On the Generalization Power of Overfitted Two-Layer Neural Tangent Kernel Models

Peizhong Ju[1] **Xiaojun Lin**[1] **Ness B. Shroff**[2]

Abstract

In this paper, we study the generalization performance of min ℓ_2-norm overfitting solutions for the neural tangent kernel (NTK) model of a two-layer neural network with ReLU activation that has no bias term. We show that, depending on the ground-truth function, the test error of overfitted NTK models exhibits characteristics that are different from the "double-descent" of other overparameterized linear models with simple Fourier or Gaussian features. Specifically, for a class of learnable functions, we provide a new upper bound of the generalization error that approaches a small limiting value, even when the number of neurons p approaches infinity. This limiting value further decreases with the number of training samples n. For functions outside of this class, we provide a lower bound on the generalization error that does not diminish to zero even when n and p are both large.

1. Introduction

Recently, there is significant interest in understanding why overparameterized deep neural networks (DNNs) can still generalize well (Zhang et al., 2017; Advani et al., 2020), which seems to defy the classical understanding of *bias-variance tradeoff* in statistical learning (Bishop, 2006; Hastie et al., 2009; Stein, 1956; James & Stein, 1992; LeCun et al., 1991; Tikhonov, 1943). Towards this direction, a recent line of study has focused on overparameterized linear models (Belkin et al., 2018b; 2019; Bartlett et al., 2020; Hastie et al., 2019; Muthukumar et al., 2019; Ju et al., 2020; Mei & Montanari, 2019). For linear models with simple features (e.g., Gaussian features and Fourier features) (Belkin et al., 2018b; 2019; Bartlett et al., 2020; Hastie et al., 2019; Muthukumar et al., 2019; Ju et al., 2020), an interesting "double-descent" phenomenon has been observed. Thus, there is a region where the number of model parameters (or linear features) is larger than the number of samples (and thus overfitting occurs), but the generalization error actually decreases with the number of features. However, linear models with these simple features are still quite different from nonlinear neural networks. Thus, although such results provide some hint why overparameterization and overfitting may be harmless, it is still unclear whether similar conclusions apply to neural networks.

In this paper, we are interested in linear models based on the neural tangent kernel (NTK) (Jacot et al., 2018), which can be viewed as a useful intermediate step towards modeling nonlinear neural networks. Essentially, NTK can be seen as a linear approximation of neural networks when the weights of the neurons do not change much. Indeed, (Li & Liang, 2018; Du et al., 2018) have shown that, for a wide and fully-connected two-layer neural network, both the neuron weights and their activation patterns do not change much after gradient descent (GD) training with a sufficiently small step size. As a result, such a shallow and wide neural network is approximately linear in the weights when there are a sufficient number of neurons, which suggests the utility of the NTK model.

Despite its linearity, however, characterizing the double descent of such a NTK model remains elusive. The work in (Mei & Montanari, 2019) also studies the double-descent of a linear version of two-layer neural network. It uses the so-called "random-feature" model, where the bottom-layer weights are random and fixed, and only the top-layer weights are trained. (In comparison, the NTK model for such a two-layer neural network corresponds to training only the bottom-layer weights.) However, the setting there requires the number of neurons, the number of samples, and the data dimension to all grow proportionally to infinity. In contrast, we are interested in the setting where the number of samples is given, and the number of neurons is allowed to be much larger than the number of samples. As a consequence of the different setting, in (Mei & Montanari, 2019) eventually only *linear* ground-truth functions can be learned. (Similar settings are also studied in (d'Ascoli et al.,

[1] School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, USA. [2] Department of ECE and CSE, The Ohio State University, Columbus, Ohio, USA. Correspondence to: Xiaojun Lin <linx@purdue.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

2020).) In contrast, we will show that far more complex functions can be learned in our setting. In a related work, (Ghorbani et al., 2019) shows that both the random-feature model and the NTK model can approximate highly *non-linear* ground-truth functions with a sufficient number of neurons. However, (Ghorbani et al., 2019) mainly studies the *expressiveness* of the models, and therefore does not explain why overfitting solutions can still generalize well. To the best of our knowledge, our work is the first to characterize the double-descent of overfitting solutions based on the NTK model.

Specifically, in this paper we study the generalization error of the min ℓ_2-norm overfitting solution for a linear model based on the NTK of a two-layer neural network with ReLU activation that has no bias. Only the bottom-layer weights are trained. We are interested in min ℓ_2-norm overfitting solutions because gradient descent (GD) can be shown to converge to such solutions while driving the training error to zero (Zhang et al., 2017) (see also Section 2). Given a class of ground truth functions (see details in Section 3), which we refer to as "learnable functions," our main result (Theorem 1) provides an upper bound on the generalization error of the min ℓ_2-norm overfitting solution for the two-layer NTK model with n samples and p neurons (for any finite p larger than a polynomial function of n). This upper bound confirms that the generalization error of the overfitting solution indeed exhibits descent in the overparameterized regime when p increases. Further, our upper bound can also account for the noise in the training samples.

Our results reveal several important insights. First, we find that the (double) descent of the overfitted two-layer NTK model is drastically different from that of linear models with simple Gaussian or Fourier features (Belkin et al., 2018b; 2019; Bartlett et al., 2020; Hastie et al., 2019; Muthukumar et al., 2019). Specifically, for linear models with simple features, when the number of features p increases, the generalization error will eventually grow again and approach the so-called "null risk" (Hastie et al., 2019), which is the error of a trivial model that predicts zero. In contrast, for the class of learnable functions described earlier, the generalization error of the overfitted NTK model will continue to descend as p grows to infinity, and will approach a limiting value that depends on the number of samples n. Further, when there is no noise, this limiting value will decrease to zero as the number of samples n increases. This difference is shown in Fig. 1(a). As p increases, the test mean-square-error (MSE) of min-ℓ_1 and min-ℓ_2 overfitting solutions for Fourier features (blue and red curves) eventually grow back to the null risk (the black dashed line), even though they exhibit a descent at smaller p. In contrast, the error of the overfitted NTK model continues to descend to a much lower level.

The second important insight is that the aforementioned behavior critically depends on the ground-truth function belonging to the class of "learnable functions." Further, this class of learnable functions depend on the specific network architecture. For our NTK model (with RELU activation that has no bias), we precisely characterize this class of learnable functions. Specifically, for ground-truth functions that are outside the class of learnable functions, we show a lower bound on the generalization error that does not diminish to zero for any n and p (see Proposition 2 and Section 4). This difference is shown in Fig. 1(b), where we use an almost identical setting as Fig. 1(a), except a different ground-truth function. We can see in Fig. 1(b) that the test-error of the overfitted NTK model is always above the null risk and looks very different from that in Fig. 1(a). We note that whether certain functions are learnable or not critically depends on the specific structure of the NTK model, such as the choice of the activation unit. Recently, (Satpathi & Srikant, 2021) shows that all polynomials can be learned by 2-layer NTK model with ReLU activation that has a bias term, provided that the number of neurons p is sufficiently large. (See further discussions in Remark 2. However, (Satpathi & Srikant, 2021) does not characterize the descent of generalization errors as p increases.) This difference in the class of learnable functions between the two settings (ReLU with or without bias) also turns out to be consistent with the difference in the expressiveness of the neural networks. That is, shallow networks with biased-ReLU are known to be universal function approximators (Ji et al., 2019), while those without bias can only approximate the sum of linear functions and even functions (Ghorbani et al., 2019).

