A Deep Conditioning Treatment of Neural Networks

Naman Agarwal   NAMANAGARWAL@GOOGLE.COM
Google AI, Princeton

Pranjal Awasthi   PRANJALAWASTHI@GOOGLE.COM
Google Research

Satyen Kale   SATYENKALE@GOOGLE.COM
Google Research

Editors: Vitaly Feldman, Katrina Ligett and Sivan Sabato

Abstract

We study the role of depth in training randomly initialized overparameterized neural networks. We give a general result showing that depth improves trainability of neural networks by improving the *conditioning* of certain kernel matrices of the input data. This result holds for arbitrary non-linear activation functions under a certain normalization. We provide versions of the result that hold for training just the top layer of the neural network, as well as for training all layers, via the neural tangent kernel. As applications of these general results, we provide a generalization of the results of Das et al. (2019) showing that learnability of deep random neural networks with a large class of non-linear activations degrades exponentially with depth. We also show how benign overfitting can occur in deep neural networks via the results of Bartlett et al. (2019b). We also give experimental evidence that normalized versions of ReLU are a viable alternative to more complex operations like Batch Normalization in training deep neural networks.

1. Introduction

Deep neural networks have enjoyed tremendous empirical success, and it has become evident that depth plays a crucial role in this success (Simonyan and Zisserman, 2014; Szegedy et al., 2015; He et al., 2016a). However, vanilla deep networks are notoriously hard to train without some form of intervention aimed to improve the optimization process, for example, Batch Normalization (Ioffe and Szegedy, 2015), Layer Normalization (Ba et al., 2016), or skip connections in Resnets (He et al., 2016b). Recent theory (Santurkar et al., 2018; Balduzzi et al., 2017) has shed some light into how these interventions help train deep networks, especially for the widely popular Batch Normalization operation. However the picture is far from clear in light of work such as (Yang et al., 2019) which argues that Batch Normalization actually *hinders* training by causing gradient explosion.

In this paper, we investigate *improved data conditioning* as a possible factor in explaining the benefits of the aforementioned interventions for training deep neural networks. While standard optimization theory tells us that good data conditioning leads to faster training, empirical performance on test data also seems to be correlated with good conditioning. Figure 1 presents examples of various deep network architectures trained on the CIFAR-10 dataset using standard techniques such as batch normalization, layer normalization, and a new normalized version of the ReLU activation that we propose in this work. In each case, as the generalization performance increases with the number of epochs, the average normalized dot products between test inputs decrease as well, indicating improved conditioning. This begs the question of whether depth helps in improving conditioning of the data, and as a result affecting optimization and generalization in deep neural networks.
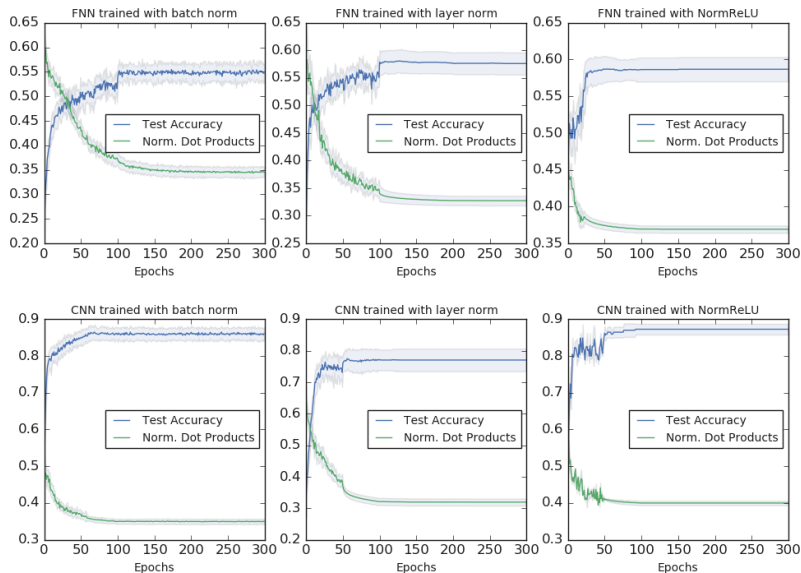
Figure 1: Test Accuracy and average normalized dot products vs. the number of epochs. The top three plots concern depth-32 feed forward networks with 3000 hidden units per layer. The top left network is trained using batch normalization, the middle network is trained using layer normalization, and the top right network is trained using a normalized version of the ReLU activation. The bottom three plots concern convolutional networks with the ResNet-32 architecture (He et al., 2016a). The bottom left network is trained using batch normalization, the middle network is trained using layer normalization and the bottom right network is trained using the normalized ReLU activation.

