Spectral Bias Outside the Training Set for Deep Networks in the Kernel Regime

Benjamin Bowman
UCLA Department of Mathematics
benbowman314@math.ucla.edu

Guido Montúfar
UCLA Departments of Mathematics and Statistics and MPI MIS
montufar@math.ucla.edu

Abstract

We provide quantitative bounds measuring the L^2 difference in function space between the trajectory of a finite-width network trained on finitely many samples from the idealized kernel dynamics of infinite width and infinite data. An implication of the bounds is that the network is biased to learn the top eigenfunctions of the Neural Tangent Kernel not just on the training set but over the entire input space. This bias depends on the model architecture and input distribution alone and thus does not depend on the target function which does not need to be in the RKHS of the kernel. The result is valid for deep architectures with fully connected, convolutional, and residual layers. Furthermore the width does not need to grow polynomially with the number of samples in order to obtain high probability bounds up to a stopping time. The proof exploits the low-effective-rank property of the Fisher Information Matrix at initialization, which implies a low effective dimension of the model (far smaller than the number of parameters). We conclude that local capacity control from the low effective rank of the Fisher Information Matrix is still underexplored theoretically.

1 Introduction

Training heavily overparameterized networks via gradient based optimization has become standard operating procedure in deep learning. Overparameterized networks are able to interpolate arbitrary labels both in principle and in practice (Zhang et al., 2017), rendering classical PAC learning theory insufficient to explain the generalization of networks within this modality. The high capacity of modern networks ensures that there are both good and bad empirical risk minimizers. Miraculously the network preferentially chooses the good solutions and sidesteps those that are unfavorable, posing a challenge and opportunity to today's researchers.

The success of overparameterized networks has prompted the theoretical community to search for more subtle forms of capacity control (Neyshabur et al., 2015, 2017; Gunasekar et al., 2017). The contemporary point-of-view is that the data distribution, model parameterization, and optimization algorithm are all relevant in limiting complexity. This has led to a variety of efforts to characterize the properties that networks and related models are biased towards when optimized via gradient descent. Examples include max-margin bias for classification problems (Soudry et al., 2018; Ji & Telgarsky, 2019; Nacson et al., 2019; Gunasekar et al., 2018), minimum nuclear norm bias for matrix factorization (Gunasekar et al., 2017; Li et al., 2018; Gunasekar et al., 2018), and minimum RKHS norm bias in the kernel regime (Zhang et al., 2020).

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

https://doi.org/10.52202/068431-2201

Empirically it is known that neural networks tend to learn low Fourier frequencies first and add higher frequencies only later in training (Rahaman et al., 2019; Xu et al., 2019; Yang et al., 2022), the phenomenon that has been titled "Spectral Bias" or the "Frequency Principle". Theoretical justifications of this have been proposed by studying networks in the kernel regime. For shallow univariate ReLU networks Basri et al. (2019, 2020) demonstrate that the dominant eigenfunctions of the Neural Tangent Kernel (NTK) (Jacot et al., 2018) correspond to the low Fourier frequencies for the uniform distribution and more generally to smoother components for nonuniform distributions. This echos the results by Williams et al. (2019) and Jin & Montúfar (2021) that show that univariate ReLU networks in the kernel regime are biased towards smooth interpolants. Abstracting away from Fourier frequencies, "Spectral Bias" can be interpreted more broadly to mean bias towards learning the top eigenfunctions of the Neural Tangent Kernel. By looking at empirical approximations to the eigenfunctions, spectral bias was demonstrated to hold on the training set by Arora et al. (2019a), Basri et al. (2020), and Cao et al. (2021). A recent work by Bowman & Montúfar (2022) was able to demonstrate that spectral bias holds off the training set for shallow feedforward networks when the network is underparameterized. In the present work we exploit the low-effective-rank property of the Fisher Information Matrix and are able to demonstrate that spectral bias holds outside the training set without the underparameterization requirement. In fact the number of samples can be on the same order as the width of the network. Furthermore, by leveraging a recent work by Liu et al. (2020b) bounding the Hessian of wide networks, our result permits deep networks with fully connected, convolutional, and residual layers. Consequently we are able to conclude that spectral bias holds for more realistic sample complexities and diverse architectures.

1.1 Our Contributions

- We provide quantitative bounds measuring the L^2 difference in function space between the trajectory of a finite-width network trained on finitely many samples from the idealized kernel dynamics of infinite width and infinite data (see Theorem 3.5 and Corollary 3.7).
- As an implication of these bounds, eigenfunctions of the NTK integral operator (not just their empirical approximations) are learned at rates corresponding to their eigenvalues (see Corollary 3.7 and Observation 3.8).
- We demonstrate that the network will inherit the bias of the kernel at the beginning of training even when the width only grows linearly with the number of samples (see Observation 3.9).

1.2 Related Work

NTK Convergence Results The NTK was introduced by Jacot et al. (2018) while almost concurrently Du et al. (2019b) used it implicitly to prove a global convergence guarantee for gradient descent applied to a shallow ReLU network. These two highly charismatic works led to a flurry of subsequent works, of which we can only hope to provide a partial list. Global convergence for arbitrary labels was addressed in a series of works (Du et al., 2019b,a; Oymak & Soltanolkotabi, 2020; Allen-Zhu et al., 2019; Nguyen & Mondelli, 2020; Nguyen, 2021; Zou et al., 2020; Zou & Gu, 2019). For arbitrary labels to our knowledge all works require the network width to either grow polynomially with the number of samples n or the inverse desired accuracy ϵ^{-1}. If one assumes the target function aligns with the NTK model, for shallow networks this can be reduced to polylogarithmic width for the logistic loss (Ji & Telgarsky, 2020) or linear width for the squared loss (E et al., 2020; Su & Yang, 2019; Bowman & Montúfar, 2022).

Spectrum of the NTK/Hessian and Generalization The fact that the NTK tends to have a small number of large outlier eigenvalues has been observed in many works (e.g. Arora et al. 2019a; Oymak et al. 2020; Li et al. 2020). Papyan (2020) demonstrated that for classification problems the logit gradients cluster within classes, which produces outliers in the spectra of the NTK and the Hessian of the loss. There have been a series of works analyzing the NTK/Hessian spectrum theoretically using random matrix theory and other tools (e.g. Karakida et al. 2021; Pennington & Worah 2018; Pennington & Bahri 2017; Fan & Wang 2020; Yang & Salman 2019). Recently the spectrum of the NTK integral operator for ReLU networks has been shown to asymptotically follow a power law (Velikanov & Yarotsky, 2021). Arora et al. (2019a) provided a generalization bound that is effective when the labels align with the top eigenvectors of the NTK. Oymak et al. (2020) were able to use the low effective rank of the NTK to obtain generalization bounds, and Li et al. (2020) used the same property to demonstrate robustness to label noise. The low effective rank of the Hessian has also been

https://doi.org/10.52202/068431-2201

incorporated into PAC-Bayes bounds, most recently by Yang et al. (2021). Interestingly, the notion of the effective dimension they define is essentially the same quantity we use to bound the model complexity of the network's linearization.