A closely related result to ours is the work in (Arora et al., 2019), which characterizes the generalization performance of wide two-layer neural networks whose bottom-layer weights are trained by gradient descent (GD) to overfit the training samples. In particular, our class of learnable functions almost coincides with that of (Arora et al., 2019). This is not surprising because, when the number of neurons is large, NTK becomes a close approximation of such two-layer neural networks. In that sense, the results in (Arora et al., 2019) are even more faithful in following the GD dynamics of the original two-layer network. However, the advantage of the NTK model is that it is easier to analyze. In particular, the results in this paper can quantify how the generalization error descends with p. In contrast, the results in (Arora et al., 2019) provide only a generalization bound that is independent of p (provided that p is sufficiently large), but do not quantify the descent behavior as p increases. Our numerical results in Fig. 1(a) suggest that, over a wide range of p, the descent behavior of the NTK model (the green curve) matches well with that of two-layer neural networks trained by gradient descent (the cyan curve). Thus, we believe that our results also provide guidance for the latter

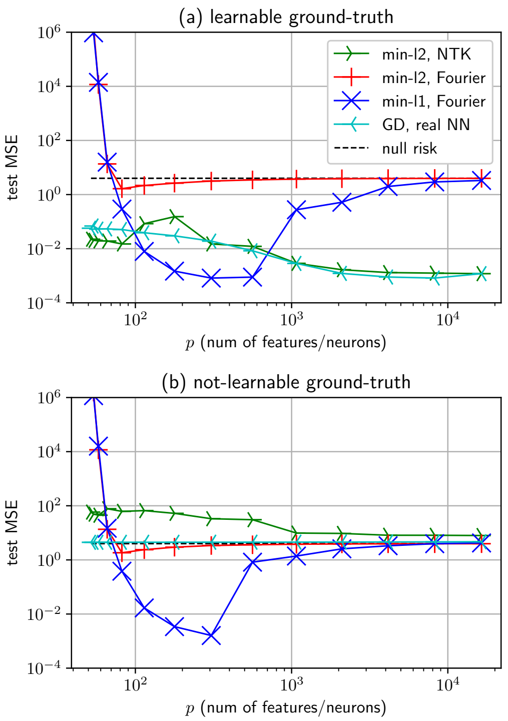

Figure 1. The test mean-square-error(MSE) vs. the number of features/neurons p for **(a)** learnable function and **(b)** not-learnable function when $n = 50$, $d = 2$, $\|\boldsymbol{\epsilon}\|_2^2 = 0.01$. The corresponding ground-truth are **(a)** $f(\theta) = \sum_{k\in\{0,1,2,4\}}(\sin(k\theta) + \cos(k\theta))$, and **(b)** $f(\theta) = \sum_{k\in\{3,5,7,9\}}(\sin(k\theta) + \cos(k\theta))$. (Note that in 2-dimension every input $\boldsymbol{x}$ on a unit circle can be represented by an angle $\theta \in [-\pi, \pi]$. See the end of Section 4.) Every curve is the average of 9 random simulation runs. For GD on the real neural network (NN), we use the step size $1/\sqrt{p}$ and the number of training epochs is fixed at 2000.

model. The work in (Fiat et al., 2019) studies a different neural network architecture with gated ReLU, whose NTK model turns out to be the same as ours. However, similar to (Arora et al., 2019), the result in (Fiat et al., 2019) does not capture the speed of descent with respect to p either. Second, (Arora et al., 2019) only provides upper bounds on the generalization error. There is no corresponding lower bound to explain whether ground-truth functions outside a certain class are *not* learnable. Our result in Proposition 2 provides such a lower bound, and therefore more completely characterizes the class of learnable functions. (See further comparison in Remark 1 of Section 3 and Remark 3 of Section 5.) Another related work (Allen-Zhu et al., 2019) also characterizes the class of learnable functions for two-layer and three-layer networks. However, (Allen-Zhu et al., 2019) studies a training method that takes a new sample in every iteration, and thus does not overfit all training data. Finally, our paper studies generalization of NTK models for the regression setting, which is different from the classification setting that assumes a separability condition, e.g., in (Ji & Telgarsky, 2019).

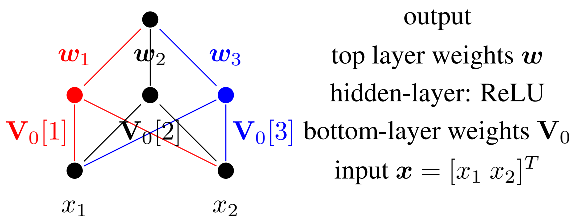

Figure 2. A two-layer neural network where $d = 2$, $p = 3$.

2. Problem Setup

We assume the following data model $y = f(\boldsymbol{x}) + \epsilon$, with the input $\boldsymbol{x} \in \mathbb{R}^d$, the output $y \in \mathbb{R}$, the noise $\epsilon \in \mathbb{R}$, and $f : \mathbb{R}^d \mapsto \mathbb{R}$ denotes the ground-truth function. Let $(\mathbf{X}_i,\ y_i)$, $i = 1, 2, \cdots, n$ denote n training samples. We collect them as $\mathbf{X} = [\mathbf{X}_1\ \mathbf{X}_2\ \cdots\ \mathbf{X}_n] \in \mathbb{R}^{d\times n}$, $\boldsymbol{y} = [y_1\ y_2\ \cdots\ y_n]^T \in \mathbb{R}^n$, $\boldsymbol{\epsilon} = [\epsilon_1\ \epsilon_2\ \cdots\ \epsilon_n]^T \in \mathbb{R}^n$, and $\mathbf{F}(\mathbf{X}) = [f(\mathbf{X}_1)\ f(\mathbf{X}_2)\ \cdots\ f(\mathbf{X}_n)]^T \in \mathbb{R}^n$. Then, the training samples can be written as $\boldsymbol{y} = \mathbf{F}(\mathbf{X}) + \boldsymbol{\epsilon}$. After training (to be described below), we denote the trained model by the function $\hat{f}$. Then, for any new test data $\boldsymbol{x}$, we will calculate the test error by $|\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x})|$, and the mean squared error (MSE) by $\mathsf{E}_{\boldsymbol{x}}[\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x})]^2$.

For training, consider a fully-connected two-layer neural network with p neurons. Let $\boldsymbol{w}_j \in \mathbb{R}$ and $\mathbf{V}_0[j] \in \mathbb{R}^d$ denote the top-layer and bottom-layer weights, respectively, of the j-th neuron, $j = 1, 2, \cdots, p$ (see Fig. 2). We collect them into $\boldsymbol{w} = [\boldsymbol{w}_1\ \boldsymbol{w}_2\ \cdots\ \boldsymbol{w}_p]^T \in \mathbb{R}^p$, and $\mathbf{V}_0 = [\mathbf{V}_0[1]^T\ \mathbf{V}_0[2]^T\ \cdots\ \mathbf{V}_0[p]^T]^T \in \mathbb{R}^{dp}$ (a column vector with dp elements). Note that with this notation, for any row or column vector $\boldsymbol{v}$ with dp elements, $\boldsymbol{v}[j]$ denotes a (row/column) vector that consists of the $(jd + 1)$-th to $(jd + d)$-th elements of $\boldsymbol{v}$. We choose ReLU as the activation function for all neurons and there is no bias term in the ReLU activation function.

Now we are ready to introduce the NTK model (Jacot et al., 2018). We fix the top-layer weights $\boldsymbol{w}$, and let the initial bottom-layer weights $\mathbf{V}_0$ be randomly chosen. We then train only the bottom-layer weights. Let $\mathbf{V}_0 + \overline{\Delta\mathbf{V}}$ denote the bottom-layer weights after training. Thus, the change of the output after training is

$$\sum_{j=1}^{n} \boldsymbol{w}_j \mathbf{1}_{\{\boldsymbol{x}^T(\mathbf{V}_0[j]+\overline{\Delta\mathbf{V}}[j])>0\}} \cdot (\mathbf{V}_0[j] + \overline{\Delta\mathbf{V}}[j])^T\boldsymbol{x}$$
$$-\sum_{j=1}^{n} \boldsymbol{w}_j \mathbf{1}_{\{\boldsymbol{x}^T\mathbf{V}_0[j]>0\}} \cdot \mathbf{V}_0[j]^T\boldsymbol{x}.$$

In the NTK model, one assumes that $\overline{\Delta\mathbf{V}}$ is very small. As a result, $\mathbf{1}_{\{\boldsymbol{x}^T(\mathbf{V}_0[j]+\overline{\Delta\mathbf{V}}[j])>0\}} = \mathbf{1}_{\{\boldsymbol{x}^T\mathbf{V}_0[j]>0\}}$ for most $\boldsymbol{x}$. Thus, the change of the output can be approximated by

$$\sum_{j=1}^{n} \boldsymbol{w}_j \mathbf{1}_{\{\boldsymbol{x}^T\mathbf{V}_0[j]>0\}} \cdot \overline{\Delta\mathbf{V}}[j]^T \boldsymbol{x} = \boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}} \Delta\mathbf{V},$$

where $\Delta\mathbf{V} \in \mathbb{R}^{dp}$ is given by $\Delta\mathbf{V}[j] := \boldsymbol{w}_j \overline{\Delta\mathbf{V}}[j],\ j = 1, 2, \cdots, p$, and $\boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}} \in \mathbb{R}^{1\times(dp)}$ is given by

$$\boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}}[j] := \mathbf{1}_{\{\boldsymbol{x}^T\mathbf{V}_0[j]>0\}} \cdot \boldsymbol{x}^T,\ j = 1, 2, \cdots, p. \tag{1}$$

In the NTK model, we assume that the output of the trained model is exactly given by Eq. (1), i.e.,

$$\hat{f}_{\Delta\mathbf{V},\mathbf{V}_0}(\boldsymbol{x}) := \boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}} \Delta\mathbf{V}. \tag{2}$$

In other words, the NTK model can be viewed as a linear approximation of the two-layer network when the change of the bottom-layer weights is small.