We elucidate the role of depth and the non-linearity of activations in improving data conditioning by considering a simple intervention: viz., we *normalize* the activations so that when fed standard Gaussian inputs, the output has zero mean and unit variance. Any standard activation function like ReLU, tanh, etc. can be normalized by centering and scaling it appropriately. Thus normalization of activations is a rather benign requirement (see also Lemma 1), but has significant consequences for improving data conditioning theoretically and trainability empirically, as explained next.

1.1. Our contributions.

1. **Exponentially improving data conditioning.** We show that for a randomly initialized neural network with an *arbitrary non-linear normalized* activation function, the condition number of the certain kernel matrices of the input data tend to the best possible value, 1, *exponentially* fast in the depth of the network. The rate at which the condition number tends to 1 is determined by a coefficient of non-linearity of the activation function, a concept that we define in this paper. This result holds for either training just the top layer of the neural network, or all layers of the network with a sufficiently small learning rate (the so-called lazy training regime (Chizat and Bach, 2018)).

2. **Fast training.** Our main result implies that when training large width neural networks of sufficient depth, gradient descent with square loss approaches ϵ training error at a $\log(1/\epsilon)$ rate, *regardless* of the initial conditioning of the data. This is in contrast to prior works (Arora et al., 2019c; Allen-Zhu et al., 2018) and demonstrates the optimization benefits of using deeper networks.

3. **Hardness of learning random neural networks.** Via our main result, we generalize the work of Das et al. (2019) and show that learning a target function that is a sufficiently deep randomly initialized neural network with a general class of activations, requires exponentially (in depth) many queries in the statistical query model of learning. Furthermore, this result holds with constant probability over the random initialization, a considerable strengthening of the prior result. See Section 5 for the formal result and a detailed comparison.

4. **Benign overfitting in deep neural networks.** We extend the work of Bartlett et al. (2019b) on interpolating classifiers and show that randomly initialized and sufficiently deep neural networks can not only fit the training data, but in fact, the minimum norm (in the appropriate RKHS) interpolating solution can achieve non-trivial excess risk guarantees as well.

5. **Empirical benefits of normalized activations.** Guided by our theoretical results, we propose a new family of activation functions called NormReLU which are normalized versions of the standard ReLU activation. Incorporating NormReLU into existing network architectures requires no overhead. Furthermore, we show via experiments on the CIFAR-10 dataset that NormReLU can serve as an effective replacement for techniques such as batch normalization and layer normalization. This leads to an alternate method for training deep networks with no loss in generalization performance and in some cases leads to significant gains in training time.

1.2. Related work

There are two very recent works with similar results to ours independently of our work. The first is the work of Xiao et al. (2019), which uses the tools of mean-field theory of deep neural network developed in a long line of work (Poole et al., 2016; Daniely, 2017a; Schoenholz et al., 2017; Pennington et al., 2017). This work considers a broader spectrum of initialization schemes and activation functions and studies the effect of depth on data conditioning. Specializing to the setting of our paper, this work shows that if the inputs are already very well-conditioned, then they converge to perfect conditioning exponentially fast in the depth of the network. In contrast, our results show exponential convergence even if the inputs are very poorly conditioned: in fact, for some activations like normalized ReLU, the initial condition number could even be infinite. The second is the work of Panigrahi et al. (2020), whose main motivation is studying the effect of smooth vs. non-smooth activation functions in shallow networks. However they do show a very similar exponential convergence result like ours for a kernel matrix closely related to the top-layer kernel matrix in this paper for a more restricted class of activation functions than considered in our paper. Furthermore the results of Panigrahi et al. (2020) assume unit length inputs, whereas in this paper we extend our results for some activations to non-unit length inputs as well. Neither paper considers the applications to optimization, SQ learning of random neural networks, and benign overfitting as done in this paper.

On the optimization side, a sequence of papers has recently shown the benefits of overparametrization via large *width* for training neural networks: see, for example, (Li and Liang, 2018; Du et al., 2019; Allen-Zhu et al., 2019; Zou and Gu, 2019) and the references therein. These papers show that with sufficiently large width, starting from a random initialization of the network weights, gradient descent provably finds a global minimizer of the loss function on the training set. While several of the aforementioned papers do analyze deep neural networks, to our knowledge, there is no prior work that provably demonstrates the benefits of depth for training neural networks in general settings.