NTK Eigenvector and Eigenfunction Convergence Rates Luo et al. (2020) explicitly tracked the dynamics of the infinite width shallow model in the Fourier domain. Arora et al. (2019a) demonstrated that when training the hidden layer of a shallow ReLU network, the residual error on the training set projected along eigenvectors of the NTK Gram matrix decays linearly at rates corresponding to the eigenvalues. Cao et al. (2021) proved a similar statement for training both layers, and Basri et al. (2020) proved the analogous statement for a deep fully connected ReLU network where the first and last layer are fixed. Our result can be viewed as the corresponding statement for the test residual instead of the empirical residual: projections of the test residual along eigen*functions* of the NTK *integral operator* are learned at rates corresponding to their eigenvalues. This was shown in a recent work (Bowman & Montúfar, 2022) for shallow fully connected networks that are underparameterized. By contrast our result does not require the network to be underparameterized, and holds for deep networks with fully connected, convolutional, and residual layers. We view our fundamental contribution as demonstrating that spectral bias holds with more realistic sample complexities and in considerable generality with respect to model architecture.

2 Preliminaries

2.1 Notation

Vectors $v \in \mathbb{R}^k$ will be column vectors by default. We will let $\langle \bullet, \bullet \rangle$ and $\|\bullet\|_2$ denote the Euclidean inner product and norm. We define $\langle \bullet, \bullet \rangle_{\mathbb{R}^n} = \frac{1}{n}\langle \bullet, \bullet \rangle$ and $\|\bullet\|_{\mathbb{R}^n} := \sqrt{\langle \bullet, \bullet \rangle_{\mathbb{R}^n}}$ to be the normalized Euclidean inner product and norm. The notation $\overline{B}(v, r) := \{w : \|w - v\|_2 \leq r\}$ will denote the *closed* Euclidean ball centered at v of radius r. $\|A\|_{op} := \sup_{\|v\|_2=1} \|Av\|_2$ will denote the operator norm for matrices. For a symmetric matrix $A \in \mathbb{R}^{k\times k}$, $\lambda_i(A)$ denotes its i-th largest eigenvalue, i.e. $\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_k(A)$. For a set A we will let $|A|$ denote its cardinality. For a natural number $k \geq 1$, we will let $[k] := \{1, \ldots, k\}$. We will let $L^p(X, \nu)$ denote the L^p space over domain X with measure ν. We will denote the inner product associated with $L^2(X, \nu)$ as $\langle \bullet, \bullet \rangle_\nu$. We will use the standard big O and Ω notation with $\tilde{O}$ and $\tilde{\Omega}$ hiding logarithmic terms.

2.2 NTK Dynamics

Let $f(x; \theta)$ be our scalar-valued neural network model taking inputs $x \in X \subset \mathbb{R}^d$ parameterized by $\theta \in \mathbb{R}^p$. For now we will not specify a specific architecture. Our training data will be n input-label pairs $\{(x_1, y_1), \ldots, (x_n, y_n)\} \subset \mathbb{R}^d \times \mathbb{R}$ where we assume that the labels y_i are generated from a fixed scalar-valued target function f^*, namely $f^*(x_i) = y_i$. We will let $y \in \mathbb{R}^n$ denote the label vector $y = (y_1, \ldots, y_n)^T$. Let $\hat{r}(\theta) \in \mathbb{R}^n$ denote the vector that measures the residual error on the training set, whose i-th entry is $\hat{r}(\theta)_i := f(x_i; \theta) - y_i$. We will optimize the squared loss

$$\Phi(\theta) := \frac{1}{2n} \|\hat{r}(\theta)\|_2^2 = \frac{1}{2} \|\hat{r}(\theta)\|_{\mathbb{R}^n}^2$$

via gradient flow

$$\partial_t \theta_t = -\partial_\theta \Phi(\theta),$$

which is the continuous time analog of gradient descent. For conciseness we will denote $\hat{r}(\theta_t)$ by $\hat{r}_t$ and let $r_t(x) := f(x; \theta_t) - f^*(x)$ denote the residual for an arbitrary input x not necessarily in the training set. We may also write $r(x; \theta) := f(x; \theta) - f^*(x)$ for the residual for an arbitrary θ.

We recall some key definitions and facts about the NTK. For a comprehensive introduction we refer the reader to Jacot et al. (2018). We recall the definition of the analytical NTK

$$K^\infty(x, x') := \mathbb{E}_{\theta_0 \sim \mu} \left[\langle \nabla_\theta f(x; \theta_0), \nabla_\theta f(x'; \theta_0) \rangle \right],$$

where the expectation is taken over the parameter initialization $\theta_0 \sim \mu$. The kernel K^∞ induces an integral operator $T_{K^\infty} : L^2(X, \rho) \to L^2(X, \rho)$

$$T_{K^\infty} g(x) := \int_X K^\infty(x, s) g(s) d\rho(s), \tag{1}$$

where X is our input space and ρ is the input distribution. We assume our training inputs $x_1, \ldots, x_n$ are i.i.d. samples from ρ. More generally, for a continuous kernel $K(x, x')$ we define $T_K : L^2(X, \rho) \to L^2(X, \rho)$

$$T_K g(x) := \int_X K(x, s) g(s) d\rho(s). \tag{2}$$

Returning back to K^∞, by Mercer's theorem we have the decomposition

$$K^\infty(x, x') = \sum_{i=1}^{\infty} \sigma_i \phi_i(x) \phi_i(x'),$$

where $\{\phi_i\}$ is an orthonormal basis for $L^2(X, \rho)$ and $\{\sigma_i\}$ is a nonincreasing sequence of positive values. We will see that the bias at the beginning of training within our framework can be described entirely through the operator T_{K^∞} and its eigenfunctions. We note that T_{K^∞} depends only on the model architecture, parameter initialization distribution μ, and input distribution ρ. The training data sample $x_1, \ldots, x_n$ introduces a discretization of the operator T_{K^∞}

$$T_n g(x) := \frac{1}{n} \sum_{i=1}^{n} K^\infty(x, x_i) g(x_i) = \int_X K^\infty(x, s) g(s) d\widehat{\rho}(s), \tag{3}$$

where $\widehat{\rho} = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}$ is the empirical measure. We now introduce the time-dependent NTK

$$K_t(x, x') := \langle \nabla_\theta f(x; \theta_t), \nabla_\theta f(x'; \theta_t) \rangle$$

with the associated time-dependent operator T_n^t

$$T_n^t g(x) := \frac{1}{n} \sum_{i=1}^{n} K_t(x, x_i) g(x_i) = \int_X K_t(x, s) g(s) d\widehat{\rho}(s). \tag{4}$$

The update rule for the residual r_t under gradient flow is given by

$$\partial_t r_t(x) = -\frac{1}{n} \sum_{i=1}^{n} K_t(x, x_i) r_t(x_i) = -T_n^t r_t.$$