Define $\mathbf{H} \in \mathbb{R}^{n\times(dp)}$ such that its i-th row is $\mathbf{H}_i := \boldsymbol{h}_{\mathbf{V}_0,\mathbf{X}_i}$. Throughout the paper, we will focus on the following min-ℓ_2-norm overfitting solution

$$\Delta\mathbf{V}^{\ell_2} := \operatorname*{arg\,min}_{\boldsymbol{v}} \|\boldsymbol{v}\|_2, \text{ subject to } \mathbf{H}\boldsymbol{v} = \boldsymbol{y}.$$

Whenever $\Delta\mathbf{V}^{\ell_2}$ exists, it can be written in closed form as

$$\Delta\mathbf{V}^{\ell_2} = \mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}\boldsymbol{y}. \tag{3}$$

The reason that we are interested in $\Delta\mathbf{V}^{\ell_2}$ is that gradient descent (GD) or stochastic gradient descent (SGD) for the NTK model in Eq. (2) is known to converge to $\Delta\mathbf{V}^{\ell_2}$ (proven in Supplementary Material, Appendix B).

Using Eq. (2) and Eq. (3), the trained model is then

$$\hat{f}^{\ell_2}(\boldsymbol{x}) := \boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}} \Delta\mathbf{V}^{\ell_2}. \tag{4}$$

In the rest of the paper, we will study the generalization error of Eq. (4).

We collect some assumptions. Define the unit sphere in $\mathbb{R}^d$ as: $\mathcal{S}^{d-1} := \{\boldsymbol{v} \in \mathbb{R}^d \mid \|\boldsymbol{v}\|_2 = 1\}$. Let $\mu(\cdot)$ denote the distribution of the input $\boldsymbol{x}$. Without loss of generality, we make the following assumptions: **(i)** the inputs $\boldsymbol{x}$ are *i.i.d.* uniformly distributed in $\mathcal{S}^{d-1}$, and the initial weights $\mathbf{V}_0[j]$'s are *i.i.d.* uniformly distributed in all directions in $\mathbb{R}^d$; **(ii)** $p \geq n/d$ and $d \geq 2$; **(iii)** $\mathbf{X}_i \not\parallel \mathbf{X}_j$ for any $i \neq j$, and $\mathbf{V}_0[k] \not\parallel \mathbf{V}_0[l]$ for any $k \neq l$. We provide detailed justification of those assumptions in Supplementary Material, Appendix C.

3. Learnable Functions and Generalization Performance

We now show that the generalization performance of the overfitted NTK model in Eq. (4) crucially depends on the ground-truth function $f(\cdot)$, where good generalization performance only occurs when the ground-truth function is "learnable." Below, we first describe a candidate class of ground-truth functions, and explain why they may correspond to the class of "learnable functions." Then, we will give an upper-bound on the generalization performance for this class of ground-truth functions. Finally, we will give a lower-bound on the generalization performance when the ground-truth functions are outside of this class.

We first define a set $\mathcal{F}^{\ell_2}$ of ground-truth functions.

Definition 1. $\mathcal{F}^{\ell_2} := \Big\{f \overset{a.e.}{=} f_g \mid f_g(\boldsymbol{x}) = \int_{\mathcal{S}^{d-1}} \boldsymbol{x}^T\boldsymbol{z}\frac{\pi-\arccos(\boldsymbol{x}^T\boldsymbol{z})}{2\pi} g(\boldsymbol{z}) d\mu(\boldsymbol{z}),\ \|g\|_1 < \infty\Big\}$.

Note that in Definition 1, $\overset{\text{a.e.}}{=}$ means two functions equals almost everywhere, and $\|g\|_1 := \int_{\mathcal{S}^{d-1}} |g(\boldsymbol{z})| d\mu(\boldsymbol{z})$. The function $g(\boldsymbol{z})$ may be any finite-value function in $L^1(\mathcal{S}^{d-1} \mapsto \mathbb{R})$. Further, we also allow $g(\boldsymbol{z})$ to contain (as components) Dirac δ-functions on $\mathcal{S}^{d-1}$. Note that a δ-function $\delta_{\boldsymbol{z}_0}(\boldsymbol{z})$ has zero value for all $\boldsymbol{z} \in \mathcal{S}^{d-1} \setminus \{\boldsymbol{z}_0\}$, but $\|\delta_{\boldsymbol{z}_0}\|_1 := \int_{\mathcal{S}^{d-1}} \delta_{\boldsymbol{z}_0}(\boldsymbol{z}) d\mu(\boldsymbol{z}) = 1$. Thus, the function $g(\boldsymbol{z})$ may contain any sum of δ-functions and finite-value L^1-functions. [1]

To see why $\mathcal{F}^{\ell_2}$ may correspond to the class of learnable functions, we can first examine what the learned function $\hat{f}^{\ell_2}$ in Eq. (4) should look like. Recall that $\mathbf{H}^T = [\mathbf{H}_1^T \ \cdots \ \mathbf{H}_n^T]$. Thus, $\boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}}\mathbf{H}^T = \sum_{i=1}^n (\boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}}\mathbf{H}_i^T)\boldsymbol{e}_i^T$, where $\boldsymbol{e}_i \in \mathbb{R}^n$ denotes the i-th standard basis. Combining Eq. (3) and Eq. (4), we can see that the learned function in Eq. (4) is of the form

$$\begin{aligned}\hat{f}^{\ell_2}(\boldsymbol{x}) =& \boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}}\mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}\boldsymbol{y} \\ =& \sum_{i=1}^n \left(\frac{1}{p}\boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}}\mathbf{H}_i^T\right) p\boldsymbol{e}_i^T(\mathbf{H}\mathbf{H}^T)^{-1}\boldsymbol{y}. \end{aligned}\tag{5}$$

For all $\boldsymbol{x}, \boldsymbol{z} \in \mathcal{S}^{d-1}$, define $\mathcal{C}_{\boldsymbol{z},\boldsymbol{x}}^{\mathbf{V}_0} := \{j \in \{1, 2, \cdots, p\} \mid \boldsymbol{z}^T\mathbf{V}_0[j] > 0, \boldsymbol{x}^T\mathbf{V}_0[j] > 0\}$, and its cardinality is given by

$$\left|\mathcal{C}_{\boldsymbol{z},\boldsymbol{x}}^{\mathbf{V}_0}\right| = \sum_{j=1}^p \mathbf{1}_{\{\boldsymbol{z}^T\mathbf{V}_0[j]>0,\ \boldsymbol{x}^T\mathbf{V}_0[j]>0\}}. \tag{6}$$

Then, using Eq. (1), we can show $\frac{1}{p}\boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}}\mathbf{H}_i^T = \boldsymbol{x}^T\mathbf{X}_i\frac{|\mathcal{C}_{\mathbf{X}_i,\boldsymbol{x}}^{\mathbf{V}_0}|}{p}$. It is not hard to show that

$$\frac{|\mathcal{C}_{\boldsymbol{z},\boldsymbol{x}}^{\mathbf{V}_0}|}{p} \xrightarrow{\mathrm{P}} \frac{\pi - \arccos(\boldsymbol{x}^T\boldsymbol{z})}{2\pi}, \text{ as } p \to \infty. \tag{7}$$

[1] Alternatively, we can also interpret $g(\boldsymbol{z})$ as a signed measure (Rao & Rao, 1983) on $\mathcal{S}^{d-1}$. Then, δ-functions correspond to point masses, and the condition $\|g\|_1 < \infty$ implies that the corresponding unsigned version of the measure on $\mathcal{S}^{d-1}$ is bounded.

where $\xrightarrow{\mathrm{P}}$ denotes converge in probability. (see Supplementary Material, Appendix D.5). Thus, if we let

$$g(\boldsymbol{z}) = \sum_{i=1}^{n} p\boldsymbol{e}_i^T(\mathbf{H}\mathbf{H}^T)^{-1}\boldsymbol{y}\delta_{\mathbf{X}_i}(\boldsymbol{z}), \tag{8}$$

then as $p \to \infty$, Eq. (5) should approach a function in $\mathcal{F}^{\ell_2}$. This explains why $\mathcal{F}^{\ell_2}$ is a candidate class of "learnable functions." However, note that the above discussion only addresses the *expressiveness* of the model. It is still unclear whether any function in $\mathcal{F}^{\ell_2}$ can be learned with low generalization error. The following result provides the answer.

