Uniform Convergence, Adversarial Spheres and a Simple Remedy

Gregor Bachmann [1] **Seyed-Mohsen Moosavi-Dezfooli** [1] **Thomas Hofmann** [1]

Abstract

Previous work has cast doubt on the general framework of uniform convergence and its ability to explain generalization in neural networks. By considering a specific dataset, it was observed that a neural network completely misclassifies a projection of the training data (adversarial set), rendering any existing generalization bound based on uniform convergence vacuous. We provide an extensive theoretical investigation of the previously studied data setting through the lens of infinitely-wide models. We prove that the Neural Tangent Kernel (NTK) also suffers from the same phenomenon and we uncover its origin. We highlight the important role of the output bias and show theoretically as well as empirically how a sensible choice completely mitigates the problem. We identify sharp phase transitions in the accuracy on the adversarial set and study its dependency on the training sample size. As a result, we are able to characterize critical sample sizes beyond which the effect disappears. Moreover, we study decompositions of a neural network into a clean and noisy part by considering its canonical decomposition into its different eigenfunctions and show empirically that for too small bias the adversarial phenomenon still persists.

1. Introduction

Neural networks have achieved astonishing performance across many learning tasks such as in computer vision (He et al., 2016), natural language processing (Devlin et al., 2019) and graph learning (Kipf & Welling, 2017). The theoretical understanding of the generalization capability of these models, on the other hand, has been lagging behind in development and can so far only offer limited insights into the inner workings of these algorithms. Almost every work concerning generalization is based on the paradigm of uniform convergence as a tool to bound the capacity of the model (Arora et al., 2018; Bartlett et al., 2019; 2017; Neyshabur et al., 2015; 2018). Recently however, Nagarajan & Kolter (2019b) have cast doubt on the power of this technique. By constructing a dataset consisting of two concentric spheres (referred to as adversarial spheres), they were able to show that a neural network misclassifies a specific projection of the training data entirely. The existence of such an adversarial dataset renders any generalization bound based on uniform convergence vacuous. This surprising behaviour has been shown to hold empirically but, to the best of our knowledge, neither a mathematical proof nor a theoretical account of its origin has been given in the literature.

In this work, we revisit the aforementioned dataset and study the phenomenon of the model mathematically through the lens of infinitely wide neural networks. We leverage the analytic structure of the Neural Tangent Kernel (NTK) (Jacot et al., 2018) to prove the observed behaviour as well as unravel the dependencies on different parameters such as the sample size and the magnitude of the bias of the output layer of the model. These theoretical findings suggest a very simple fix consisting in increasing the output bias sufficiently. We validate our theoretical results using numerical experiments on the adversarial spheres dataset. Moreover, we explore the hypothesis put forth by Nagarajan & Kolter (2019b) suggesting that there may exist a decomposition of the model into a clean and a noisy part. The noisy submodel should encapsulate the observed degeneracies while the clean submodel enjoys good generalization and robustness, making it amenable to uniform convergence. We investigate the most natural decomposition induced by the eigendecomposition of the kernel and show that even a restriction to the optimal set of eigenfunctions does not eliminate the adversarial effect.

Our mathematical analysis suggests that the failure of uniform convergence in this particular setting is not pointing towards a deeper problem in neural architectures but is rather a result of the specific dataset and the architectural bias encouraging the network to rely on angular features instead of radial information. This questions the relevance of the observation in Nagarajan & Kolter (2019b) regarding more realistic datasets containing angular structure.

[1] Department of Computer Science, ETH Zürich. Correspondence to: Gregor Bachmann <gregor.bachmann@inf.ethz.ch>.

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

We structure our work as follows. We first discuss related work in Section 2, followed by an overview of the mathematical setting and notation in Section 3. In Section 4, we proceed to summarize the main results of Nagarajan & Kolter (2019b) and Jacot et al. (2018) as we build upon their findings. We then present our own theoretical and numerical results in Sections 5 and 6, detailing the origin of the adversarial effect and its behaviour under a decomposition of the model. Finally, we provide a discussion of the implications of our work in Section 7.

2. Related Work

The goal of understanding generalization capabilities of neural networks gave rise to a rich line of work. A multitude of approaches to this task have been explored in the literature, Bartlett et al. (2017); Neyshabur et al. (2015) for instance derive guarantees based on Rademacher complexities and covering numbers, resulting in upper bounds involving diverse norms of the weight matrices of the network. Other works investigate how compressing the model might help to derive meaningful guarantees, ensuring that the original and the compressed version remain close (Arora et al., 2018; Zhou et al., 2019). Others focus on randomized neural networks, leveraging the rich PAC-Bayesian theory to derive non-vacuous bounds (Dziugaite & Roy, 2017; Zhou et al., 2019). Derandomization of those bounds on the other hand strongly deteriorates their effectiveness (Neyshabur et al., 2018; Nagarajan & Kolter, 2019a).

The underlying framework shared between these diverse approaches is uniform convergence. This widely used paradigm has been recently questioned by Nagarajan & Kolter (2019b), demonstrating its failure in the most optimistic setting for a neural network with a very simple data distribution. To the best of our knowledge, little to no work in the literature has provided a theoretical account of this phenomenon or described its origin mathematically. The work closest to ours is Negrea et al. (2020), describing how a (possibly random) surrogate of the model can make uniform convergence applicable again. We however directly analyze the model in question instead of studying an approximation. Thus any insights derived from our analysis can point to a deeper problem in neural architectures.

Similar limitations have been discovered for kernel regression (Belkin et al., 2018) but in contrast to Nagarajan & Kolter (2019b), the results only apply in the presence of label noise.

Recent works have established a direct correspondence between kernel regression and an infinitely wide fully-connected neural network at initialization (Lee et al., 2018) as well as during gradient flow training (Jacot et al., 2018). Various follow-up works have refined these results, extending the analysis to various architectures (Arora et al., 2019; Huang et al., 2020; Du et al., 2019) and discrete gradient descent (Lee et al., 2019). The direct connection to the field of kernel regression makes the mathematical analysis of various phenomena in neural network training tractable. We leverage the convenient closed-form expression for a network trained with gradient descent in order to unravel the degeneracy of the model on the adversarial dataset outlined in Nagarajan & Kolter (2019b).