Speaking loosely, as the network width tends to infinity the time-dependent NTK $K_t(x, x')$ becomes constant so that $K_t(x, x') = K^\infty(x, x')$ uniformly in t. If $K_t = K^\infty$ then we have the operator equality $T_n^t = T_n$. Similarly, heuristically as $n \to \infty$ we have $T_n \to T_{K^\infty}$. Thus in the idealized infinite width, infinite data limit the update rule becomes

$$\partial_t r_t = -T_{K^\infty} r_t,$$

which has the solution $r_t = \exp(-T_{K^\infty} t) r_0$ which is defined via its projections

$$\langle r_t, \phi_i \rangle_\rho = \exp(-\sigma_i t) \langle r_0, \phi_i \rangle_\rho.$$

Thus in this idealized setting the network learns eigenfunctions ϕ_i at rates determined by their eigenvalues σ_i. The dependence of the convergence rate on the magnitude of σ_i is particularly relevant as the NTK tends to have a very skewed spectrum. We can estimate the spectrum of K^∞ by randomly initializing a network and computing the Gram matrix $(G_0)_{i,j} := K_0(x_i, x_j)$. In Figure 1 we plot the spectrum of the NTK Gram Matrix $(G_0)_{i,j} := K_0(x_i, x_j)$ at initialization. We observe a small number of outlier eigenvalues of large magnitude followed by a long tail of small eigenvalues. This phenomenon has appeared in many works (e.g. Arora et al. 2019a; Oymak et al. 2020; Li et al. 2020). For ReLU networks the spectrum is known to asymptotically follow a power law $\sigma_i \sim \Lambda i^{-\nu}$ (Velikanov & Yarotsky, 2021). The goal of this work is to quantify the extent to which a finite-width network trained on finitely many samples behaves like the idealized kernel dynamics $r_t = \exp(-T_{K^\infty} t) r_0$ corresponding to infinite width and infinite data.

2.3 Applicable Architectures

We now specify an architecture for our model $f(x; \theta)$. We consider deep networks of the form

$$\begin{aligned} \alpha^{(0)} &:= x, \\ \alpha^{(l)} &:= \psi_l(\theta^{(l)}, \alpha^{(l-1)}), \quad l \in [L], \\ f(x; \theta) &:= \frac{1}{\sqrt{m_L}} v^T \alpha^{(L)}, \end{aligned}$$

https://doi.org/10.52202/068431-2201

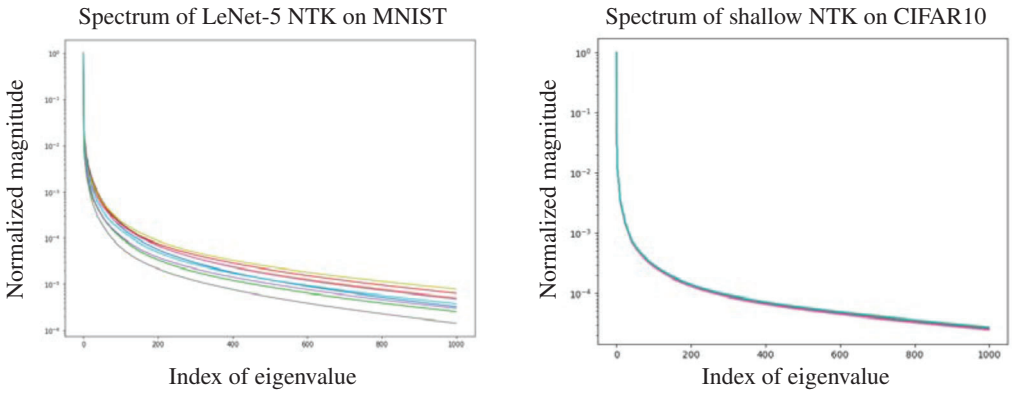

Figure 1: We plot the NTK spectrum on MNIST and CIFAR10 for two networks using 10 random parameter initializations and data batches. In both plots the x-axis represents the eigenvalue index k (linear scale) and the y-axis the normalized eigenvalue λ_k/λ_1 magnitude (log scale). To avoid numerical issues, we compute the NTK on a batch of size 2000 and plot the first 1000 eigenvalues. The left plot computed the NTK corresponding to the logit of class 0 for LeNet-5 on MNIST. The right plot is for a shallow fully-connected softplus network with 4000 hidden units on CIFAR10.

where each $\psi_l(\theta^{(l)}, \bullet) : \mathbb{R}^{m_{l-1}} \to \mathbb{R}^{m_l}$ is a vector-valued function parameterized by $\theta^{(l)} \in \mathbb{R}^{p_l}$ and $v \in \mathbb{R}^{m_L}$. We define $\theta^{(L+1)} := v$ and set $\theta := ((\theta^{(1)})^T, \ldots, (\theta^{(L+1)})^T)^T$ to be the collection of all parameters. We assume each layer mapping ψ_l has one of the following forms:

$$\text{Fully Connected} : \psi_l(\theta^{(l)}, \alpha^{(l-1)}) = \omega\left(\frac{1}{\sqrt{m_{l-1}}} W^{(l)} \alpha^{(l-1)}\right)$$

$$\text{Convolutional} : \psi_l(\theta^{(l)}, \alpha^{(l-1)}) = \omega\left(\frac{1}{\sqrt{m_{l-1}}} W^{(l)} * \alpha^{(l-1)}\right)$$

$$\text{Residual} : \psi_l(\theta^{(l)}, \alpha^{(l-1)}) = \omega\left(\frac{1}{\sqrt{m_{l-1}}} W^{(l)} \alpha^{(l-1)}\right) + \alpha^{(l-1)}$$

Here $\theta^{(l)} = vec(W^{(l)})$ and ω is a twice continuously differentiable function such that ω and ω' are Lipschitz. All parameters of the network will be trained as in practice. For feedforward and residual layers $W^{(l)} \in \mathbb{R}^{m_l \times m_{l-1}}$ is a matrix. For the case of convolutional layers $W^{(l)} \in \mathbb{R}^{K \times m_l \times m_{l-1}}$ is an order-3 tensor with filter size K. The precise definition of the convolution $*$ is offered in the appendix. We will let $m = \min_l m_l$ denote the minimum width of the network. We will assume that $\max_l \frac{m_l}{m} = O(1)$. The input dimension $d := m_0$, the depth L, and the filter sizes K of convolutional layers will be treated as constant. The depth L being constant is essential for NTK convergence; see Hanin & Nica (2020) for an explanation of failure modes whenever depth is nonconstant.