For some $m \in \left[1,\ \frac{\ln n}{\ln\frac{\pi}{2}}\right]$, define (recall that d is the dimension of $\boldsymbol{x}$)

$$J_m(n,d) := 2^{1.5d+5.5}d^{0.5d}n^{\left(2+\frac{1}{m}\right)(d-1)}. \tag{9}$$

Theorem 1. *Assume a ground-truth function* $f \overset{a.e.}{=} f_g \in \mathcal{F}^{\ell_2}$ *where* $\|g\|_\infty < \infty$[2], $n \geq 2$, $m \in \left[1,\ \frac{\ln n}{\ln\frac{\pi}{2}}\right]$, $d \leq n^4$, *and* $p \geq 6J_m(n,d)\ln\left(4n^{1+\frac{1}{m}}\right)$. *Then, for any* $q \in [1,\ \infty)$ *and for almost every* $\boldsymbol{x} \in \mathcal{S}^{d-1}$, *we must have*[3]

$$\begin{aligned}
&\Pr_{\mathbf{V}_0,\mathbf{X}}\Bigg\{|\hat{f}^{\ell_2}(\boldsymbol{x}) - f(\boldsymbol{x})| \geq \underbrace{n^{-\frac{1}{2}\left(1-\frac{1}{q}\right)}}_{\textit{Term 1}} \\
&+ \underbrace{\left(1+\sqrt{J_m(n,d)n}\right)p^{-\frac{1}{2}\left(1-\frac{1}{q}\right)}}_{\textit{Term 2}} + \underbrace{\sqrt{J_m(n,d)n}\|\boldsymbol{\epsilon}\|_2}_{\textit{Term 3}}, \\
&\quad \textit{for all } \boldsymbol{\epsilon} \in \mathbb{R}^n\Bigg\} \leq 2e^2\Bigg(\underbrace{\exp\left(-\frac{\sqrt[q]{n}}{8\|g\|_\infty^2}\right)}_{\textit{Term 4}} \\
&+ \underbrace{\exp\left(-\frac{\sqrt[q]{p}}{8\|g\|_1^2}\right)}_{\textit{Term 5}} + \underbrace{\exp\left(-\frac{\sqrt[q]{p}}{8n\|g\|_1^2}\right)}_{\textit{Term 6}}\Bigg) + \underbrace{\frac{4}{\sqrt[m]{n}}}_{\textit{Term 7}}.
\end{aligned} \tag{10}$$

To interpret Theorem 1, we can first focus on the noiseless case, where $\boldsymbol{\epsilon}$ and Term 3 are zero. If we fix n and let $p \to \infty$, then Terms 2, 5, and 6 all approach zero. We can then conclude that, in the noiseless and heavily overparameterized setting ($p \to \infty$), the generalization error will converge to a small limiting value (Term 1) that depends only on n. Further, this limiting value (Term 1) will converge to zero (so do Terms 4 and 7) as $n \to \infty$, i.e., when there are sufficiently many training samples. Finally, Theorem 1 holds even when there is noise.

The parameters of q and m can be tuned to make Eq. (10) sharper when n and p are large. For example, as we increase q, Term 1 will approach $n^{-0.5}$. Although a larger q makes Terms 4, 5, and 6 bigger, as long as n and p are sufficiently large, those terms will still be close to 0. Similarly, if we increase m, then $J_m(n,d)$ will approach the order of $n^{2(d-1)}$. As a result, Term 3 approaches the order of $n^{2d-0.5}$ times $\|\boldsymbol{\epsilon}\|_2$ and the requirement $p \geq 6J_m(n,d)\ln\left(4n^{1+\frac{1}{m}}\right)$ approaches the order of $n^{2(d-1)}\ln n$.

Remark 1. We note that (Arora et al., 2019) shows that, for two-layer neural networks whose bottom-layer weights are trained by gradient descent, the generalization error for sufficiently large p has the following upper bound: for any $\zeta > 0$,

$$\begin{aligned}
\Pr\Bigg\{\mathop{\mathsf{E}}_{\boldsymbol{x}}|\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x})| &\leq \sqrt{\frac{2\boldsymbol{y}^T(\mathbf{H}^\infty)^{-1}\boldsymbol{y}}{n}} \\
&+ O\left(\sqrt{\frac{\log\frac{n}{\zeta\cdot\min\mathsf{eig}(\mathbf{H}^\infty)}}{n}}\right)\Bigg\} \geq 1-\zeta,
\end{aligned} \tag{11}$$

where $\mathbf{H}^\infty = \lim_{p\to\infty}(\mathbf{H}\mathbf{H}^T/p) \in \mathbb{R}^{n\times n}$. For certain class of learnable functions (we will compare them with our $\mathcal{F}^{\ell_2}$ in Section 4), the quantity $\boldsymbol{y}^T(\mathbf{H}^\infty)^{-1}\boldsymbol{y}$ is bounded. Thus, $\sqrt{\frac{2\boldsymbol{y}^T(\mathbf{H}^\infty)^{-1}\boldsymbol{y}}{n}}$ also decreases at the speed $1/\sqrt{n}$. The second $O(\cdot)$-term in Eq. (11) contains the minimum eigenvalue of $\mathbf{H}^\infty$, which decreases with n. (Indeed, we show that this minimum eigenvalue is upper bounded by $O(n^{-\frac{1}{d-1}})$ in Supplementary Material, Appendix G.) Thus, Eq. (11) may decrease a little bit slower than $1/\sqrt{n}$, which is consistent with Term 1 in Eq. (10) (when q is large). Note that the term $2\boldsymbol{y}^T(\mathbf{H}^\infty)^{-1}\boldsymbol{y}$ in Eq. (11) captures how the complexity of the ground-truth function affects the generalization error. Similarly, the norm of $g(\cdot)$ also captures the impact[4] of the complexity of the ground-truth function in Eq. (10). However, we caution that the GD solution in (Arora et al., 2019) is based on the original neural network, which is usually different from our min ℓ_2-norm solution based on the NTK model (even though they are close for very large p). Thus, the two results may not be directly comparable.

Theorem 1 reveals several important insights on the generalization performance when the ground-truth function belongs to $\mathcal{F}^{\ell_2}$.

(i) Descent in the overparameterized region: When p increases, both sides of Eq. (10) decreases, suggesting that the test error of the overfitted NTK model decreases with

[2] The requirement of $\|g\|_\infty < \infty$ can be relaxed. We show in Supplementary Material, Appendix L that, even when g is a δ-function (so $\|g\|_\infty = \infty$), we can still have a similar result of Eq. (10) but Term 1 will have a slower speed of decay $O(n^{-\frac{1}{2(d-1)}(1-\frac{1}{q})})$ with respect to n instead of $O(n^{-\frac{1}{2}}(1-\frac{1}{q}))$ shown in Eq. (10). Term 4 of Eq. (10) will also be different when g is a δ-function, but it still goes to zero when p and n are large.

[3] The notion $\Pr_M$ in Eq. (10) emphasizes that randomness is in M.

[4] Although Term 1 in Eq. (10) in its current form does not depend on $g(\cdot)$, it is possible to modify our proof so that the norm of $g(\cdot)$ also enters Term 1.

p. In Fig. 1(a), we choose a ground-truth function in $\mathcal{F}^{\ell_2}$ (we will explain why this function is in $\mathcal{F}^{\ell_2}$ later in Section 4). The test MSE of the aforementioned NTK model (green curve) confirms the overall trend[5] of descent in the overparameterized region. We note that while (Arora et al., 2019) provides a generalization error upper-bound for large p (i.e., Eq. (11)), the upper bound there does not capture the dependency in p and thus does not predict this descent.