3. Notation and Definitions

We will establish some notation for the quantities of interest throughout this paper. Denote a fully-connected L-layer neural network through the recursive equations

- $\boldsymbol{f}^{(l+1)}(\boldsymbol{x}) = \boldsymbol{W}^{(l+1)}\boldsymbol{\alpha}^{(l)}(\boldsymbol{x}) + \boldsymbol{b}^{(l+1)}$
- $\boldsymbol{\alpha}^{(l)}(\boldsymbol{x}) = \sigma\left(\boldsymbol{f}^{(l)}(\boldsymbol{x})\right)$

where $l = 0, \dots L-1$, $f^{(0)}(\boldsymbol{x}) = \boldsymbol{x}$ and scalar output $f^{(L)}(\boldsymbol{x}) = \boldsymbol{W}^{(L)}\boldsymbol{\alpha}^{(L-1)}(\boldsymbol{x}) + b^{(L)} \in \mathbb{R}$. We have an input $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d$, weight matrices $\boldsymbol{W}^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$, biases $\boldsymbol{b}^{(l)} \in \mathbb{R}^{d_l}$ and a component-wise non-linearity $\sigma : \mathbb{R} \to \mathbb{R}$. We denote by $\mathcal{F}$ the function class consisting of all possible neural networks. Moreover, define $\boldsymbol{\theta} \in \mathbb{R}^M$ as the concatenation of all parameters $\left(\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}\right), \dots, \left(\boldsymbol{W}^{(L)}, b^{(L)}\right)$ of the network, where $M \in \mathbb{N}$ denotes the total number of parameters in the model.

Consider a dataset $\mathcal{S} = \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\}$ where $(\boldsymbol{x}_i, y_i) \overset{\text{i.i.d}}{\sim} \mathcal{D}$ for $i = 1, \dots, n$ are distributed according to some probability distribution $\mathcal{D}$. We refer to $\boldsymbol{x}_i \in \mathcal{X}$ as the input with corresponding targets $y_i \in \mathcal{Y} \subset \mathbb{R}$. To make the notation clearer, we will sometimes use $y_{\boldsymbol{x}}$ to denote the label y corresponding to $\boldsymbol{x}$. Occasionally, we will use $\boldsymbol{x}_i \sim p$ where p is the marginal distribution of $\mathcal{D}$ with respect to the inputs. We will denote by $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ and $\boldsymbol{y} \in \mathbb{R}^n$ the stacking of all observations into a matrix and vector respectively. We define a loss function $L_f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that quantifies how close the prediction $f(\boldsymbol{x}_i)$ is to the ground truth y_i. We then train the model to minimize the empirical loss $\hat{L}$ consisting of the losses incurred on each sample:

$$\hat{L}_S : \mathcal{F} \to \mathbb{R} \, , \, f \mapsto \hat{L}_S(f) = \sum_{i=1}^{n} L_f(\boldsymbol{x}_i, y_i)$$

The more important quantity from a practical point of view, however, is given by the generalization error of the model:

$$L : \mathcal{F} \to \mathbb{R}, \, f \mapsto L(f) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[L_f(\boldsymbol{x}, y)]$$

Understanding how much the generalization error can deviate from the empirical loss for a given data distribution and model is of paramount importance both in theory as well as in practice.

4. Uniform Convergence and Neural Tangent Kernel

In this section we give a brief overview of the previous work we will build upon. We first outline the key result of Nagarajan & Kolter (2019b) on uniform convergence in order to motivate our theoretical analysis. We then shortly summarize the NTK framework introduced in Jacot et al. (2018) as it serves as the main tool in our work.

4.1. Uniform Convergence and its Weaknesses

Recently, Nagarajan & Kolter (2019b) investigated how well the performance of neural networks can be captured by the very general machinery of uniform convergence. They study the most optimistic setup for uniform convergence by assuming that a perfect characterization of the solution space of gradient descent is known, critically reducing the hypothesis space needed to control. Under this assumption, a dataset is constructed which provably cannot be explained by uniform convergence. The argument goes along the following lines. Assume we have some algorithm $\mathcal{A}$ that chooses $f \in \mathcal{F}$ given a particular realization of a training set $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. A uniform convergence bound is defined as the smallest ϵ_{unif} such that:

$$\mathbb{P}_{\mathcal{S}\sim\mathcal{D}^n}\left(\sup_{f\in\mathcal{F}} |L(f) - \hat{L}_S(f)| \leq \epsilon_{\text{unif}}\right) \geq 1-\delta$$

One considers a supremum over $\mathcal{F}$ to strip the chosen hypothesis $f \in \mathcal{F}$ of its complicated dependency on the data, making it more amenable to mathematical analysis. However $\mathcal{A}$ will never pick most of the hypotheses in $\mathcal{F}$, leading to an inflation of ϵ_{unif}. Ideally, to reduce the supremum, one would restrict $\mathcal{F}$ to only those hypotheses that are considered by $\mathcal{A}$, denoted by $\mathcal{F}_{\mathcal{A}}$. Further pruning the search space is not possible as we would exclude models that could actually be chosen by $\mathcal{A}$. We can reformulate the uniform bound as follows. Consider a set of sets $\mathcal{S}_\delta$ consisting of different realizations of training sets $\mathcal{S}$ such that

$$\mathbb{P}_{\mathcal{S}\sim\mathcal{D}^n}(\mathcal{S}\in\mathcal{S}_\delta) \geq 1-\delta$$

Then the most optimistic uniform bound is given by the smallest $\epsilon_{\text{unif},\mathcal{A}}$ such that

$$\sup_{S\in\mathcal{S}_\delta} \sup_{f\in\mathcal{F}_{\mathcal{A}}} |L(f) - \hat{L}_S(f)| \leq \epsilon_{\text{unif},\mathcal{A}}$$

This formulation reveals the following weakness: Even if a classifier generalizes well ($L(f) = 0$), we might still be able to leverage the data dependence of $f \in \mathcal{F}_{\mathcal{A}}$ on a particular draw $\mathcal{S} \in \mathcal{S}_\delta$ to construct a new training set $\mathcal{S}' \in \mathcal{S}_\delta$ as a function of $\mathcal{S}$ for which $\hat{L}_{\mathcal{S}'}(f)$ is big. Nagarajan & Kolter (2019b) construct such an in-distribution adversarial construction $\mathcal{S}'$ for a very simple data distributions and every dataset $\mathcal{S} \in \mathcal{S}_\delta$, resulting in a huge supremum and thus provably vacuous generalization bounds. We will describe said construction in the following.