We will now discuss our initialization scheme. We will perform the antisymmetric initialization trick introduced by Zhang et al. (2020) so that the model is identically zero at initialization $f(\bullet; \theta_0) \equiv 0$. Let $f(x; \theta)$ be any neural network of the form described above. Then let $\tilde{\theta} = \left[\begin{smallmatrix} \theta \\ \theta' \end{smallmatrix}\right]$ where $\theta, \theta' \in \mathbb{R}^p$. We then define

$$f_{ASI}(x; \tilde{\theta}) := \frac{1}{\sqrt{2}} f(x; \theta) - \frac{1}{\sqrt{2}} f(x; \theta')$$

which takes the difference of two rescaled copies of our original model $f(x; \theta)$ with parameters θ and θ' that are optimized freely. The antisymmetric initialization trick initializes $\theta_0 \sim N(0, I)$ then sets $\tilde{\theta}_0 = \left[\begin{smallmatrix} \theta_0 \\ \theta_0 \end{smallmatrix}\right]$. We then optimize the model f_{ASI} starting from the initialization $\tilde{\theta}_0$. This trick simultaneously ensures that the model is identically zero at initialization without changing the NTK at initialization (Zhang et al., 2020). For ease of notation we will simply assume from now on that $f(x; \theta) = f_{ASI}(x; \theta)$ and not write the subscript ASI.

https://doi.org/10.52202/068431-2201

3 Main Results

Before stating our main result, we enumerate our key assumptions for the sake of clarity, assumed to hold throughout. Detailed proofs are deferred to the appendix.

Assumption 3.1. *The activation ω is twice continuously differentiable and ω and ω' are Lipschitz.*

Assumption 3.2. *The input domain X is compact with strictly positive Borel measure ρ.*

Assumption 3.3. *The target function f^* satisfies $\|f^*\|_{L^\infty(X,\rho)} = O(1)$.*

Assumption 3.4. *We use the antisymmetric initilization trick so that $f(\bullet;\theta_0) \equiv 0$.*

Most activation functions except for ReLU satisfy Assumption 3.1, such as Softplus $\omega(x) = \ln(1 + e^x)$, Sigmoid $\omega(x) = \frac{1}{1+e^{-x}}$, and Tanh $\omega(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. Assumption 3.2 is a sufficient condition for Mercer's Theorem to hold. While Mercer's theorem is often assumed to hold implicitly, we prefer to make this assumption explicit. Assumption 3.3 simply means the target function is bounded. We believe the antisymmetric initialization specified in Assumption 3.4 is not strictly necessary but it greatly simplifies the proofs and associated bounds. To sidestep 3.4 one would utilize high probability bounds on the magnitude $|f(x;\theta_0)|$ at initialization. In the following results $f(x;\theta)$ will be any of the architectures discussed in Section 2.3. We are now ready to introduce the main result.

Theorem 3.5. *Let $T \geq 1, \epsilon > 0$. Let $K(x, x')$ be a fixed continuous, symmetric, positive definite kernel. For $k \in \mathbb{N}$ let $P_k : L^2(X,\rho) \to L^2(X,\rho)$ denote the orthogonal projection onto the span of the top k eigenfunctions of the operator T_K defined in Equation (2). Let $\sigma_k > 0$ denote the k-th eigenvalue of T_K. Then $m = \tilde{\Omega}(T^4/\epsilon^2)$ and $n = \tilde{\Omega}(T^2/\epsilon^2)$ suffices to ensure with probability at least $1 - O(mn)\exp(-\Omega(\log^2(m))$ over the parameter initilization θ_0 and the training samples $x_1, \ldots, x_n$ that for all $t \leq T$ and $k \in \mathbb{N}$*

$$\|P_k(r_t - \exp(-T_K t)r_0)\|^2_{L^2(X,\rho)} \leq \left[\frac{1 - \exp(-\sigma_k t)}{\sigma_k}\right]^2 \cdot \left[4\|f^*\|^2_\infty \|K - K_0\|^2_{L^2(X^2,\rho\otimes\rho)} + \epsilon\right]$$

and

$$\|r_t - \exp(-T_K t)r_0\|^2_{L^2(X,\rho)} \leq t^2 \cdot \left[4\|f^*\|^2_\infty \|K - K_0\|^2_{L^2(X^2,\rho\otimes\rho)} + \epsilon\right].$$

3.1 Interpretation and Consequences

Theorem 3.5 compares the dynamics of the residual $r_t(x) := f(x;\theta_t) - f^*(x)$ of our finite-width model trained on finitely many samples to the idealized dynamics of a kernel method $\exp(-T_K t)r_0$ with infinite data. We recall that if ϕ_i is an eigenfunction of T_K with eigenvalue σ_i then $\langle \exp(-T_K t)r_0, \phi_i\rangle_\rho = \exp(-\sigma_i t)\langle r_0, \phi_i\rangle_\rho$. Thus the term $\exp(-T_K t)r_0$ learns the projection along eigenfunction ϕ_i linearly at rate σ_i. Whenever the NTK at initialization K_0 concentrates around K, the residual r_t will inherit this bias of the kernel dynamics $\exp(-T_K t)r_0$. Furthermore, the bound for the projected difference $\|P_k(r_t - \exp(-T_K t)r_0)\|^2_{L^2(X,\rho)}$ is smaller whenever σ_k is large. Therefore the bias appears more pronounced along eigendirections with large eigenvalues.

Consequences for the special case $K = K^\infty$ In the infinite width limit, we have that K_0 approaches K^∞ for general architectures (Yang, 2020). For fixed x, x', by concentration results the typical rate of convergence is $|K_0(x,x') - K^\infty(x,x')| = \tilde{O}(1/\sqrt{m})$ with high probability (Du et al., 2019b,a; Huang & Yau, 2020). Bounds that hold uniformly over x, x' of the same rate were provided by Bowman & Montúfar (2022) and Buchanan et al. (2021). A more pessimistic estimate of $1/m^{1/4}$ is provided by Arora et al. (2019b). Even if the rate is $1/m^{1/4}$, we have that $m = \tilde{\Omega}(\epsilon^{-2})$ is strong enough to ensure that $|K_0(x,x') - K^\infty(x,x')| \leq \epsilon^{1/2}$. Given these results, it is reasonable to make the following assumption for the architectures we consider (see Appendix E).

Assumption 3.6. *$m = \tilde{\Omega}(\epsilon^{-2})$ suffices to ensure that $\|K_0 - K^\infty\|^2_{L^2(X\times X,\rho\otimes\rho)} \leq \epsilon$ holds with high probability $1 - \delta(m)$ over the initialization θ_0 where $\delta(m) = o(1)$.*

Under this assumption, by setting $K = K^\infty$ in Theorem 3.5 we get the following corollary.