More importantly, we note a significant difference between the descent in Theorem 1 and that of min ℓ_2-norm overfitting solutions for linear models with simple features (Belkin et al., 2018b; 2019; Bartlett et al., 2020; Hastie et al., 2019; Muthukumar et al., 2019; Liao et al., 2020; Jacot et al., 2020). For example, for linear models with Gaussian features, we can obtain (see, e.g., Theorem 2 of (Belkin et al., 2019)):

$$\text{MSE} = \|f\|_2^2\left(1 - \frac{n}{p}\right) + \frac{\sigma^2 n}{p-n-1}, \text{ for } p \geq n+2 \tag{12}$$

where σ^2 denotes the variance of the noise. If we let $p \to \infty$ in Eq. (12), we can see that the MSE quickly approaches $\|f\|_2^2$, which is referred to as the "null risk" (Hastie et al., 2019), i.e., the MSE of a model that predicts zero. Note that the null-risk is at the level of the signal, and thus is quite large. In contrast, as $p \to \infty$, the test error of the NTK model converges to a value determined by n and $\boldsymbol{\epsilon}$ (and is independent of the null risk). This difference is confirmed in Fig. 1(a), where the test MSE for the NTK model (green curve) is much lower than the null risk (the dashed line) when $p \to \infty$, while both the min ℓ_2-norm (the red curve) and the min ℓ_1-norm solutions (the blue curve) (Ju et al., 2020) with Fourier features rise to the null risk when $p \to \infty$. Finally, note that the descent in Theorem 1 requires p to increase much faster than n. Specifically, to keep Term 2 in Eq. (10) small, it suffices to let p increase a little bit faster than $\Omega(n^{4d-1})$. This is again quite different from the descent shown in Eq. (12) and in other related work using Fourier and Gaussian features (Liao et al., 2020; Jacot et al., 2020), where p only needs to grow proportionally with n.

(ii) Speed of the descent: Since Theorem 1 holds for finite p, it also characterizes the speed of descent. In particular, Term 2 is proportional to $p^{-\frac{1}{2}(1-\frac{1}{q})}$, which approaches $1/\sqrt{p}$ when q is large. Again, such a speed of descent is not captured in (Arora et al., 2019). As we show in Fig. 1(a), the test error of the gradient descent solution under the original neural network (cyan curve) is usually quite close to that of the NTK model (green curve). Thus, our result provides useful guidance on how fast the generalization error descends with p for such neural networks.

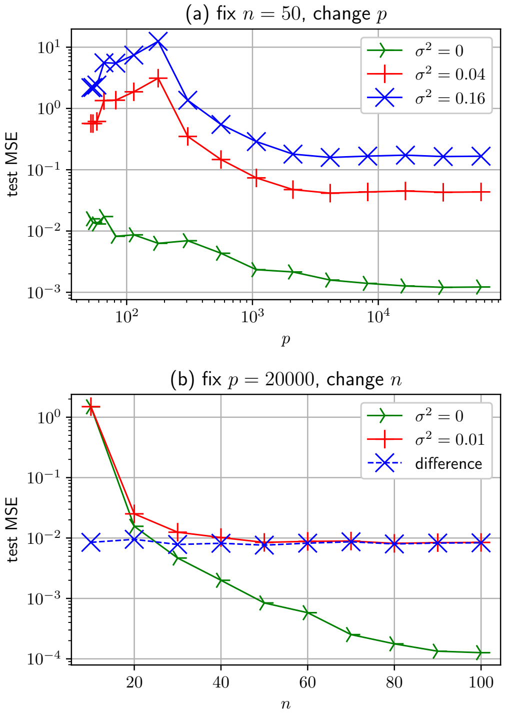

Figure 3. The test MSE of the overfitted NTK model for the same ground-truth function as Fig. 1(a). **(a)** We fix $n = 50$ and increase p for different noise level σ^2. **(b)** We fix $p = 20000$ and increase n. All data points in this figure are the average of five random simulation runs.

(iii) The effect of noise: Term 3 in Eq. (10) characterizes the impact of the noise $\boldsymbol{\epsilon}$, which does not decrease or increase with p. Notice that this is again very different from Eq. (12), i.e., results of min ℓ_2-norm overfitting solutions for simple features, where the noise term $\frac{\sigma^2 n}{p-n-1} \to 0$ when $p \to \infty$. We use Fig. 3(a) to validate this insight. In Fig. 3(a), we fix $n = 50$ and plot curves of test MSE of NTK overfitting solution as p increases. We let the noise ϵ_i in the i-th training sample be *i.i.d.* Gaussian with zero mean and variance σ^2. The green, red, and blue curves in Fig. 3(a) corresponds to the situation $\sigma^2 = 0$, $\sigma^2 = 0.04$, and $\sigma^2 = 0.16$, respectively. We can see that all three curves become flat when p is very large, and this phenomenon implies that the gap across different noise levels does not decrease when $p \to \infty$, which is in contrast to Eq. (12).

In Fig. 3(b), we instead fix $p = 20000$, and increase n). We plot the test MSE both for the noiseless setting (green curve) and for $\sigma^2 = 0.01$ (red curve). The difference between the two curves (dashed blue curve) then captures the impact of noise, which is related to Term 3 in Eq. (10). Somewhat

[5] This curve oscillates at the early stage when p is small. We suspect it is because, at small p, the convergence in Eq. (7) has not occurred yet, and thus the randomness in $\mathbf{V}_0[j]$ makes the simulation results more volatile.

surprisingly, we find that the dashed blue curve is insensitive to n, which suggests that Term 3 in Eq. (10) may have room for improvement.

In summary, we have shown that any ground-truth function in $\mathcal{F}^{\ell_2}$ leads to low generalization error for overfitted NTK models. It is then natural to ask what happens if the ground-truth function is not in $\mathcal{F}^{\ell_2}$. Let $\overline{\mathcal{F}^{\ell_2}}$ denote the closure[6] of $\mathcal{F}^{\ell_2}$, and $D(f, \overline{\mathcal{F}^{\ell_2}})$ denotes the L^2-distance between f and $\mathcal{F}^{\ell_2}$ (i.e., the infimum of the L^2-distance from f to every function in $\mathcal{F}^{\ell_2}$).

Proposition 2. *(i) For any given* $(\mathbf{X}, \boldsymbol{y})$*, there exists a function* $\hat{f}_{\infty}^{\ell_2} \in \mathcal{F}^{\ell_2}$ *such that, uniformly over all* $\boldsymbol{x} \in \mathcal{S}^{d-1}$, $\hat{f}^{\ell_2}(\boldsymbol{x}) \xrightarrow{P} \hat{f}_{\infty}^{\ell_2}(\boldsymbol{x})$ *as* $p \to \infty$*. **(ii)** Consequently, if the ground-truth function* $f \notin \overline{\mathcal{F}^{\ell_2}}$ *(or equivalently,* $D(f, \overline{\mathcal{F}^{\ell_2}}) > 0$*), then the MSE of* $\hat{f}_{\infty}^{\ell_2}$ *(with respect to the ground-truth function* f*) is at least* $D(f, \overline{\mathcal{F}^{\ell_2}})$.

Intuitively, Proposition 2 (proven in Supplementary Material Appendix J) suggests that, if a ground-truth function is outside the closure of $\mathcal{F}^{\ell_2}$, then no matter how large n is, the test error of a NTK model with infinitely many neurons cannot be small (regardless whether or not the training samples contain noise). We validate this in Fig. 1(b), where a ground-truth function is chosen outside $\overline{\mathcal{F}^{\ell_2}}$. The test MSE of NTK overfitting solutions (green curve) is above null risk (dashed black line) and thus is much higher compared with Fig. 1(a). We also plot the test MSE of the GD solution of the real neural network (cyan curve), which seems to show the same trend.

Comparing Theorem 1 and Proposition 2, we can clearly see that, all functions in $\mathcal{F}^{\ell_2}$ are learnable by the overfitted NTK model, and all functions not in $\overline{\mathcal{F}^{\ell_2}}$ are not.

4. What Exactly are the Functions in $\mathcal{F}^{\ell_2}$?

Our expression for learnable functions in Definition 1 is still in an indirect form, i.e., through the unknown function $g(\cdot)$. In (Arora et al., 2019), the authors show that all functions of the form $(\boldsymbol{x}^T\boldsymbol{a})^l$, $l \in \{0, 1, 2, 4, 6, \cdots\}$ are learnable by GD (assuming large p and small step size), for a similar 2-layer network with ReLU activation that has no bias. In the following, we will show that our learnable functions in Definition 1 also have a similar form. Further, we can show that any functions of the form $(\boldsymbol{x}^T\boldsymbol{a})^l$, $l \in \{3, 5, 7, \cdots\}$ are not learnable. Our characterization uses an interesting connection to harmonics and filtering on $\mathcal{S}^{d-1}$, which may be of independent interest.