Prevailing wisdom is that while depth enables the network to express more complicated functions (see, for example, (Eldan and Shamir, 2016; Telgarsky, 2016; Raghu et al., 2017; Lee et al., 2017; Daniely, 2017a) and the references therein), it hinders efficient training, which is the primary concern in this paper. Indeed, the papers mentioned earlier showing convergence of gradient descent either assume very shallow (one hidden layer) networks, or expend considerable effort to show that depth doesn't *degrade* training by more than a polynomial factor. In contrast, we show that after a certain threshold depth (which depends logarithmically on δ, initial separation), increasing depth improves the convergence rate exponentially.

To provide one such precise comparison, the work of Allen-Zhu et al. (2019) under the same separation assumption as us, proves that overparametrized networks with ReLU activations converge in time polynomial in depth, $1/\delta$ and $\log(1/\epsilon)$. Our results show that if depth is $\Omega(\log(1/\delta))$, then the convergence rate is only proportional to $\log(1/\epsilon)$, *independent* of δ, if one uses normalized activations.

A few exceptions to the above line of work are the papers (Arora et al., 2018b, 2019a) which do show that depth helps in training neural networks, but are restricted to very specific problems with linear activations.

See Appendix A for an in-depth discussion of these and other related works.

2. Notation and preliminaries

For two vectors x and x' of like dimension, we denote their inner product by $x \cdot x'$. Unless otherwise specified, $\|\cdot\|$ denotes the Euclidean norm for vectors and the spectral norm for matrices. For a symmetric positive definite matrix M, the condition number $\kappa(M)$ is defined to be the ratio $\frac{\lambda_{\max}(M)}{\lambda_{\min}(M)}$, where $\lambda_{\max}(M)$ and $\lambda_{\min}(M)$ are the largest and smallest eigenvalues respectively of M. For a positive integer n, define $[n] = \{1, 2, \ldots, n\}$.

We are given a training set of n examples: $S = \{(x_i, y_i) \in \mathbb{R}^d \times \mathcal{Y}\}_{i=1}^n$, where $\mathcal{Y}$ is the output space. For simplicity we begin by assuming, as is standard in related literature, that for all i we have $\|x_i\| = 1$. We provide extensions of our results to non-unit-length inputs in Section 3.5. Let $\underline{K} \in \mathbb{R}^{n \times n}$ be the Gram matrix of the training data, i.e. $\underline{K}_{ij} = x_i \cdot x_j$. We make the following (very standard in the literature, see e.g. (Allen-Zhu et al., 2018; Zou and Gu, 2019)) assumption on the input data:

Assumption A *For all $i, j \in [n]$ with $i \neq j$, we have $|x_i \cdot x_j| \leq 1 - \delta$.*

To keep the presentation as clean as possible, we assume a very simple architecture of the neural network[1]: it has L hidden fully-connected layers, each of width m, and takes $x \in \mathbb{R}^d$ as input and outputs $y \in \mathbb{R}$, with activation function $\sigma : \mathbb{R} \to \mathbb{R}$ to $\mathbb{R}^m$ by entry-wise application. The network can thus be defined as the following function[2] $f_{\vec{W}} : \mathbb{R}^d \to \mathbb{R}$:

$$f_{\vec{W}}(x) = v \cdot \tfrac{1}{\sqrt{m}} \sigma(W_L \tfrac{1}{\sqrt{m}} \sigma(W_{L-1} \cdots \tfrac{1}{\sqrt{m}} \sigma(W_1 x) \cdots)), \tag{1}$$

where $W_1 \in \mathbb{R}^{m \times d}$, $W_2, \ldots, W_L \in \mathbb{R}^{m \times m}$ denote the weight matrices for the hidden layers, $v \in \mathbb{R}^m$ denotes the weight vector of the output layer, $\vec{W} \in \mathbb{R}^{dm+(L-1)m^2+m}$ denotes a vector obtained by

---

1. Extending our analysis to layers of different sizes and outputs of length greater than 1 poses no mathematical difficulty and is omitted for the sake of clarity of notation.
2. Note that we're using the so-called neural tangent kernel parameterization (Jacot et al., 2018) instead of the standard parameterization here.

concatenating vectorizations of the weight matrices. We use the notation $\mathcal{N}(\mu, \Sigma)$ for the normal distribution with mean μ and covariance Σ. All weights are initialized to independent, standard normal variables (i.e. drawn i.i.d. from $\mathcal{N}(0, 1)$).