Corollary 3.7. *Let $\delta(m)$ be defined as in Assumption 3.6 which we assume to hold. Let $T \geq 1$ and $\epsilon > 0$. For $k \in \mathbb{N}$ let $P_k : L^2(X,\rho) \to L^2(X,\rho)$ denote the orthogonal projection onto the span of*

https://doi.org/10.52202/068431-2201

the top k eigenfunctions of the operator T_{K^∞} defined in Equation (1). Let $\sigma_k > 0$ denote the k-th eigenvalue of T_{K^∞}. Then $m = \tilde{\Omega}(T^4/\epsilon^2)$ and $n = \tilde{\Omega}(T^2/\epsilon^2)$ suffices to ensure with probability at least $1 - O(mn)\exp(-\Omega(\log^2(m))) - \delta(m)$ that for all $t \leq T$ and $k \in \mathbb{N}$

$$\|P_k(r_t - \exp(-T_{K^\infty}t)r_0)\|^2_{L^2(X,\rho)} \leq \left[\frac{1-\exp(-\sigma_k t)}{\sigma_k}\right]^2 \cdot \epsilon$$

and

$$\|r_t - \exp(-T_{K^\infty}t)r_0\|^2_{L^2(X,\rho)} \leq t^2 \cdot \epsilon.$$

Informally Corollary 3.7 states that up to the stopping time T, we have that $r_t \approx \exp(-T_{K^\infty}t)r_0$. As discussed before, the term $\exp(-T_{K^\infty}t)r_0$ projected along the i-th eigenfunction of K^∞ decays linearly, $\langle \exp(-T_{K^\infty}t)r_0, \phi_i\rangle_\rho = \exp(-\sigma_i t)\langle r_0, \phi_i\rangle_\rho$. Given that K^∞ tends to have a highly skewed spectrum (see, e.g. Figure 1), the effect the magnitude of σ_i has on the convergence rate is particularly relevant. Furthermore the bound on the projected difference $\|P_k(r_t - \exp(-T_{K^\infty}t)r_0)\|_{L^2(X,\rho)}$ is smaller whenever σ_k is large due to the dependence of the bound on the inverse eigenvalue σ_k^{-1}. Thus we have that the bias along the top eigenfunctions is particularly pronounced. Hence we make the following important observation.

Observation 3.8. *At the beginning of training the network learns projections along eigenfunctions of the Neural Tangent Kernel integral operator T_{K^∞} at rates corresponding to their eigenvalues. This is particularly true for the eigenfunctions with large eigenvalues.*

Scaling with respect to width and number of training data samples Now let us interpret how the width m and number of training samples n in the theorem scale. We note that as long as $n \leq m^\alpha$ for some $\alpha > 0$ the failure probability $O(mn)\exp(-\Omega(\log^2(m)))$ goes to zero as $m \to \infty$. Thus once m and n are sufficiently large relative to the stopping time T and precision ϵ, they can both tend to infinity at just about any rate to achieve a high probability bound. We also observe that m and n both have the same scaling with respect to ϵ, namely $m, n = \tilde{\Omega}(\epsilon^{-2})$. Thus for a fixed stopping time T we can send m and n to infinity at the same rate $m \sim n$ to send the error $\epsilon \to 0$. This is significant as typical NTK analysis requires $m = \Omega(poly(n))$. We reach following important conclusion.

Observation 3.9. *The network will inherit the bias of the kernel at the beginning of training even when the width m only grows linearly with the number of samples n.*

Scaling with respect to stopping time We will now address the scaling with respect to the stopping time T. The relevant question is how quickly the terms $P_k \exp(-T_{K^\infty}t)r_0$ and $\exp(-T_{K^\infty}t)r_0$ converge to zero. We observe that

$$\|P_k \exp(-T_{K^\infty}t)r_0\|_{L^2(X,\rho)} \leq \exp(-\sigma_k t)\|r_0\|_{L^2(X,\rho)} \leq \exp(-\sigma_k t)\|f^*\|_{L^\infty(X,\rho)},$$

where we have used the antisymmetric initialization $r_0 = f(\bullet;\theta_0) - f^* = 0 - f^* = -f^*$ and the basic inequality $\|\bullet\|_{L^2(X,\rho)} \leq \|\bullet\|_{L^\infty(X,\rho)}$. Based on this we have that $t \geq \log(\|f^*\|_{L^\infty(X,\rho)}/\epsilon)/\sigma_k$ suffices to ensure $\|P_k \exp(-T_{K^\infty}t)r_0\|_{L^2(X,\rho)} \leq \epsilon$. Using this fact we get the following corollary.

Corollary 3.10. *Let $\delta(m)$ be defined as in Assumption 3.6 which is assumed to hold. Let $T = \tilde{\Omega}(1/\sigma_k)$ and $\epsilon > 0$. For $k \in \mathbb{N}$ let $P_k : L^2(X,\rho) \to L^2(X,\rho)$ denote the orthogonal projection onto the span of the top k eigenfunctions of the operator T_{K^∞} defined in Equation (1). Let $\sigma_k > 0$ denote the k-th eigenvalue of T_{K^∞}. Then $m = \tilde{\Omega}({\sigma_k}^{-8}/\epsilon^2)$ and $n = \tilde{\Omega}(\sigma_k^{-6}/\epsilon^2)$ suffices to ensure that with probability at least $1 - O(mn)\exp(-\Omega(\log^2(m))) - \delta(m)$*

$$\|P_k r_T\|^2_{L^2(X,\rho)} \leq \epsilon$$

and in particular

$$\frac{1}{2}\|r_T\|^2_{L^2(X,\rho)} \leq \tilde{O}(\epsilon) + \|(I - P_k)r_0\|^2_{L^2(X,\rho)}.$$

The interpretation of the Corollary 3.10 is that the stopping time $T = \tilde{\Omega}(1/\sigma_k)$ is long enough to ensure that the network has learned the top k eigenfunctions to ϵ accuracy provided that $m = \tilde{\Omega}(\sigma_k^{-8}\epsilon^{-2})$ and $n = \tilde{\Omega}(\sigma_k^{-6}\epsilon^{-2})$. We note that the second conclusion of Corollary 3.10 is a bound on the test error $\frac{1}{2}\|r_t\|^2_{L^2(X,\rho)}$. From the antisymmetric initialization $r_0 = -f^*$ so that

$\|(I-P_k)r_0\|^2_{L^2(X,\rho)} = \|(I-P_k)f^*\|^2_{L^2(X,\rho)}$. For a general target f^*, this quantity can decay arbitrary slowly with respect to k. Our goal with Theorem 3.5 was not to get a learning guarantee, but to describe how the bias of the kernel K^∞ is inherited by the finite-width network at the beginning of training even for general target functions. Nevertheless we will briefly sketch how it is possible to get a learning guarantee from Corollary 3.7 when f^* is in the RKHS of K^∞. In this case one can show that $\|\exp(-T_{K^\infty}t)r_0\|^2_{L^2(X,\rho)} = O\left(\frac{\|f^*\|^2_{\mathcal{H}}}{t}\right)$ where $\|\bullet\|_{\mathcal{H}}$ is the RKHS norm. Then treating $\|f^*\|_{\mathcal{H}}$ as a constant one can choose the stopping time $T \sim \epsilon^{-1}$ to bring the test error to ϵ provided that $m, n = \tilde{\Omega}(poly(\epsilon^{-1}))$. More generally Velikanov & Yarotsky (2021) derive sufficient conditions for the power law $\|\exp(-T_{K^\infty}t)r_0\|^2_{L^2(X,\rho)} \sim Ct^{-\xi}$ to hold. Using a similar argument in this case one can choose the stopping time $T \sim \epsilon^{-1/\xi}$ and get a learning guarantee for $m, n = \tilde{\Omega}(poly(\epsilon^{-1}))$.