Towards this end, we first note that the integral form in Definition 1 can be viewed as a convolution on $\mathcal{S}^{d-1}$ (denoted by $\circledast$). Specifically, for any $f_g \in \mathcal{F}^{\ell_2}$, we can rewrite it as

$$f_g(\boldsymbol{x}) = g \circledast h(\boldsymbol{x}) := \int_{\mathsf{SO}(d)} g(\mathbf{S}\boldsymbol{e}) h(\mathbf{S}^{-1}\boldsymbol{x}) d\mathbf{S}, \quad (13)$$

$$h(\boldsymbol{x}) := \boldsymbol{x}^T \boldsymbol{e} \frac{\pi - \arccos(\boldsymbol{x}^T\boldsymbol{e})}{2\pi}, \quad (14)$$

where $\boldsymbol{e} := [0\, 0\, \cdots\, 0\, 1]^T \in \mathbb{R}^d$, and $\mathbf{S}$ is a $d \times d$ orthogonal matrix that denotes a rotation in $\mathcal{S}^{d-1}$, chosen from the set $\mathsf{SO}(d)$ of all rotations. An important property of the convolution Eq. (13) is that it corresponds to multiplication in the frequency domain, similar to Fourier coefficients. To define such a transformation to the frequency domain, we use a set of hyper-spherical harmonics $\Xi_{\mathbf{K}}^l$ (Vilenkin, 1968; Dokmanic & Petrinovic, 2009) when $d \geq 3$, which forms an orthonormal basis for functions on $\mathcal{S}^{d-1}$. These harmonics are indexed by l and $\mathbf{K}$, where $\mathbf{K} = (k_1, k_2, \cdots, k_{d-2})$ and $l = k_0 \geq k_1 \geq k_2 \geq \cdots \geq k_{d-2} \geq 0$ (those k_i's and l are all non-negative integers). Any function $f \in L^2(\mathcal{S}^{d-1} \mapsto \mathbb{R})$ (including even δ-functions (Li & Wong, 2013)) can be decomposed uniquely into these harmonics, i.e., $f(\boldsymbol{x}) = \sum_l \sum_{\mathbf{K}} c_f(l, \mathbf{K}) \Xi_{\mathbf{K}}^l(\boldsymbol{x})$, where $c_f(\cdot, \cdot)$ are projections of f onto the basis function. In Eq. (13), let $c_g(\cdot, \cdot)$ and $c_h(\cdot, \cdot)$ denote the coefficients corresponding to the decompositions of g and h, respectively. Then, we must have (Dokmanic & Petrinovic, 2009)

$$c_{f_g}(l, \mathbf{K}) = \Lambda \cdot c_g(l, \mathbf{K}) c_h(l, \mathbf{0}), \quad (15)$$

where Λ is some normalization constant. Notice that in Eq. (15), the coefficient for h is $c_h(l, \mathbf{0})$ instead of $c_h(l, \mathbf{K})$, which is due to the intrinsic rotational symmetry of such convolution (Dokmanic & Petrinovic, 2009).

The above decomposition has an interesting "filtering" interpretation as follows. We can regard the function h as a "filter" or "channel," while the function g as a transmitted "signal." Then, the function f_g in Eq. (13) and Eq. (15) can be regarded as the received signal after g goes through the channel/filter h. Therefore, when coefficient $c_h(l, \mathbf{0})$ of h is non-zero, then the corresponding coefficient $c_{f_g}(l, \mathbf{K})$ for f_g can be any value (because we can arbitrarily choose g). In contrast, if a coefficient $c_h(l, \mathbf{0})$ of h is zero, then the corresponding coefficient $c_{f_g}(l, \mathbf{K})$ for f_g must also be zero for all $\mathbf{K}$.

Ideally, if h contains all "frequencies," i.e., all coefficients $c_h(l, \mathbf{0})$ are non-zero, then f_g can also contain all "frequencies," which means that $\mathcal{F}^{\ell_2}$ can contain almost all functions. Unfortunately, this is not true for the function h given in Eq. (14). Specifically, using the harmonics defined in (Dokmanic & Petrinovic, 2009), the basis $\Xi_{\mathbf{0}}^l$ for $(l, \mathbf{0})$ turns out to have the form

$$\Xi_{\mathbf{0}}^l(\boldsymbol{x}) = \sum_{k=0}^{\lfloor \frac{l}{2} \rfloor} (-1)^k \cdot a_{l,k} \cdot (\boldsymbol{x}^T\boldsymbol{e})^{l-2k}, \quad (16)$$

[6] We consider the normed space of all functions in $L^2(\mathcal{S}^{d-1} \mapsto \mathbb{R})$. Notice that although $g(\boldsymbol{z})$ in Definition 1 may not be in L^2, f_g is always in L^2. Specifically, $f_g(\boldsymbol{x})$ is bounded for every $\boldsymbol{x} \in \mathcal{S}^{d-1}$ when $\|g\|_1 < \infty$.

where $a_{l,k}$ are positive constants. Note that the expression Eq. (16) contains either only even powers of $\boldsymbol{x}^T\boldsymbol{e}$ (if l is even) or odd powers of $\boldsymbol{x}^T\boldsymbol{e}$ (if l is odd). Then, for the function h in Eq. (14), we have the following proposition (proven in Supplementary Material, Appendix K.4). We note that (Basri et al., 2019) has a similar harmonics analysis, where the expression of $c_h(l, \mathbf{0})$ is given. However, it is not obvious that the expression of $c_h(l, \mathbf{0})$ for all $l = 0, 1, 2, 4, 6, \cdots$ given in (Basri et al., 2019) must be non-zero, which is made clear by Proposition 3 as follows.

Proposition 3. *$c_h(l, \mathbf{0})$ is zero for $l = 3, 5, 7, \cdots$ and is non-zero for $l = 0, 1, 2, 4, 6, \cdots$.*

We are now ready to characterize what functions are in $\mathcal{F}^{\ell_2}$. By the form of Eq. (16), for any non-negative integer k, any even power $(\boldsymbol{x}^T\boldsymbol{e})^{2k}$ is a linear combination of $\Xi_{\mathbf{0}}^0, \Xi_{\mathbf{0}}^2, \cdots, \Xi_{\mathbf{0}}^{2k}$, and any odd power $(\boldsymbol{x}^T\boldsymbol{e})^{2k+1}$ is a linear combination of $\Xi_{\mathbf{0}}^1, \Xi_{\mathbf{0}}^3, \cdots, \Xi_{\mathbf{0}}^{2k+1}$. By Proposition 3, we thus conclude that any function $f_g(\boldsymbol{x}) = (\boldsymbol{x}^T\boldsymbol{e})^l$ where $l \in \{0, 1, 2, 4, 6, \cdots\}$ can be written in the form of Eq. (15) in the frequency domain, and thus are in $\mathcal{F}^{\ell_2}$. In contrast, any function $f(\boldsymbol{x}) = (\boldsymbol{x}^T\boldsymbol{e})^l$ where $l \in \{3, 5, 7, \cdots\}$ cannot be written in the form of Eq. (15), and are thus not in $\mathcal{F}^{\ell_2}$. Further, the ℓ_2-norm of any latter function will also be equal to its distance to $\mathcal{F}^{\infty}$. Therefore, the generalization-error lower-bound in Proposition 2 will apply (with $D(f, \mathcal{F}^{\ell_2}) = \|f\|_2$). Finally, by Eq. (13), $\mathcal{F}^{\ell_2}$ is invariant under rotation and finite linear summation. Therefore, any finite sum of $(\boldsymbol{x}^T\boldsymbol{a})^l$, $l = 0, 1, 2, 4, 6, \cdots$ must also belong to $\mathcal{F}^{\ell_2}$.

For the special case of $d = 2$, the input $\boldsymbol{x}$ corresponds to an angle $\theta \in [-\pi, \pi]$, and the above-mentioned harmonics become Fourier series $\sin(k\theta)$ and $\cos(k\theta)$, $k = 0, 1, \cdots$. We can then get similar results that frequencies of $k \in \{0, 1, 2, 4, 6, \cdots\}$ are learnable (while others are not), which explains the learnable and not-learnable functions in Fig. 1. Details can be found in Supplementary Material, Appendix K.5.