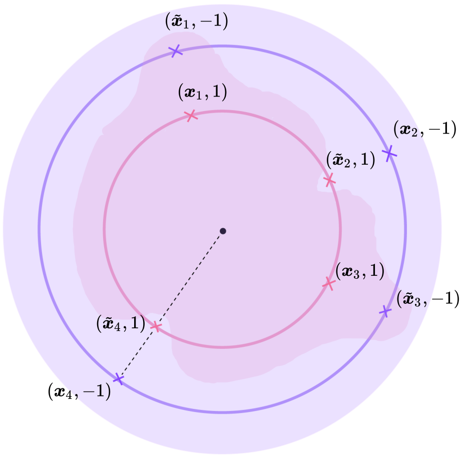
$(\tilde{\boldsymbol{x}}_1, -1)$
$(\boldsymbol{x}_1, 1)$
$(\boldsymbol{x}_2, -1)$
$(\tilde{\boldsymbol{x}}_2, 1)$
$(\boldsymbol{x}_3, 1)$
$(\tilde{\boldsymbol{x}}_3, -1)$
$(\tilde{\boldsymbol{x}}_4, 1)$
$(\boldsymbol{x}_4, -1)$

Figure 1. Schematic visualization of the decision boundary induced by a neural network trained with gradient descent on the 2 dimensional adversarial spheres. Here the training set is $\mathcal{S}_{\text{train}} = \{(\boldsymbol{x}_1, 1), (\boldsymbol{x}_2, -1), (\boldsymbol{x}_3, 1), (\boldsymbol{x}_4, -1)\}$ and adversarial set is $\mathcal{S}_{\text{adv}} = \{(\tilde{\boldsymbol{x}}_1, -1), (\tilde{\boldsymbol{x}}_2, 1), (\tilde{\boldsymbol{x}}_3, -1), (\tilde{\boldsymbol{x}}_4, 1)\}$. Notice how the data distribution is captured correctly except for $\mathcal{S}_{\text{adv}}$ which is misclassified completely.

4.2. Adversarial Spheres

Consider the following simple dataset described by the input data distribution

$$p(\boldsymbol{x}) = q p_{r_1}(\boldsymbol{x}) + (1-q) p_{r_2}(\boldsymbol{x})$$

with $r_1 < r_2$, $0 < q < 1$, and $p_r(\cdot)$ the uniform density over the sphere $\boldsymbol{S}_r^{d-1} = \{\boldsymbol{x} \in \mathbb{R}^d : ||\boldsymbol{x}||_2 = r\}$:

$$p_r(\boldsymbol{x}) = \frac{1}{A_r}\mathbb{1}_{\{\boldsymbol{x}\in\boldsymbol{S}_r^d\}}$$

where A_r denotes the surface area of a $d-1$-dimensional sphere with radius r. Whenever $||\boldsymbol{x}||_2 = r_1$, we label the point as $y_{\boldsymbol{x}} = 1$ and when $||\boldsymbol{x}||_2 = r_2$ we set $y_{\boldsymbol{x}} = -1$. We define the class probabilities as $\mathbb{P}(y = 1) = q$ and $\mathbb{P}(y = -1) = 1 - q$. As before, we refer to p as the input distribution ($\boldsymbol{x} \sim p$) and to $\mathcal{D}$ as the data distribution ($(\boldsymbol{x}, y) \sim \mathcal{D}$).

It will be very important to study how a point $\boldsymbol{x} \sim p$ will determine the behaviour of a model f at the corresponding projection $\tilde{\boldsymbol{x}}$ on the other sphere. To this end we introduce the projection

$$\boldsymbol{x} \mapsto \mathcal{P}(\boldsymbol{x}) = \frac{r_1}{r_2}\boldsymbol{x}\mathbb{1}_{\{\boldsymbol{x}\in\mathbb{S}^d_{r_2}\}} + \frac{r_2}{r_1}\boldsymbol{x}\mathbb{1}_{\{\boldsymbol{x}\in\mathbb{S}^d_{r_1}\}}$$

We will refer to $\mathcal{P}(\boldsymbol{x})$ both as the projection of $\boldsymbol{x}$ as well as the adversarial point of $\boldsymbol{x}$. We call the set

$$\mathcal{S}_{\text{adv}} = \Big\{ (\mathcal{P}(\boldsymbol{x}_i), -y_i) : i = 1, \dots, n \Big\}$$

the adversarial set. Crucially, the distribution of $(\boldsymbol{x}, y)$ remains invariant under $\mathcal{P}$ due to the uniformity on both spheres. As empirically observed in Nagarajan & Kolter (2019b), surprisingly, a neural network trained by gradient descent on $\mathcal{S}$ completely misclassifies $\mathcal{S}_{\text{adv}}$. We depict this phenomena for the 2-dimensional case in Figure 1. As a consequence, any uniform convergence-based bound is rendered vacuous, as outlined in Section 4.1.

This observation questions the validity of uniform convergence as it already fails to explain the generalization on such a simple data distribution for quite generic neural networks. It is thus crucial to understand how this degeneracy in neural networks arises and to determine if the effect is simply a consequence of the particular data distribution or pointing to a deeper problem of neural architectures.

4.3. NNGP and Neural Tangent Kernel

Recently, a novel tool for analyzing neural networks emerged in the form of the NNGP (Lee et al., 2018) and the NTK (Jacot et al., 2018). These works assume a different parametrization of the network by introducing a scaling $\frac{1}{\sqrt{d_l}}$ at each layer. Every weight is initialized according to $W_{ij}^{(l)} \overset{i.i.d.}{\sim} \mathcal{N}(0, 1)$ whereas the bias follows $b_i^{(l)} \sim \mathcal{N}(0, \beta_l^2)$. As shown in Lee et al. (2018), as the widths $d_i \to \infty$ for $i = 1, \dots, L$, the neural network at initialization exhibits a Gaussian process behaviour:

$$f(\cdot) \sim \mathcal{GP}(0, \Sigma^{(L)})$$

governed by the NNGP kernel $\Sigma^{(L)}$ defined recursively as $\Sigma^{(1)}(\boldsymbol{x}, \boldsymbol{x'}) = \frac{1}{\sqrt{d_0}}\boldsymbol{x}^T\boldsymbol{x'} + \beta_l^2$ and

$$\Sigma^{(l)}(\boldsymbol{x}, \boldsymbol{x'}) = \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0}, \tilde{\boldsymbol{\Sigma}}^{(l-1)})}\big[\sigma(z_1)\sigma(z_2)\big] + \beta_l^2$$