Our analysis hinges on the following key normalization assumption on σ:

$$\underset{X \sim \mathcal{N}(0,1)}{\mathbb{E}} \left[\sigma(X) \right] = 0 \ \text{ and } \ \underset{X \sim \mathcal{N}(0,1)}{\text{Var}} \left[\sigma(X) \right] = \underset{X \sim \mathcal{N}(0,1)}{\mathbb{E}} \left[\sigma^2(X) \right] = 1. \tag{2}$$

This normalization requirement is rather mild since any standard activation function can be easily normalized by centering it by subtracting a constant and scaling the result by a constant. The only somewhat non-standard part of the normalization is the requirement that the activation is centered so that its expectation on standard normal inputs is 0. This requirement can be relaxed (see Section 3.4) at the price of worse conditioning. Furthermore, the following lemma proved in Appendix B shows that in the presence of other normalization techniques, normalized activations may be assumed without loss of generality:

Lemma 1 *If the neural network in (1) incorporates batch normalization in each layer, then the network output is the same regardless of whether the activation σ is normalized or not. The same holds if instead layer normalization is employed, but on the post-activation outputs rather than pre-activation inputs.*

Throughout the paper, statements of the type "If $q = \Theta(r)$ then [consequence]." should be taken to mean that there exist universal constants c_1, c_2 such that if $c_1 \cdot r \le q \le c_2 \cdot r$ then [consequence] follows. We use $O(\cdot)$ and $\Omega(\cdot)$ notation in a similar manner. Similarly, statements of the type "If $q = \text{poly}(\cdot)$ then [consequence]." should be taken to mean that there exists a polynomial of bounded degree in the arguments such that if q equals that polynomial then [consequence] follows.

3. Main results on conditioning of kernel matrices

3.1. Top layer kernel matrix.

The first kernel matrix we study is the one defined by (random) feature mapping generated at the top layer by the lower layer weights, i.e.[3] $\Phi_{\vec{W}}(x) := \frac{1}{\sqrt{m}} \sigma(W_L \frac{1}{\sqrt{m}} \sigma(W_{L-1} \cdots \frac{1}{\sqrt{m}} \sigma(W_1 x) \cdots))$. The feature mapping $\Phi_{\vec{W}}$ defines a kernel function k and the associated $n \times n$ kernel matrix K on a training set S as $K_{ij} := k(x_i, x_j)$ where $k(x, x') := \Phi_{\vec{W}}(x) \cdot \Phi_{\vec{W}}(x')$.

The main results on conditioning in this paper are cleanest to express in the limit of infinite width neural networks, i.e. $m \to \infty$. In this limit, the kernel function k and the kernel matrix K, tend almost surely to *deterministic* limits (Daniely et al., 2016), denoted as $\bar{k}$ and $\bar{K}$ respectively. We study the conditioning of $\bar{K}$ next. The rate at which the condition number of $\bar{K}$ improves with depth depends on the following notion of degree of non-linearity of the activation function σ:

Definition 2 *The* coefficient of non-linearity *of the activation function σ is defined to be $\mu :=$ $1 - \left(\mathbb{E}_{X \sim \mathcal{N}(0,1)}[X \sigma(X)] \right)^2$.*

The normalization (2) of the activation function implies via Lemma 20 (in Appendix C, where all missing proofs of results in this section can be found) that for any non-linear activation function σ, we have $0 < \mu \le 1$. To state our main result, it is convenient to define the following quantities: for

---

3. Note that $\Phi_{\vec{W}}$ does not depend on the v component of $\vec{W}$; this notation is chosen for simplicity.

any $\nu \in (0, 1]$, $\delta \in (0, 1)$ and a positive integer L, let $L_0(\delta) = \max\left\{\left\lceil \frac{\log(\frac{1}{2\delta})}{\log(1+\frac{\nu}{2})} \right\rceil, 0\right\} = O\left(\frac{\log(1/\delta)}{\nu}\right)$, and define

$$B_\nu(L, \delta) := \begin{cases} 1 - \delta(1 + \frac{\nu}{2})^L & \text{if } L \leq L_0(\delta) \\ \frac{1}{2}(1 - \frac{\nu}{2})^{L-L_0(\delta)} & \text{if } L > L_0(\delta). \end{cases}$$

For clarity of notation, we will denote by $B(L, \delta)$ the quantity $B_\mu(L, \delta)$. We are now ready to state our main result on conditioning of the kernel matrix:

Theorem 3   *Under Assumption A, we have $|\bar{K}_{ij}| \leq B(L, \delta)$ for all $i, j \in [n]$ with $i \neq j$.*

The following corollary is immediate, showing that the condition number of the kernel matrix $\bar{K}$ approaches the smallest possible value, 1, *exponentially* fast as depth increases.