3.2 Technical Comparison to Prior Work

Lee et al. (2019); Arora et al. (2019b) compared the network $f(x;\theta)$ to its linearization $f_{lin}(x;\theta) := \langle \nabla_\theta f(x;\theta_0), \theta - \theta_0\rangle + f(x;\theta_0)$ in the regime where $m = \Omega(poly(n))$. When $m = \Omega(poly(n))$ one can show the loss converges to zero and the parameter changes $\|\theta_t - \theta_0\|_2$ are bounded. By contrast we avoid the condition $m = \Omega(poly(n))$ by employing a stopping time. Arora et al. (2019a); Cao et al. (2021); Basri et al. (2020) proved statements similar to Theorem 3.5 and Corollary 3.7 that roughly correspond to replacing T_{K^∞} with its Gram matrix induced by the training data $(G^\infty)_{i,j} = K^\infty(x_i, x_j)$ and replacing ρ with the empirical measure $\hat{\rho} = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$. Arora et al. (2019a); Basri et al. (2020) operate in the regime where $m = \Omega(poly(n))$ and as a benefit do not need to employ a stopping time. Cao et al. (2021) instead of requiring $m = \Omega(poly(n))$ requires that the width m satisfies at least $m = \Omega(\max\{\sigma_k^{-14}, \epsilon^{-6}\})$ where σ_k is the cutoff eigenvalue. The most similar work is Bowman & Montúfar (2022), which demonstrated a version of Corollary 3.7 for a shallow feedforward network that is underparameterized. If p is the total number of parameters, they require $m = \tilde{\Omega}(\epsilon^{-1}T^2)$ and $n = \tilde{\Omega}(\epsilon^{-1}pT^2)$. This requires the network to be greatly underparameterized $n \gg p$. Our result was able to remove the dependence of n on p and demonstrate the result for general deep architectures at the expense of slightly worse scaling with respect to T and ϵ.

4 Proof Sketch

For simplicity we will go through the case where $K = K^\infty$. At a high level the proof revolves around bounding the difference between the operators T_{K^∞} and T_n^t defined in Equations (1) and (4).

Bounding Operator Deviations Bowman & Montúfar (2022) demonstrated

$$r_t = \exp(-T_{K^\infty}t)r_0 + \int_0^t \exp(-T_{K^\infty}(t-s))(T_{K^\infty} - T_n^s)r_s ds.$$

This exhibits the residual r_t as a sum of $\exp(-T_{K^\infty}t)r_0$ and a correction term. The proof of Theorem 3.5 revolves around bounding the correction term which involves bounding

$$\|(T_{K^\infty} - T_n^s)r_s\|_{L^2(X,\rho)} \leq \|(T_{K^\infty} - T_n)r_s\|_{L^2(X,\rho)} + \|(T_n - T_n^s)r_s\|_{L^2(X,\rho)}.$$

At a high level $\|(T_n - T_n^s)r_s\|_{L^2(X,\rho)}$ will be small whenever the kernel deviations $K_0 - K_s$ are small. On the other hand by metric entropy based arguments we have that $\|(T_{K^\infty} - T_n)r_s\|_{L^2(X,\rho)}$ will be small whenever n is large enough relative to the complexity of the residual functions r_s.

Comparison with Linearization Let $H(x;\theta) := \nabla^2_\theta f(x;\theta)$ denote the Hessian of our network with respect to the parameters θ for a fixed input x. It turns out that if $\|H(x,\theta)\|_{op}$ was uniformly small over x and θ then the kernel deviations $K_0 - K_s$ would be bounded and the complexity of our model $f(x;\theta)$ would be controlled by the complexity of the linearized model $f_{lin}(x;\theta) := \langle \nabla_\theta f(x;\theta_0), \theta - \theta_0\rangle$. The caveat to this approach is we do not in fact have a way to bound the Hessian $H(x,\theta)$ uniformly. However Liu et al. (2020b) demonstrated that for *fixed* x and $R > 0$ we have with high probability over the initialization θ_0

$$\sup_{\theta \in \overline{B}(\theta_0, R)} \|H(x,\theta)\|_{op} = \tilde{O}\left(\frac{R}{\sqrt{m}} poly(R/\sqrt{m})\right). \tag{5}$$

https://doi.org/10.52202/068431-2201

Using a priori parameter norm deviation bounds we have that $\|\theta_t - \theta_0\|_2 = O(\sqrt{t})$ and thus we can set $R = O(\sqrt{T})$. The difficulty then arises to get bounds that only depend on the Hessian $H(x;\theta)$ evaluated only on finitely many inputs x. We overcome this difficulty by showing for fixed θ_0 one has high probability bounds over the sampling of the training data $x_1, \ldots, x_n$ that only require the Hessian evaluated on a finite point set. This requires some elaborate calculations involving Rademacher complexity. We then use the Fubini-Tonelli theorem and the Hessian bound (5) to get a bound over the simultaneous sampling of θ_0 and $x_1, \ldots, x_n$.

Covering Number of the Linearized Model The complexity of the residual functions r_s up to the stopping time T can be controlled by bounding the complexity of the function class $\mathcal{C} = \{f_{lin}(x;\theta) : \theta \in \overline{B}(\theta_0, R)\}$. In Appendix A we show that the $L^2(X,\rho)$ metric entropy of the linearized model $\mathcal{C} = \{f_{lin}(x;\theta) : \theta \in \overline{B}(\theta_0, R)\}$ is determined by the spectrum of the Fisher Information Matrix

$$F := \int_X \nabla_\theta f(x;\theta_0) \nabla_\theta f(x;\theta_0)^T d\rho(x). \tag{6}$$

Let $\lambda_1^{1/2} \geq \lambda_2^{1/2} \geq \cdots \geq 0$ denote the eigenvalues of $F^{1/2}$. We define the effective rank of $F^{1/2}$ at scale ϵ as

$$\tilde{p}(F^{1/2}, \epsilon) = |\{i : \lambda_i^{1/2} > \epsilon\}|.$$

This measures the number of dimensions within the unit ball whose image under $F^{1/2}$ can be larger than ϵ in Euclidean norm. In Appendix A we demonstrate that the ϵ covering number of $\mathcal{C}$ in $L^2(X,\rho)$, denoted $\mathcal{N}(\mathcal{C}, \|\bullet\|_{L^2(X,\rho)}, \epsilon)$, has the bound

$$\log \mathcal{N}(\mathcal{C}, \|\bullet\|_{L^2(X,\rho)}, \epsilon) = \tilde{O}(\tilde{p}(F^{1/2}, 0.75\epsilon/R)).$$

It turns out that for $\|(T_{K^\infty} - T_n) r_s\|_{L^2(X,\rho)}$ to be on the order of ϵ we merely need n to be large relative to $\tilde{p}(F^{1/2}, 0.75\epsilon/R)$. By contrast Bowman & Montúfar (2022) required that the network was underparameterized so that n was large relative to the total number of parameters p. Since $\tilde{p} \ll p$, this is what lets us relax the sample complexity dramatically. In fact for fixed R and ϵ we have that $\tilde{p} = \tilde{O}(1)$ with high probability as the width grows to infinity whereas $p \to \infty$. Interestingly, the quantity $\tilde{p}$ for the loss Hessian at convergence was used recently to derive analytical PAC-Bayes bounds (Yang et al., 2021). Note for the squared loss the (empirical) FIM[1] can be taken as an approximation to the Hessian, and at a minimizer this approximation becomes exact. Thus these two notions are closely related.