for $l = 1, \dots, L$ and where $\tilde{\boldsymbol{\Sigma}}^l \in \mathbb{R}^{2\times 2}$ is obtained from evaluating Σ^l on the set $\{\boldsymbol{x}, \boldsymbol{x}'\}$. Jacot et al. (2018) extended this result by incorporating gradient descent dynamics. They introduced the empirical neural tangent kernel

$$\hat{\Theta}(\boldsymbol{x}, \boldsymbol{x}') = \left(\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x})\right)^T \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}')$$

and showed that in the infinite-width regime, the kernel becomes deterministic and remains constant along the training trajectory induced by gradient flow. Moreover, the limiting kernel, denoted by Θ, has a closed-form expression given by the recursion $\Theta^{(1)}(\boldsymbol{x}, \boldsymbol{x}') = \Sigma^{(1)}(\boldsymbol{x}, \boldsymbol{x}')$ and

$$\Theta^{(L+1)}(\boldsymbol{x}, \boldsymbol{x}') = \Theta^{(L)}(\boldsymbol{x}, \boldsymbol{x}')\dot{\Sigma}^{(L+1)}(\boldsymbol{x}, \boldsymbol{x}') + \Sigma^{(L+1)}(\boldsymbol{x}, \boldsymbol{x}')$$

where $\dot{\Sigma}^{(l)}(\boldsymbol{x}, \boldsymbol{x'}) = \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0}, \tilde{\boldsymbol{\Sigma}}^{(l-1)})}\big[\dot{\sigma}(z_1)\dot{\sigma}(z_2)\big]$. As a consequence, under mean squared loss, neural network training can be viewed as kernel regression and admits a simple formula for a network trained for infinitely long:

$$f_\infty(\boldsymbol{x}) = \left(\Theta^{(L)}(\boldsymbol{x}, \boldsymbol{X})\right)^T \left(\Theta^{(L)}(\boldsymbol{X}, \boldsymbol{X})\right)^{-1} \boldsymbol{y}$$

This convenient formula lends itself better to mathematical analysis, compared to finite-width networks, while still preserving lots of important structure.

Lee et al. (2019) later extended the analysis to gradient descent with a small enough step size and proved that the NNGP can be viewed as the limiting kernel of a neural network with only a trainable last layer. Thus the NNGP can be viewed as a special case of the NTK.

5. Adversarial Spheres and Infinite Width

We now turn to the study of adversarial spheres in the infinite width setting, employing both the NNGP and the NTK. Although clearly a classification task, we will use mean squared loss as often done in the literature (Chen et al., 2020; Arora et al., 2019). For generality and elegance of the argument, we define a general class of kernels that admit a certain property.

Definition 1. *Consider a kernel $K : \mathbb{R}^d \times \mathbb{R}^d$. We call K* ***semi-homogeneous*** *if and only if there exists $\zeta \in \mathbb{R}$ such that $\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$ and $\alpha > 0$, it holds that*

$$K(\alpha\boldsymbol{x}, \boldsymbol{x}') = \alpha K(\boldsymbol{x}, \boldsymbol{x}') + \zeta^2(1 - \alpha)$$

We have the following theorem that shows that both the NTK and the NNGP using ReLU non-linearity belong to the family of semi-homogeneous kernels. Due to its strongly technical nature, we postpone the proof to the appendix.

Theorem 1. *Consider a fully-connected neural network with NTK parametrization as introduced in Section 4.3, equipped with a 1-homogeneous activation function (such as ReLU). Set every bias to zero ($\beta_i = 0$) except for the output bias, $b^{(L)} \sim \mathcal{N}(0, \beta^2)$. Then it holds that both $\Theta^{(L)}$ and $\Sigma^{(L)}$ are semi-homogeneous kernels with $\zeta = \beta$.*

Semi-homogeneous kernels form an interesting family of kernels for the adversarial spheres because one can easily quantify their change under the projection operator $\mathcal{P}$ introduced in Section 4.2, as shown in the following lemma.

Lemma 2. *Fix a semi-homogeneous kernel K and two data points sampled according to the adversarial spheres measure, $\boldsymbol{x}, \boldsymbol{z} \sim p$. Consider the projection $\mathcal{P}(\boldsymbol{x})$. Denote $r = ||\boldsymbol{x}||_2$ and $\tilde{r} = ||\mathcal{P}(\boldsymbol{x})||_2$. Then it holds that:*

$$K(\mathcal{P}(\boldsymbol{x}), \boldsymbol{z}) = \frac{\tilde{r}}{r}K(\boldsymbol{x}, \boldsymbol{z}) + \zeta^2(1 - \frac{\tilde{r}}{r})$$

This offers the following interesting insight. A semi-homogeneous kernel is only affected under the projection $\mathcal{P}$ through the magnitude of the inputs. In the case of the adversarial spheres, the change is thus entirely determined through the label information $y_{\boldsymbol{x}}$ since it is a function of $||\boldsymbol{x}||_2$. The angular component in $\boldsymbol{x}$ is completely irrelevant to the model. This becomes more crucial when studying the predictive function f_K induced by kernel regression with K under mean squared loss:

$$f_K(\boldsymbol{x}) = K(\boldsymbol{x}, \boldsymbol{X})K(\boldsymbol{X}, \boldsymbol{X})^{-1}\boldsymbol{y}$$

Using the insight from the previous lemma, we can relate the prediction of f_K on both $\boldsymbol{x}$ and $\mathcal{P}(\boldsymbol{x})$, which is a crucial step towards understanding how the performance of f_K on $\mathcal{S}_{\text{train}}$ relates to the one on $\mathcal{S}_{\text{adv}}$.

Corollary 2.1. *Fix a semi-homogeneous kernel K and a data point sampled according to the adversarial spheres measure, $\boldsymbol{x} \sim p$. Consider the projection $\mathcal{P}(\boldsymbol{x})$. Denote $r = ||\boldsymbol{x}||_2$ and $\tilde{r} = ||\mathcal{P}(\boldsymbol{x})||_2$. Then it holds that*

$$f_K\left(\mathcal{P}(\boldsymbol{x})\right) = \frac{\tilde{r}}{r} f_K(\boldsymbol{x}) + \zeta^2\left(1 - \frac{\tilde{r}}{r}\right)\gamma_K(n)$$

where we define $\gamma_K(n) = \boldsymbol{1}_n^T K(\boldsymbol{X}, \boldsymbol{X})^{-1}\boldsymbol{y}$ and $\boldsymbol{1}_n = (1, \dots, 1)^T \in \mathbb{R}^n$.