Remark 2. We caution that the above claim on non-learnable functions critically depends on the network architecture. That is, we assume throughout this paper that the ReLU activation has no bias. It is known from an expressiveness point of view that, using ReLU without bias, a shallow network can only approximate the sum of linear functions and even functions (Ghorbani et al., 2019). Thus, it is not surprising that other odd-power (but non-linear) polynomials cannot be learned. In contrast, by adding a bias, a shallow network using ReLU becomes a universal approximator (Ji et al., 2019). The recent work of (Satpathi & Srikant, 2021) shows that polynomials with all powers can be learned by the corresponding 2-layer NTK model. These results are consistent with ours because a ReLU activation function operating on $\tilde{\boldsymbol{x}} \in \mathbb{R}^{d-1}$ with a bias can be equivalently viewed as one operating on a d-dimension input (with the last-dimension being fixed at $1/\sqrt{d}$) but with no bias. Even though only a subset of functions are learnable in the d-dimension space, when projected into a $(d-1)$-dimension subspace, they may already span all functions. For example, one could write $(\boldsymbol{x}^T\boldsymbol{a})^3$ as a linear combination of $(\begin{bmatrix} \tilde{\boldsymbol{x}} \\ 1/\sqrt{d} \end{bmatrix}^T \boldsymbol{b}_i)^{l_i}$, where $i \in \{1, 2, \cdots, 5\}$, $[l_1, \cdots, l_5] = [4, 4, 2, 1, 0]$, and $\boldsymbol{b}_i \in \mathbb{R}^d$ depends only on $\boldsymbol{a}$. (See Supplementary Material, Appendix K.6 for details.) It remains an interesting question whether similar difference arises for other network architectures (e.g., with more than 2 layers).

5. Proof Sketch of Theorem 1

In this section, we sketch the key steps to prove Theorem 1. Starting from Eq. (3), we have

$$\Delta\mathbf{V}^{\ell_2} = \mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}(\mathbf{F}(\mathbf{X}) + \boldsymbol{\epsilon}). \qquad (17)$$

For the learned model $\hat{f}^{\ell_2}(\boldsymbol{x}) = \boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}}\Delta\mathbf{V}^{\ell_2}$ given in Eq. (4), the error for any test input $\boldsymbol{x}$ is then

$$\hat{f}^{\ell_2}(\boldsymbol{x}) - f(\boldsymbol{x}) = \left(\boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}}\mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{F}(\mathbf{X}) - f(\boldsymbol{x})\right) + \boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}}\mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}\boldsymbol{\epsilon}. \qquad (18)$$

In the classical "bias-variance" analysis with respect to MSE (Belkin et al., 2018a), the first term on the right-hand-side of Eq. (18) contributes to the bias and the second term contributes to the variance. We first quantify the second term (i.e., the variance) in the following proposition.

Proposition 4. *For any $n \geq 2$, $m \in \left[1, \frac{\ln n}{\ln \frac{\pi}{2}}\right]$, $d \leq n^4$, if $p \geq 6J_m(n,d)\ln\left(4n^{1+\frac{1}{m}}\right)$, we must have $\Pr_{\mathbf{X},\mathbf{V}_0}\left\{|\boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}}\mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}\boldsymbol{\epsilon}| \leq \sqrt{J_m(n,d)n}\|\boldsymbol{\epsilon}\|_2, \text{ for all } \boldsymbol{\epsilon} \in \mathbb{R}^n\right\} \geq 1 - \frac{2}{\sqrt[m]{n}}$.*

The proof is in Supplementary Material Appendix F. Proposition 4 implies that, for fixed n and d, when $p \to \infty$, with high probability the variance will not exceed a certain factor of the noise $\|\boldsymbol{\epsilon}\|_2$. In other words, the variance will not go to infinity when $p \to \infty$. The main step in the proof is to lower bound $\min \operatorname{eig}\left(\mathbf{H}\mathbf{H}^T\right)/p$, which is given by $1/(J_m(n,d)n)$. Note that this is the main place where we used the assumption that $\boldsymbol{x}$ is uniformly distributed. We expect that our main proof techniques can be generalized to other distributions (with a different expression of $J_m(n,d)$), which we leave for future work.

Remark 3. In the upper bound in (Arora et al., 2019) (i.e., Eq. (11)), any noise added to $\boldsymbol{y}$ will at least contribute to the generalization upper bound Eq. (11) by a positive term $\boldsymbol{\epsilon}^T(\mathbf{H}^\infty)^{-1}\boldsymbol{\epsilon}/n$. Thus, their upper bound may also grow as $\min \operatorname{eig}(\mathbf{H}^\infty)$ decreases. One of the contribution of Proposition 4 is to characterize this minimum eigenvalue.

We now bound the bias part. We first study the class of ground-truth functions that can be learned with fixed $\mathbf{V}_0$. We refer to them as *pseudo ground-truth*, to differentiate them with the set $\mathcal{F}^{\ell_2}$ of learnable functions for random $\mathbf{V}_0$. They are defined with respect to the same $g(\cdot)$ function, so that we can later extend to the "real" ground-truth functions in $\mathcal{F}^{\ell_2}$ when considering the randomness of $\mathbf{V}_0$.

Definition 2. *Given $\mathbf{V}_0$, for any learnable ground-truth function $f_g \in \mathcal{F}^{\ell_2}$ with the corresponding function $g(\cdot)$, define the corresponding* ***pseudo ground-truth*** *as*

$$f^g_{\mathbf{V}_0}(\boldsymbol{x}) := \int_{\mathcal{S}^{d-1}} \boldsymbol{x}^T \boldsymbol{z} \frac{|\mathcal{C}^{\mathbf{V}_0}_{\boldsymbol{z},\boldsymbol{x}}|}{p} g(\boldsymbol{z}) d\mu(\boldsymbol{z}).$$

The reason that this class of functions may be the learnable functions for fixed $\mathbf{V}_0$ is similar to the discussions in Eq. (5) and Eq. (6). Indeed, using the same choice of $g(\boldsymbol{z})$ in Eq. (8), the learned function $\hat{f}^{\ell_2}$ in Eq. (5) at fixed $\mathbf{V}_0$ is always of the form in Definition 2.

The following proposition gives an upper bound of the generalization performance when the data model is based on the pseudo ground-truth and the NTK model uses exactly the same $\mathbf{V}_0$.

Proposition 5. *Assume fixed $\mathbf{V}_0$ (thus p and d are also fixed), there is no noise. If the ground-truth function is $f = f^g_{\mathbf{V}_0}$ in Definition 2 and $\|g\|_\infty < \infty$, then for any $\boldsymbol{x} \in \mathcal{S}^{d-1}$ and $q \in [1,\ \infty)$, we have $\mathsf{Pr}_{\mathbf{X}} \left\{ |\hat{f}^{\ell_2}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq n^{-\frac{1}{2}\left(1-\frac{1}{q}\right)} \right\} \geq 1 - 2e^2 \exp\left(-\frac{\sqrt[q]{n}}{8\|g\|_\infty^2}\right)$.*

The proof is in Supplementary Material, Appendix H. Note that both the threshold of the probability event and the upper bound coincide with Term 1 and Term 4, respectively, in Eq. (10). Here we sketch the proof of Proposition 5. Based on the definition of the pseudo ground-truth, we can rewrite $f^g_{\mathbf{V}_0}$ as $f^g_{\mathbf{V}_0}(\boldsymbol{x}) = \boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}} \Delta\mathbf{V}^*$, where $\Delta\mathbf{V}^* \in \mathbb{R}^{dp}$ is given by, for all $j \in \{1, 2, \cdots, p\}$, $\Delta\mathbf{V}^*[j] = \int_{\mathcal{S}^{d-1}} \mathbf{1}_{\{\boldsymbol{z}^T \mathbf{V}_0[j] > 0\}} \boldsymbol{z} \frac{g(\boldsymbol{z})}{p} d\mu(\boldsymbol{z})$. From Eq. (3) and Eq. (4), we can see that the learned model is $\hat{f}^{\ell_2}(\boldsymbol{x}) = \boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}} \mathbf{P} \Delta\mathbf{V}^*$ where $\mathbf{P} := \mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{H}$. Note that $\mathbf{P}$ is an orthogonal projection to the row-space of $\mathbf{H}$. Further, it is easy to show that $\|\boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}}\|_2 \leq \sqrt{p}$. Thus, we have $|\hat{f}^{\ell_2}(\boldsymbol{x}) - f^g_{\mathbf{V}_0}(\boldsymbol{x})| = |\boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}}(\mathbf{P} - \mathbf{I})\Delta\mathbf{V}^*| \leq \sqrt{p}\|(\mathbf{P} - \mathbf{I})\Delta\mathbf{V}^*\|_2$. The term $(\mathbf{P} - \mathbf{I})\Delta\mathbf{V}^*$ can be interpreted as the distance from $\Delta\mathbf{V}^*$ to the row-space of $\mathbf{H}$. Note that this distance is no greater than the distance between $\Delta\mathbf{V}^*$ and any point in the row-space of $\mathbf{H}$. Thus, in order to get an upper bound on $\|(\mathbf{P} - \mathbf{I})\Delta\mathbf{V}^*\|_2$, we only need to find a vector $\boldsymbol{a} \in \mathbb{R}^n$ that makes $\|\Delta\mathbf{V}^* - \mathbf{H}^T\boldsymbol{a}\|_2$ as small as possible, especially when n is large. Our proof uses the vector $\boldsymbol{a}$ such that its i-th element is $\boldsymbol{a}_i := \frac{g(\mathbf{X}_i)}{np}$. See Supplementary Material, Appendix H for the rest of the details.