5 Conclusion and Future Directions

We provided quantitative bounds measuring the L^2 difference in function space between a finite-width network trained on finitely many samples and the corresponding kernel method with infinite width and infinite data. As a consequence, the network will inherit the bias of the kernel at the beginning of training even when the width scales linearly with the number of samples. This bias is not only over the training data but over the entire input space. The key property that allows this is the low-effective-rank property of the Fisher Information Matrix (FIM) at initialization which controls the capacity of the model at the beginning of training. An interesting avenue for future work is to investigate if flat minima manifesting a FIM of low effective rank at the end of training can be related to the behavior of the network on out-of-sample data after training.

Limitations Our framework can only characterize the network's bias up to a stopping time. There is compelling evidence that the kernel adapts to the target function later in training (Baratin et al., 2021; Atanasov et al., 2022), and this falls outside our framework. Accounting for adaptations in the kernel is an important problem that is still being addressed by the theoretical community.

Broader impacts We do not foresee any negative societal impacts of characterizing the spectral bias of neural networks. To the contrary we believe that cataloging the properties that networks are biased towards in a variety of regimes will be essential to developing fair and interpretable artificial intelligence over the long-term.

[1] Note that we define F as an expectation over the true input distribution ρ. To approximate the Hessian of the empirical loss one must replace ρ with the empirical measure $\hat{\rho}$.

https://doi.org/10.52202/068431-2201

Acknowledgments and Disclosure of Funding

This project has received funding from UCLA FCDA, the National Science Foundation Division of Mathematical Sciences (NSF DMS-2145630), and from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement nº 757983). The authors would like to thank Yonatan Dukler for sharing code to compute the NTK Gram matrix in PyTorch.

References

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 242–252. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/allen-zhu19a.html.

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 322–332. PMLR, 09–15 Jun 2019a. URL https://proceedings.mlr.press/v97/arora19a.html.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019b. URL https://proceedings.neurips.cc/paper/2019/file/dbc4d84bfcfe2284ba11beffb853a8c4-Paper.pdf.

Alexander Atanasov, Blake Bordelon, and Cengiz Pehlevan. Neural networks as kernel learners: The silent alignment effect. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=1NvflqAdoom.

Aristide Baratin, Thomas George, César Laurent, R Devon Hjelm, Guillaume Lajoie, Pascal Vincent, and Simon Lacoste-Julien. Implicit regularization via neural feature alignment. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2269–2277. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/baratin21a.html.

Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 3(null):463–482, mar 2003. ISSN 1532-4435.

Ronen Basri, David Jacobs, Yoni Kasten, and Shira Kritchman. The convergence rate of neural networks for learned functions of different frequencies. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/5ac8bb8a7d745102a978c5f8ccdb61b8-Paper.pdf.

Ronen Basri, Meirav Galun, Amnon Geifman, David Jacobs, Yoni Kasten, and Shira Kritchman. Frequency bias in neural networks for input of non-uniform density. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 685–694. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/basri20a.html.

Benjamin Bowman and Guido Montúfar. Implicit bias of MSE gradient optimization in underparameterized neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=VLgmhQDVBV.

Sam Buchanan, Dar Gilboa, and John Wright. Deep networks and the multiple manifold problem. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=O-6Pm_d_Q-.

https://doi.org/10.52202/068431-2201

Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Quanquan Gu. Towards understanding the spectral bias of deep learning. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 2205–2211. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/304. URL https://doi.org/10.24963/ijcai.2021/304. Main Track.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1675–1685. PMLR, 09–15 Jun 2019a. URL https://proceedings.mlr.press/v97/du19c.html.

Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019b. URL https://openreview.net/forum?id=S1eK3i09YQ.

Ilya Dumer. Covering an ellipsoid with equal balls. *J. Comb. Theory Ser. A*, 113(8):1667–1676, nov 2006. ISSN 0097-3165. doi: 10.1016/j.jcta.2006.03.021. URL https://doi.org/10.1016/j.jcta.2006.03.021.

Weinan E, Chao Ma, and Lei Wu. A comparative analysis of optimization and generalization properties of two-layer neural network and random feature models under gradient descent dynamics. *Science China Mathematics*, 63:1235–1258, 2020.

Zhou Fan and Zhichao Wang. Spectra of the conjugate kernel and neural tangent kernel for linear-width neural networks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Y. Gordon, Hermann König, and Carsten Schütt. Geometric and probabilistic estimates for entropy and approximation numbers of operators. *Journal of Approximation Theory*, 49:219–239, 1987.

Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/58191d2a914c6dae66371c9dcdc91b41-Paper.pdf.

Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/0e98aeeb54acf612b9eb4e48a269814c-Paper.pdf.

Boris Hanin and Mihai Nica. Finite depth and width corrections to the neural tangent kernel. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJgndT4KwB.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Jiaoyang Huang and Horng-Tzer Yau. Dynamics of deep neural networks and neural tangent hierarchy. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4542–4551. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/huang20l.html.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3): 90–95, 2007. doi: 10.1109/MCSE.2007.55.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/5a4be1fa34e62bb8a6ec6b91d2462f5a-Paper.pdf.

Ziwei Ji and Matus Telgarsky. The implicit bias of gradient descent on nonseparable data. In Alina Beygelzimer and Daniel Hsu (eds.), *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pp. 1772–1798. PMLR, 25–28 Jun 2019. URL https://proceedings.mlr.press/v99/ji19a.html.

Ziwei Ji and Matus Telgarsky. Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow ReLU networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HygegyrYwH.

Hui Jin and Guido Montúfar. Implicit bias of gradient descent for mean squared error regression with wide neural networks, 2021. arXiv:2006.07356.

Ryo Karakida, Shotaro Akaho, and Shun-ichi Amari. Pathological Spectra of the Fisher Information Metric and Its Variants in Deep Neural Networks. *Neural Computation*, 33(8):2274–2307, July 2021. ISSN 0899-7667. doi: 10.1162/neco_a_01411. URL https://doi.org/10.1162/neco_a_01411. _eprint: https://direct.mit.edu/neco/article-pdf/33/8/2274/1930880/neco_a_01411.pdf.

Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Willing. Jupyter notebooks – a publishing format for reproducible computational workflows. In F. Loizides and B. Schmidt (eds.), *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pp. 87 – 90. IOS Press, 2016.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/0d1a9651497a38d8b1c3871c84528bd4-Paper.pdf.

Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 4313–4324. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/li20j.html.

Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet (eds.), *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pp. 2–47. PMLR, 06–09 Jul 2018. URL https://proceedings.mlr.press/v75/li18a.html.

Chaoyue Liu, Libin Zhu, and Mikhail Belkin. On the linearity of large non-linear models: when and why the tangent kernel is constant. *CoRR*, abs/2010.01092, 2020a. URL https://arxiv.org/abs/2010.01092.

Chaoyue Liu, Libin Zhu, and Misha Belkin. On the linearity of large non-linear models: when and why the tangent kernel is constant. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 15954–15964. Curran Associates, Inc., 2020b. URL https://proceedings.neurips.cc/paper/2020/file/b7ae8fecf15b8b6c3c69eceae636d203-Paper.pdf.

https://doi.org/10.52202/068431-2201

Tao Luo, Zheng Ma, Zhi-Qin John Xu, and Yaoyu Zhang. On the exact computation of linear frequency principle dynamics and its generalization, 2020. URL https://arxiv.org/abs/2010.08153.

Mor Shpigel Nacson, J. Lee, Suriya Gunasekar, Nathan Srebro, and Daniel Soudry. Convergence of gradient descent on separable data. In *AISTATS*, 2019.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6614.

Behnam Neyshabur, Ryota Tomioka, Ruslan Salakhutdinov, and Nathan Srebro. Geometry of optimization and implicit regularization in deep learning, 2017. arXiv:1705.03071.

Quynh Nguyen. On the proof of global convergence of gradient descent for deep relu networks with linear widths. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8056–8062. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/nguyen21a.html.

Quynh Nguyen and Marco Mondelli. Global convergence of deep networks with one wide layer followed by pyramidal topology. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 11961–11972. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/8abfe8ac9ec214d68541fcb888c0b4c3-Paper.pdf.

Samet Oymak and Mahdi Soltanolkotabi. Toward moderate overparameterization: Global convergence guarantees for training shallow neural networks. *IEEE Journal on Selected Areas in Information Theory*, 1(1):84–105, 2020.

Samet Oymak, Zalan Fabian, Mingchen Li, and Mahdi Soltanolkotabi. Generalization guarantees for neural networks via harnessing the low-rank structure of the Jacobian, 2020. URL https://openreview.net/forum?id=ryl5CJSFPS.

Vardan Papyan. Traces of class/cross-class structure pervade deep learning spectra. *Journal of Machine Learning Research*, 21(252):1–64, 2020. URL http://jmlr.org/papers/v21/20-933.html.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Jeffrey Pennington and Yasaman Bahri. Geometry of neural network loss surfaces via random matrix theory. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2798–2806. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/pennington17a.html.

Jeffrey Pennington and Pratik Worah. The spectrum of the fisher information matrix of a single-hidden-layer neural network. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/18bb68e2b38e4a8ce7cf4f6b2625768c-Paper.pdf.

Fernando Pérez and Brian E. Granger. IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29, May 2007. ISSN 1521-9615. doi: 10.1109/MCSE.2007.53. URL https://ipython.org.

Gilles Pisier. *The Volume of Convex Bodies and Banach Space Geometry*. Cambridge Tracts in Mathematics. Cambridge University Press, 1989. doi: 10.1017/CBO9780511662454.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5301–5310. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/rahaman19a.html.

Lorenzo Rosasco, Mikhail Belkin, and Ernesto De Vito. On learning with integral operators. *Journal of Machine Learning Research*, 11(30):905–934, 2010. URL http://jmlr.org/papers/v11/rosasco10a.html.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(70):1–57, 2018. URL http://jmlr.org/papers/v19/18-188.html.

Lili Su and Pengkun Yang. On learning over-parameterized neural networks: A functional approximation perspective. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/253f7b5d921338af34da817c00f42753-Paper.pdf.

Matus Telgarsky. Deep learning theory lecture notes. https://mjt.cs.illinois.edu/dlt/, 2021. Version: 2021-10-27 v0.0-e7150f2d (alpha).

Maksim Velikanov and Dmitry Yarotsky. Explicit loss asymptotics in the gradient descent training of neural networks. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 2570–2582, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/14faf969228fc18fcd4fcf59437b0c97-Abstract.html.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In *Compressed Sensing*, chapter 5. Cambridge University Press, 2012.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019. doi: 10.1017/9781108627771.

Francis Williams, Matthew Trager, Daniele Panozzo, Claudio Silva, Denis Zorin, and Joan Bruna. Gradient dynamics of shallow univariate relu networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/1f6419b1cbe79c71410cb320fc094775-Paper.pdf.

Zhi-Qin John Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain. In Tom Gedeon, Kok Wai Wong, and Minho Lee (eds.), *Neural Information Processing*, pp. 264–274, Cham, 2019. Springer International Publishing.

Ge Yang, Anurag Ajay, and Pulkit Agrawal. Overcoming the spectral bias of neural value approximation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=vIC-xLFuM6.

Greg Yang. Tensor programs II: Neural tangent kernel for any architecture, 2020. URL https://arxiv.org/abs/2006.14548.

 https://doi.org/10.52202/068431-2201

Greg Yang and Hadi Salman. A fine-grained spectral perspective on neural networks, 2019. URL https://arxiv.org/abs/1907.10599.

Rubing Yang, Jialin Mao, and Pratik Chaudhari. Does the data induce capacity control in deep learning? *ArXiv*, abs/2110.14163, 2021.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Sy8gdB9xx.

Yaoyu Zhang, Zhi-Qin John Xu, Tao Luo, and Zheng Ma. A type of generalization error induced by initialization in deep neural networks. In Jianfeng Lu and Rachel Ward (eds.), *Proceedings of The First Mathematical and Scientific Machine Learning Conference*, volume 107 of *Proceedings of Machine Learning Research*, pp. 144–164. PMLR, 20–24 Jul 2020. URL https://proceedings.mlr.press/v107/zhang20a.html.

Difan Zou and Quanquan Gu. An improved analysis of training over-parameterized deep neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/6a61d423d02a1c56250dc23ae7ff12f3-Paper.pdf.

Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep ReLU networks. *Machine learning*, 109:467–492, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 2 and Section 3
   (b) Did you include complete proofs of all theoretical results? [Yes] The proofs are included in the supplementary material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Appendix F
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] In Figure 1 we plot the curve for each of the 10 runs of the experiment.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix F
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Appendix F
   (b) Did you mention the license of the assets? [Yes] See Appendix F
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See Appendix F
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See Appendix F
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Appendix F
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

https://doi.org/10.52202/068431-2201