Crucially, $\gamma_K(n)$ is entirely agnostic to the data point $(\boldsymbol{x}, y)$ and solely depends on the kernel K and the training data $\mathcal{S}_{\text{train}}$. Once more, the label information fully determines how f_K will change under the projection.

5.1. Adversarial Accuracy

Equipped with these results we can now turn our attention to the adversarial set $\mathcal{S}_{\text{adv}}$ and the resulting accuracy,

$$a_{\text{adv}} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\left\{\text{sgn}(f_K(\mathcal{P}(\boldsymbol{x}_i)) = -y_i\right\}}$$

coined adversarial accuracy and present our main theoretical findings. In Nagarajan & Kolter (2019b), it was empirically observed that

$$a_{\text{adv}} = 0$$

for a specific neural architecture. Here we are able to prove this phenomenon mathematically for semi-homogeneous kernels and unravel the dependencies of a_{adv} on parameters like the sample size n, the radii r_1, r_2 and the semi-homogeneous parameter ζ.

Theorem 3. *Take a semi-homogeneous kernel K and consider a training set $\mathcal{S}_{train} \overset{i.i.d.}{\sim} \mathcal{D}^n$ along with the corresponding adversarial set $\mathcal{S}_{adv}$. Then it holds that a_{adv} is quantized to only three values:*

$$a_{adv} \in \Big\{0, 1 - q, 1\Big\}$$

Moreover, we can characterize the phase transitions in sample size n as

$$a_{adv} = \begin{cases} 0 & \textit{if} \quad \gamma_K(n) \leq \frac{r_1}{\zeta^2(r_2 - r_1)} \\ 1 - q & \textit{if} \quad \frac{r_1}{\zeta^2(r_2 - r_1)} \leq \gamma_K(n) \leq \frac{r_2}{\zeta^2(r_2 - r_1)} \\ 1 & \textit{if} \quad \gamma_K(n) \geq \frac{r_2}{\zeta^2(r_2 - r_1)} \end{cases}$$

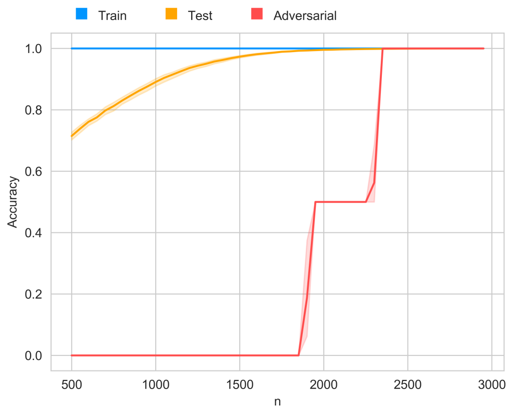

Figure 2. Train, test and adversarial accuracy of 3 layer NTK evaluated on 100-dimensional adversarial spheres, plotted against sample size n. Results are averaged over 8 runs.

We can see that the adversarial accuracy goes through phase transitions governed by $\gamma_K(n)$. We validate this surprising result through numerical experiments, displayed in Figure 2 and Figure 3. We use a two 100-dimensional spheres with radii $r_1 = 1$ and $r_2 = 1.11$, similar to the setup considered in Nagarajan & Kolter (2019b). As predicted, we observe very sharp phase transitions in the adversarial accuracy and they occur exactly at the sample sizes predicted by our theory. Both NNGP and NTK indeed display the effect at small sample sizes but as n increases we recover perfect accuracy.

To gain a better understanding in terms of the sample size n, we need to analyze $\gamma_K(n)$ in more detail.

5.2. Properties of $\gamma_K(n)$

In this section, we restrict our attention to semi-homogeneous kernels of the form

$$K(\boldsymbol{x}, \boldsymbol{x}') = C(\boldsymbol{x}, \boldsymbol{x}') + \beta^2$$

where $C : \mathbb{R}^d \times \mathbb{R}^d$ is a homogeneous kernel, $C(\alpha\boldsymbol{x}, \boldsymbol{x}') = \alpha C(\boldsymbol{x}, \boldsymbol{x}') \; \forall\alpha > 0$. A simple calculation indeed reveals that K is semi-homogeneous. As outlined in the proof of

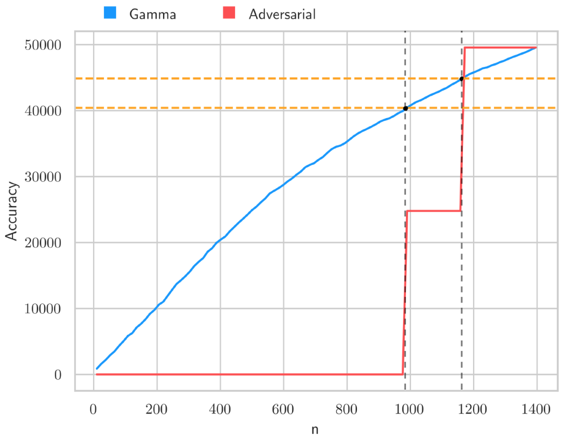

Figure 3. Scaled adversarial accuracy of 3 layer NNGP evaluated on 100-dimensional adversarial spheres, plotted against $\gamma_K(n)$. Horizontal lines indicate predicted phase transitions in γ_K.

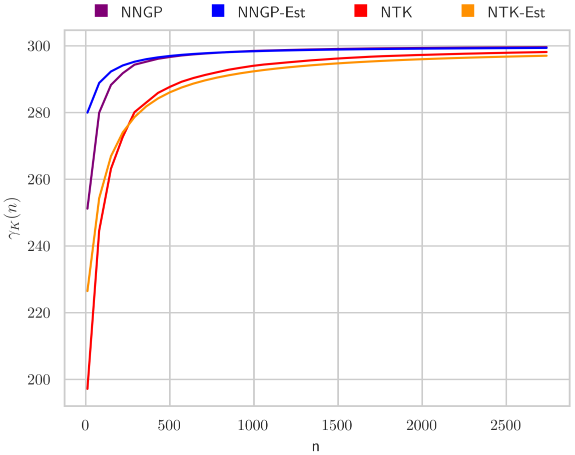

Figure 4. $\gamma_K(n)$ plotted against sample size n for a 1-layer NNGP and NTK along with the corresponding approximation $\gamma_{\tilde{K}}(n)$. Results are averaged over 8 runs.