The final step is to allow $\mathbf{V}_0$ to be random. Given any random $\mathbf{V}_0$, any function $f_g \in \mathcal{F}^{\ell_2}$ can be viewed as the summation of a pseudo ground-truth function (with the same $g(\cdot)$) and a difference term. This difference can be viewed as a special form of "noise", and thus we can use Proposition 4 to quantify its impact. Further, the magnitude of this "noise" should decrease with p (because of Eq. (7)). Combining this argument with Proposition 5, we can then prove Theorem 1. See Supplementary Material, Appendix I for details.

6. Conclusions

In this paper, we studied the generalization performance of the min ℓ_2-norm overfitting solution for a two-layer NTK model. We provide a precise characterization of the learnable ground-truth functions for such models, by providing a generalization upper bound for all functions in $\overline{\mathcal{F}^{\ell_2}}$, and a generalization lower bound for all functions not in $\overline{\mathcal{F}^{\ell_2}}$. We show that, while the test error of the overfitted NTK model also exhibits descent in the overparameterized regime, the descent behavior can be quite different from the double descent of linear models with simple features.

There are several interesting directions for future work. First, based on Fig. 3(b), our estimation of the effect of noise could be further improved. Second, it would be interesting to explore whether the methodology can be extended to NTK model for other neural networks, e.g., with different activation functions and with more than two layers.

Acknowledgements

This work is partially supported by an NSF sub-award via Duke University (IIS-1932630), by NSF grants CNS-1717493, CNS-1901057, and CNS-2007231, and by Office of Naval Research under Grant N00014-17-1-241. The authors would like to thank Professor R. Srikant at the University of Illinois at Urbana-Champaign and anonymous reviewers for their valuable comments and suggestions.

References

Advani, M. S., Saxe, A. M., and Sompolinsky, H. High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 132:428–446, 2020.

Allen-Zhu, Z., Li, Y., and Liang, Y. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in neural information processing systems*, pp. 6158–6169, 2019.

Arora, S., Du, S., Hu, W., Li, Z., and Wang, R. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pp. 322–332, 2019.

Bartlett, P. L., Long, P. M., Lugosi, G., and Tsigler, A. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 2020.

Basri, R., Jacobs, D., Kasten, Y., and Kritchman, S. The convergence rate of neural networks for learned functions of different frequencies. *arXiv preprint arXiv:1906.00425*, 2019.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. Reconciling modern machine learning and the bias-variance trade-off. *arXiv preprint arXiv:1812.11118*, 2018a.

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

Belkin, M., Hsu, D., and Xu, J. Two models of double descent for weak features. *arXiv preprint arXiv:1903.07571*, 2019.

Bishop, C. M. *Pattern recognition and machine learning*. Springer, 2006.

Chaudhry, M. A., Qadir, A., Rafique, M., and Zubair, S. Extension of euler's beta function. *Journal of computational and applied mathematics*, 78(1):19–32, 1997.

Dokmanic, I. and Petrinovic, D. Convolution on the n-sphere with application to pdf modeling. *IEEE transactions on signal processing*, 58(3):1157–1170, 2009.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2018.

Dutka, J. The incomplete beta function—a historical profile. *Archive for history of exact sciences*, pp. 11–29, 1981.

d'Ascoli, S., Refinetti, M., Biroli, G., and Krzakala, F. Double trouble in double descent: Bias and variance (s) in the lazy regime. In *International Conference on Machine Learning*, pp. 2280–2290. PMLR, 2020.

Fiat, J., Malach, E., and Shalev-Shwartz, S. Decoupling gating from linearity. *arXiv preprint arXiv:1906.05032*, 2019.

Ghorbani, B., Mei, S., Misiakiewicz, T., and Montanari, A. Linearized two-layers neural networks in high dimension. *arXiv preprint arXiv:1904.12191*, 2019.

Goemans, M. Chernoff bounds, and some applications. *URL http: //math.mit.edu /goemans /18310S15 /chernoff-notes.pdf*, 2015.

Hastie, T., Tibshirani, R., and Friedman, J. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

Hastie, T., Montanari, A., Rosset, S., and Tibshirani, R. J. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*, 2019.

Hayes, T. P. A large-deviation inequality for vector-valued martingales. *Combinatorics, Probability and Computing*, 2005.

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.

Jacot, A., Simsek, B., Spadaro, F., Hongler, C., and Gabriel, F. Implicit regularization of random feature models. In *International Conference on Machine Learning*, pp. 4631–4640. PMLR, 2020.

James, W. and Stein, C. Estimation with quadratic loss. In *Breakthroughs in Statistics*, pp. 443–460. Springer, 1992.

Ji, Z. and Telgarsky, M. Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow relu networks. *arXiv preprint arXiv:1909.12292*, 2019.

Ji, Z., Telgarsky, M., and Xian, R. Neural tangent kernels, transportation mappings, and universal approximation. *arXiv preprint arXiv:1910.06956*, 2019.

Ju, P., Lin, X., and Liu, J. Overfitting can be harmless for basis pursuit, but only to a degree. *Advances in Neural Information Processing Systems*, 33, 2020.

LeCun, Y., Kanter, I., and Solla, S. A. Second order properties of error surfaces: Learning time and generalization. In *Advances in Neural Information Processing Systems*, pp. 918–924, 1991.

Li, S. Concise formulas for the area and volume of a hyperspherical cap. *Asian Journal of Mathematics and Statistics*, 4(1):66–70, 2011.

Li, Y. and Liang, Y. Learning overparameterized neural networks via stochastic gradient descent on structured data. *Advances in Neural Information Processing Systems*, 31: 8157–8166, 2018.

Li, Y. and Wong, R. Integral and series representations of the dirac delta function. *arXiv preprint arXiv:1303.1943*, 2013.

Liao, Z., Couillet, R., and Mahoney, M. W. A random matrix analysis of random fourier features: beyond the gaussian kernel, a precise phase transition, and the corresponding double descent. *arXiv preprint arXiv:2006.05013*, 2020.

Mei, S. and Montanari, A. The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv preprint arXiv:1908.05355*, 2019.

Muthukumar, V., Vodrahalli, K., and Sahai, A. Harmless interpolation of noisy data in regression. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 2299–2303. IEEE, 2019.

Rao, K. B. and Rao, M. B. *Theory of charges: a study of finitely additive measures*. Academic Press, 1983.

Satpathi, S. and Srikant, R. The dynamics of gradient descent for overparametrized neural networks. In *3rd Annual Learning for Dynamics and Control Conference (L4DC)*, 2021.

Stein, C. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. Technical report, Stanford University Stanford United States, 1956.

Tikhonov, A. N. On the stability of inverse problems. In *Dokl. Akad. Nauk SSSR*, volume 39, pp. 195–198, 1943.

Vilenkin, N. Y. Special functions and the theory of group representations. providence: American mathematical society. *sftp*, 1968.

Wainwright, M. Uniform laws of large numbers, 2015. https://www.stat.berkeley.edu/~mjwain/stat210b/Chap4_Uniform_Feb4_2015.pdf, Accessed: Feb. 7, 2021.

Wendel, J. G. A problem in geometric probability. *Math. Scand*, 11:109–111, 1962.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017*, 2017.