; Arora et al., 2019; Lee et al., 2019). We demonstrated how the RKHS eigenspectrum of NTK encodes a preferential bias to learn high spectral modes only after the sample size p is sufficiently large. Our theory fits kernel regression experiments remarkably well. We further experimentally verified that the theoretical learning curves obtained in the infinite width limit provide a good approximation of the learning curves for wide but finite-width neural networks. Our MNIST result suggests that our theory can be applied to datasets with practical value.

Acknowledgements

We thank Matthieu Wyart and Stefano Spigler for comments and pointing to a recent version of their paper (Spigler et al., 2019) with an independent derivation of the generalization error scaling for power law kernel and target spectra (see (SI.43)). C. Pehlevan thanks the Harvard Data Science Initiative, Google and Intel for support.

References

Abramowitz, M. and Stegun, I. *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*. U.S. Department of Commerce, National Bureau of Standards, 1972.

Arfken, G. *Mathematical Methods for Physicists*. Academic Press, Inc., San Diego, third edition, 1985.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R. R., and Wang, R. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, pp. 8139–8148, 2019.

Belkin, M., Hsu, D. J., and Mitra, P. Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate. In *Advances in neural information processing systems*, pp. 2300–2311, 2018a.

Belkin, M., Ma, S., and Mandal, S. To understand deep learning we need to understand kernel learning. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 541–549, 2018b.

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

Belkin, M., Rakhlin, A., and Tsybakov, A. B. Does data interpolation contradict statistical optimality? In *International Conference on Artificial Intelligence and Statistics*, pp. 1611–1619, 2019b.

Bietti, A. and Mairal, J. On the inductive bias of neural tangent kernels. In *Advances in Neural Information Processing Systems*, pp. 12873–12884, 2019.

Cao, Y., Fang, Z., Wu, Y., Zhou, D.-X., and Gu, Q. Towards understanding the spectral bias of deep learning, 2019.

Cohen, O., Malka, O., and Ringel, Z. Learning curves for deep neural networks: A gaussian field theory perspective, 2019.

Cucker, F. and Smale, S. Best choices for regularization parameters in learning theory: On the bias-variance problem. *Foundations of Computational Mathematics*, 2:413–428, 2002.

Dai, F. and Xu, Y. *Approximation Theory and Harmonic Analysis on Spheres and Balls*. Springer New York, 2013.

Dietrich, R., Opper, M., and Sompolinsky, H. Statistical mechanics of support vector networks. *Physical Review Letters*, 82(14):2975–2978, 1999.

Efthimiou, C. and Frye, C. *Spherical Harmonics In P Dimensions*. World Scientific Publishing Company, 2014.

Evgeniou, T., Pontil, M., and Poggio, T. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13, 1999.

Gyorfi, L., Kohler, M., Krzyzak, A., and Walk, H. A distribution-free theory of nonparametric regression. *Journal of the American Statistical Association*, 98(464): 1084–1084, 2003.

Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., Patwary, M. M. A., Yang, Y., and Zhou, Y. Deep learning scaling is predictable, empirically, 2017.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pp. 8571–8580, 2018.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521(7553):436–444, 2015.

Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in neural information processing systems*, pp. 8570–8581, 2019.

Liang, T. and Rakhlin, A. Just interpolate: Kernel "ridgeless" regression can generalize, 2018.

Luo, T., Ma, Z., Xu, Z.-Q. J., and Zhang, Y. Theory of the frequency principle for general deep neural networks, 2019.

Mercer, J. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A*, 209:415–446, 1909.

Mézard, M., Parisi, G., and Virasoro, M. *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*, volume 9. World Scientific Publishing Company, 1987.

Novak, R., Xiao, L., Hron, J., Lee, J., Alemi, A. A., Sohl-Dickstein, J., and Schoenholz, S. S. Neural tangents: Fast and easy infinite neural networks in python. In *International Conference on Learning Representations*, 2020.

Opper, M. and Vivarelli, F. General bounds on bayes errors for regression with gaussian processes. In *Advances in Neural Information Processing Systems 11*, pp. 302–308, 1998.

Pehlevan, C., Ali, F., and Ölveczky, B. P. Flexibility in motor timing constrains the topology and dynamics of pattern generator circuits. *Nature communications*, 9(1): 1–15, 2018.

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., and Courville, A. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 5301–5310, 2019.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

Schölkopf, B. and Smola, A. J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

Sherrington, D. and Kirkpatrick, S. Solvable model of a spin-glass. *Phys. Rev. Lett.*, 35:1792–1796, 1975.

Smola, A. J., Ovari, Z. L., and Williamson, R. C. Regularization with dot-product kernels. In *Advances in neural information processing systems*, pp. 308–314, 2001.

Sollich, P. Learning curves for gaussian processes. In *Advances in neural information processing systems*, pp. 344–350, 1999.

Sollich, P. Gaussian process regression with mismatched models. In *Advances in Neural Information Processing Systems*, pp. 519–526, 2002.

Sollich, P. and Halees, A. Learning curves for gaussian process regression: Approximations and bounds. *Neural Computation*, 14(6):1393–1428, 2002.

Spigler, S., Geiger, M., and Wyart, M. Asymptotic learning curves of kernel methods: empirical data v.s. Teacher-Student paradigm. *arXiv e-prints*, art. arXiv:1905.10843, 2019.

Spigler, S., Geiger, M., and Wyart, M. Asymptotic learning curves of kernel methods: empirical data v.s. teacher-student paradigm, 2019.

Vapnik, V. N. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

Wahba, G. *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, Philadelphia, 1990.

Xu, Z.-Q. J., Zhang, Y., Luo, T., Xiao, Y., and Ma, Z. Frequency principle: Fourier analysis sheds light on deep neural networks, 2019a.

Xu, Z.-Q. J., Zhang, Y., and Xiao, Y. Training behavior of deep neural network in frequency domain. In Gedeon, T., Wong, K. W., and Lee, M. (eds.), *Neural Information Processing*, pp. 264–274, 2019b.

Yang, G. and Salman, H. A fine-grained spectral perspective on neural networks, 2019.

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

Zhang, Y., Xu, Z.-Q. J., Luo, T., and Ma, Z. Explicitizing an implicit bias of the frequency principle in two-layer neural networks, 2019.