Theorem 1, this restricted family still includes the NTK and the NNGP with an output bias β. As a first step, we can isolate the role of the semi-homogeneous parameter β.

Lemma 4. *Assume that K is of the above form and denote by C the corresponding homogeneous kernel. Then it holds that*

$$\gamma_K(n) = \frac{1}{1+\beta^2 s\left(C(\boldsymbol{X},\boldsymbol{X})^{-1}\right)}\gamma_C(n)$$

where we define $s(\boldsymbol{A}) = \sum_{i,j} A_{ij}$.

Let us assume in the following that for simplicity, $q = \frac{1}{2}$ and that we have a balanced training dataset. Define $2m = n$ for $m \in \mathbb{N}$. Moreover, without loss of generality, we permute the order of the training samples such that the first m entries in $\boldsymbol{y}$ correspond to the positive class ($y = 1$) and the last m entries to the negative class ($y = -1$). To get qualitative insights into γ_K, we analyze the behaviour in expectation over the dataset. With a slight abuse of notation, we define $\gamma_{\boldsymbol{A}}(n) = \mathbf{1}_n^T \boldsymbol{A}^{-1}\boldsymbol{y}$ for $\boldsymbol{A} \in \mathbb{R}^{n\times n}$.

Theorem 5. *Consider the expected kernel $\tilde{K} = \mathbb{E}_{\boldsymbol{X}\sim p^n}\left[K(\boldsymbol{X},\boldsymbol{X})\right]$. We have that $\gamma_{\tilde{K}}$ is asymptotically given by*

$$\gamma_{\tilde{K}}(n) \propto \frac{C_1 + \eta n}{C_2 - \beta^2 C_3 n}$$

for constants $C_1, C_2, C_3, \eta \in \mathbb{R}$ and the limit is given by

$$\gamma_{\tilde{K}}(n) \xrightarrow{n\to\infty} \frac{r_1 + r_2}{\beta^2(r_2 - r_1)}$$

Surprisingly, the limiting capacity is independent of the particular kernel except for its semi-homogeneous parameter β. Moreover, as intuitively expected, $\gamma_{\tilde{K}}(n)$ is an increasing function. As a consequence a model will experience the phase transitions outlined in Theorem 3 in sequence. We verify our predictions numerically by plotting γ_K for different kernels and comparing them with the averaged case in Figure 4. We can readily see that the kernel in expectation is a good approximation and provides a tight fit especially for moderately large to large n. We provide more numerical evidence in the Appendix B.2.

5.3. The Role of the Bias β^2

In this section we will study the influence of the output bias β for fixed sample sizes n. Again we restrict the analysis to kernels of the form $K(\boldsymbol{x},\boldsymbol{x}') = C(\boldsymbol{x},\boldsymbol{x}') + \beta^2$. Due to Lemma 4, studying the behaviour of a_{adv} for varying bias β^2 but fixed sample size n now becomes feasible. One can easily see that

$$g(\beta) = \beta^2\gamma_K(n) = \frac{\beta^2}{1+\beta^2 s\left(C(\boldsymbol{X},\boldsymbol{X})^{-1}\right)}\gamma_C(n)$$

is an increasing function in β. As a consequence, for a fixed sample size n, also a_{adv} is increasing in β. We can calculate the capacity limit as

$$\beta^2\gamma_K(n) \xrightarrow{\beta\to\infty} \frac{\gamma_C(n)}{s\left(C\left(\boldsymbol{X},\boldsymbol{X}\right)^{-1}\right)}$$

Thus an increasing bias leads to better robustness in terms of the adversarial accuracy but there is an upper limit to the benefit. Depending on this capacity limit, a big enough bias potentially leads to a perfect adversarial accuracy. As a result, a simple increase in the bias of the network could potentially mitigate the problem entirely. We verify our results again through numerical experiments. We fix the sample size $n \in \mathbb{N}$ such that for small bias β we observe a strong adversarial effect. We then vary β and show the test, train and adversarial accuracy as a function of β in Figure 5. Again we observe sharp phase transitions in the

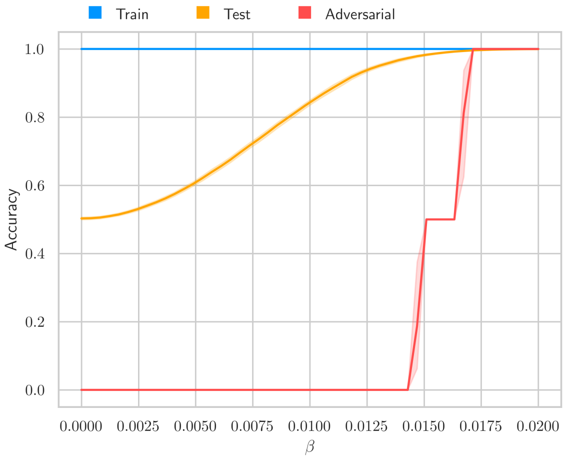

Figure 5. Different accuracies for a 2-layer NNGP model plotted against bias β for fixed sample size. Results are averaged over 8 runs.

adversarial accuracy as well as an increase in generalization. Indeed, a bigger output bias alleviates the adversarial effect completely without any increase in sample size. Moreover, although our theory only holds for the infinite width case, we observe the same phenomenon for finite-width networks trained with gradient descent under mean squared error. In Figure 6 we show the accuracies of a 2 hidden layer network of width 1000 plotted against different bias initialization magnitudes. Again we observe the same phase transitions in the adversarial accuracy.

6. Decomposition of Neural Network

We have identified sharp phase transitions in a_{adv} and that a simple increase in the bias β can completely mitigate the problem. In this section we explore an alternative approach, advocated by Nagarajan & Kolter (2019b), given by decompositions of the network into a clean part f_{clean} and a noisy part f_{noisy} such that

$$f(\boldsymbol{x}) = f_{\text{clean}}(\boldsymbol{x}) + f_{\text{noisy}}(\boldsymbol{x})$$

Ideally, f_{clean} would capture the good generalization capability of f while being more robust against the adversarial effect. Instead of analyzing f, one could study f_{clean} with tools based on uniform convergence. Here we study the canonical decomposition of the network, induced by the eigenfunctions of the kernel K. We show empirically that such a decomposition does not alleviate the adversarial effect and that sufficient bias is still necessary.

6.1. Eigendecomposition of Kernel

Consider the Mercer decomposition of a kernel K:

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

where (ϕ, λ) is an eigenfunction-eigenvalue pair of the Fredholm integral operator

$$T_K^p : H \to L^2 \, , \phi \mapsto \int_{\mathbb{R}} K(\boldsymbol{x}, \boldsymbol{z}) \phi(\boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z}$$

where p denotes the input data measure and H is some function space. The study of the eigenfunctions of T_K for dot-product kernels has been mainly limited to the uniform measure over a single sphere (Basri et al., 2019; Bietti & Mairal, 2019). Recently, Basri et al. (2020) have extended this analysis to a piece-wise constant density on the sphere. In order to analyze eigendecompositions for the adversarial spheres, we need to understand the spectral properties of T_K^p. It turns out that for semi-homogeneous dot-product kernels K, one can extend the eigenanalysis to the more general class of isotropic distributions (see Theorem 5 in (Geifman et al., 2020)):

Theorem 6. *Consider an input distribution $p(\boldsymbol{x})$ such that the conditional distribution $p(\boldsymbol{x} \big| ||\boldsymbol{x}||_2 = r)$ is the uniform measure over $\boldsymbol{S}_r^{d-1}$. Denote by $p_R(r)$ the distribution of $||\boldsymbol{x}||_2$ and by p_1 the uniform measure over the sphere. Fix an eigenfunction eigenvalue pair $(\tilde{\phi}, \tilde{\lambda})$ of $T_K^{p_1}$. Then we can express the eigenfunction eigenvalue pairs (ϕ, λ) of T_K^p as*

- $\tilde{\phi}(\boldsymbol{x}) = \frac{1}{\sqrt{\mathbb{E}_{R \sim p_R}[R^2]}} \phi(\boldsymbol{x})$
- $\tilde{\lambda} = \lambda \mathbb{E}_{R \sim p_R} \left[R^2 \right]$

This theorem relates the eigenfunctions associated with the isotropic measure directly to the eigenfunctions of the rather well-understood uniform measure on the sphere. As shown for instance in Basri et al. (2019); Bietti & Mairal (2019), the eigenfunctions of the NTK and NNGP are given by the spherical harmonics. The eigenvalues are trickier to study and depend on the structure of the employed kernel K. Some specific architectures such as one hidden layer networks do admit analytic expressions (Basri et al., 2019).

In particular, for the adversarial spheres, we observe that

$$p_R(r) = q\delta_{r_1}(r) + (1-q)\delta_{r_2}(r)$$

where $\delta_z(x)$ denotes a Dirac Delta centered at z. We can thus easily calculate $\mathbb{E}_{R \sim p_R}[R^2] = qr_1^2 + (1-q)r_2^2 > 0$ and hence conclude that the number of non-zero eigenvalues are the same as for the uniform measure over $\boldsymbol{S}_1^{d-1}$.

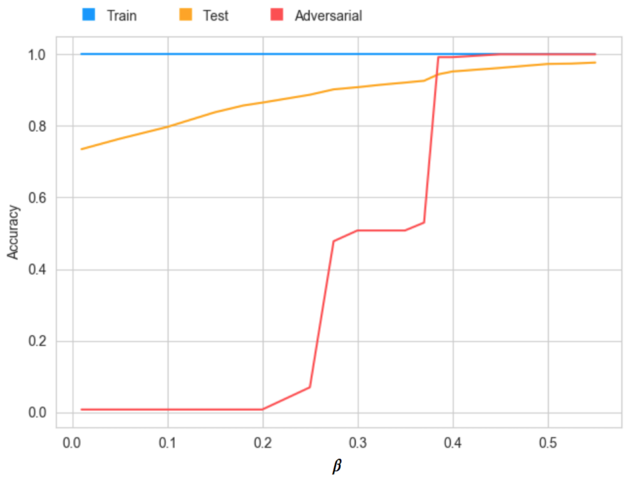

Figure 6. Different accuracies for a 2-layer neural network of width 1000 plotted against bias β for fixed sample size, trained with gradient descent under MSE.

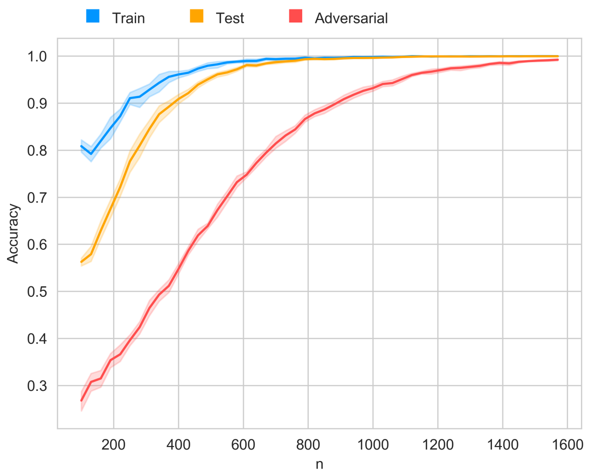

Figure 7. Train, test and adversarial accuracies plotted against sample size for the dominant eigenfunction of a 2-layer NNGP model. Results are averaged over 8 runs.

6.2. Canonical Decomposition of Predictive Function

We want to investigate the question whether using a decomposition of the neural networks induced by the Mercer decomposition can alleviate the problem encountered in adversarial spheres. The eigenfunctions associated with the integral operator however are not the ideal arena to study the problem as they are infinite sample quantities, stemming from the complete knowledge of the data distribution. The adversarial effect on the other hand is a finite sample effect that starts to vanish as the sample size n increases, as seen in the previous section. As a result, we instead study finite-sample estimators of the eigenfunctions and eigenvalues.

Consider the spectral decomposition of the kernel matrix

$$K(\boldsymbol{X}, \boldsymbol{X}) = \boldsymbol{V}\,\mathrm{diag}(\boldsymbol{\mu})\boldsymbol{V}^T \in \mathbb{R}^{n\times n}$$

where $\boldsymbol{\mu} \in \mathbb{R}^n$ are the eigenvalues and $\boldsymbol{V} \in \mathbb{R}$ contains the associated eigenvectors. Using these quantities, we can form estimators of the eigenfunction ϕ and eigenvalues λ as follows:

- $\hat{\lambda}_i = \frac{1}{n}\mu_i$
- $\hat{\phi}_i(\boldsymbol{x}) = \frac{1}{\mu_i}\sum_{k=1}^{n} V_{ki}K(\boldsymbol{x}_k, \boldsymbol{x})$

We refer to Baker (1977) for an in-depth treatment of these finite approximations to Fredholm integral problems. These estimators in turn induce a decomposition on the predictive function at any finite sample size:

Lemma 7. *Consider any kernel K and its associated predictive function f_K. We can decompose f_K into its different spectral components*

$$f_K(\boldsymbol{x}) = \sum_{k=1}^{n} \left(\boldsymbol{v}_k^T\boldsymbol{y}\right)\hat{\phi}_k(\boldsymbol{x})$$

where $\boldsymbol{v}_k$ denotes the k-th eigenvector of $K(\boldsymbol{X}, \boldsymbol{X})$.

Essentially, $\boldsymbol{v}_i^T\boldsymbol{y}$ measures the importance of the eigenfunction $\hat{\phi}_i$ to the task. Eigenvectors that are well-aligned with the targets $\boldsymbol{y}$ will contribute more to the prediction while orthogonal eigenvectors will not be considered. This decomposition gives rise to very natural splittings of the form

$$f_K(\boldsymbol{x}) = \underbrace{\sum_{k\in\mathcal{I}} \left(\boldsymbol{v}_k^T\boldsymbol{y}\right)\hat{\phi}_k(\boldsymbol{x})}_{f_{\text{clean}}(\boldsymbol{x})} + \underbrace{\sum_{k\notin\mathcal{I}} \left(\boldsymbol{v}_k^T\boldsymbol{y}\right)\hat{\phi}_k(\boldsymbol{x})}_{f_{\text{noisy}}(\boldsymbol{x})}$$

where $\mathcal{I} \subset \{1, \dots, n\}$ is an index set which can be varied. we will study numerically how restricting the full predictive function to such a subset of eigenfunctions might improve the adversarial accuracy, for a fixed small bias β. We refer to $\hat{\phi}_i$ with $i = \mathrm{argmax}_{1\le j\le n}|\boldsymbol{v}_j^T\boldsymbol{y}|$ as the dominant eigenfunction. We study the decomposition

$$f_{\text{clean}}(\boldsymbol{x}) = \hat{\phi}_i(\boldsymbol{x})$$

Interestingly, f_{clean} perfectly captures the data distribution, as illustrated in Figure 7, visible in the perfect training and test accuracy. It however does not alleviate the adversarial effect completely as it persists for small sample sizes and only very slowly converges. We study different combinations of eigenfunctions in the Appendix B.3 but none can improve over the dominant eigenfunction in terms of adversarial accuracy. Again, only an increase in the output bias β can remove the degeneracy, highlighting once more the simple nature of the problem.

7. Discussion

In this work, we provide a mathematical account of the adversarial phenomenon observed in Nagarajan & Kolter

(2019b). We identified its origin, pin-pointing it to the output bias of the model which trades-off how much a network relies on radial information in the data. We studied the different phase transitions in the adversarial accuracy and linked them to a data-dependent quantity $\gamma_K(n)$ which we derived in closed-form for the expected kernel. Moreover, we studied how the adversarial effect behaves under eigendecompositions and showed numerically that even a restriction to the ideal eigenfunction does not alleviate the problem. The adversarial effect thus really is a consequence of the data distribution solely containing radial information, which in turn makes a neural network vulnerable if the output bias is not large enough. The problem observed in Nagarajan & Kolter (2019b) does hence not point towards a deeper problem in the design of neural models or the optimizer and does not translate to other datasets directly.

References

Arora, S., Ge, R., Neyshabur, B., and Zhang, Y. Stronger generalization bounds for deep nets via a compression approach. *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R., and Wang, R. On exact computation with an infinitely wide neural net. *33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Baker, C. The numerical treatment of integral equations. *Clarendon press Oxford, volume 13*, 1977.

Bartlett, P., Foster, D. J., and Telgarsky, M. Spectrally-normalized margin bounds for neural networks. *31st Conference on Neural Information Processing Systems (Neurips)*, 2017.

Bartlett, P. L., Harvey, N., Liaw, C., and Mehrabian, A. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research 20*, pp. 1–17, 2019.

Basri, R., Jacobs, D., Kasten, Y., and Kritchman, S. The convergence rate of neural networks for learned functions of different frequencies. *33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Basri, R., Galun, M., Geifman, A., Jacobs, D., Kasten, Y., and Kritchman, S. Frequency bias in neural networks for input of non-uniform density. *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Belkin, M., Ma, S., and Mandal, S. To understand deep learning we need to understand kernel learning. *Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80*, 2018.

Bietti, A. and Mairal, J. On the inductive bias of neural tangent kernels. *33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Chen, S., He, H., and Su, W. J. Label-aware neural tangent kernel: Toward better generalization and local elasticity. *34th Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT*, 2019.

Du, S. S., Hou, K., Póczos, B., Salakhutdinov, R., Wang, R., and Xu, K. Graph neural tangent kernel: Fusing graph neural networks with graph kernels. *34rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Dziugaite, G. K. and Roy, D. M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*, 2017.

Geifman, A., Yadav, A., Kasten, Y., Galun, M., Jacobs, D., and Basri, R. On the similarity between the laplace and neural tangent kernels. *34th Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Huang, K., Wang, Y., Tao, M., and Zhao, T. Why do deep residual networks generalize better than deep feedforward networks? – a neural tangent kernel perspective. *34rd Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *32rd Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *Proceedings of the 5th International Conference on Learning Representations*, 2017.

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

Lee, J., Xiao, L., Schoenholz, S. S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. *33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Nagarajan, V. and Kolter, J. Z. Deterministic pac-bayesian generalization bounds for deep networks via generalizing noise-resilience. *International Conference on Learning Representations (ICLR)*, 2019a.

Nagarajan, V. and Kolter, J. Z. Uniform convergence may be unable to explain generalization in deep learning. *33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019b.

Negrea, J., Dziugaite, G. K., and Roy, D. M. In defense of uniform convergence: Generalization via derandomization with an application to interpolating predictors. *Proceedings of the 37th International Conference on Machine Learning (PMLR)*, 2020.

Neyshabur, B., Tomioka, R., and Srebro, N. Norm-based capacity control in neural networks. *Proceedings of The 28th Conference on Learning Theory (PMLR)*, 2015.

Neyshabur, B., Bhojanapalli, S., and Srebro, N. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *International Conference on Learning Representations (ICLR)*, 2018.

Zhou, W., Veitch, V., Austern, M., Adams, R. P., and Orbanz, P. Non-vacuous generalization bounds at the imagenet scale: A pac-bayesian compression approach. *International Conference on Learning Representations (ICLR)*, 2019.