Corollary 4   *Under Assumption A, if $L \geq L_1(\delta) := \left\lceil \frac{\log(n)}{-\log(1-\frac{\mu}{2})} \right\rceil + L_0(\delta)$, then $\kappa(\bar{K}) \leq 1 + 2n(1 - \frac{\mu}{2})^{L-L_1(\delta)}$.*

3.2. Neural tangent kernel matrix.

The second kernel matrix we study arises from the *neural tangent kernel*, which was introduced by Jacot et al. (2018). This kernel matrix naturally arises when all the layers of the neural network are trained via gradient gradient. For a given set of network weights $\vec{W}$, the neural tangent kernel matrix $\mathcal{K} \in \mathbb{R}^{n \times n}$ is defined as $\mathcal{K}_{ij} = (\partial_{\vec{W}} f_{\vec{W}}(x_i)) \cdot (\partial_{\vec{W}} f_{\vec{W}}(x_j))$. As in the previous section, as the width m of the hidden layers tends to infinity, the random $\mathcal{K}$ tends to a deterministic limit, $\bar{\mathcal{K}}$. For this infinite width limit, we have the following theorem analogous to part 1 of Theorem 3:

Theorem 5   *The diagonal entries of $\bar{\mathcal{K}}$ are all equal. Assume that $L \geq 2L_0(\delta)$. Under Assumption A, we have $|\bar{\mathcal{K}}_{ij}| \leq 2B(L/2, \delta) \cdot \bar{\mathcal{K}}_{11}$ for all $i, j \in [n]$ with $i \neq j$.*

The following corollary, analogous to Corollary 4, is immediate:

Corollary 6   *Under Assumption A, if $L \geq L_2(\delta) := \left\lceil \frac{2\log(2n)}{-\log(1-\frac{\mu}{2})} \right\rceil + 2L_0(\delta)$, then $\kappa(\bar{\mathcal{K}}) \leq 1 + 4n(1 - \frac{\mu}{2})^{L/2-L_2(\delta)}$.*

3.3. Better conditioning under stronger assumption

The following somewhat stronger assumption than Assumption A leads to a better conditioning result:

Assumption B $\lambda_{\min}(\underline{K}) \geq \delta$.

While Assumption B implies Assumption A, it is still quite benign, and is easily satisfied if $n \geq d$ and there is even a tiny amount of inherent white noise in the data. Furthermore, as discussed in Appendix H, for certain activations like ReLU, the representations derived after passing a dataset satisfying Assumption A through one layer satisfy Assumption B.

We have the following stronger versions of Theorem 3 and Corollary 4 under Assumption B (all proofs appear in Appendix C):

Theorem 7   *Under Assumption B, we have $\lambda_{\min}(\bar{K}) \geq 1 - B(L, \delta)$.*

Corollary 8 *Under Assumption B, we have $\kappa(\bar{K}) \leq 1 + \frac{n}{\delta}(1 + \frac{\mu}{2})^{-L}$.*

Similarly, we have the following stronger versions of Theorem 5 and Corollary 6 under Assumption B:

Theorem 9 *Under Assumption B, we have $\lambda_{\min}(K) \geq (1 - 2B(L/2, \delta))\bar{\mathcal{K}}_{11}$.*

Corollary 10 *Under Assumption B, if $L \geq 4L_0(\delta)$, then $\kappa(\bar{\mathcal{K}}) \leq 1 + \frac{2n}{\delta}(1 + \frac{\mu}{2})^{-L/2}$.*

3.4. Extension to uncentered activations

The analysis techniques of the previous sections also extend to activations that need not be normalized (2). Specifically, we only assume that the activation σ satisfies $\mathbb{E}_{X \sim \mathcal{N}(0,1)}[\sigma^2(X)] = 1$. I.e., we allow $\mathbb{E}_{X \sim \mathcal{N}(0,1)}[\sigma(X)]$ to be non-zero. We call σ *non-affine* if it cannot be written as $u \mapsto a + bu$ for constants a, b. For non-affine activations, we can show that dot products of input representations at the top layer converge to a fixed point $\bar{\rho} \in [0, 1]$ as the depth increases. The rate of convergence depends on two quantities. The first is a *coefficient of non-affinity* which generalizes the coefficient of non-linearity:

Definition 11 *The* coefficient of non-affinity *of the activation function σ is defined to be $\tilde{\mu} := 1 - \left(\mathbb{E}_{X \sim \mathcal{N}(0,1)}[\sigma(X)]\right)^2 - \left(\mathbb{E}_{X \sim \mathcal{N}(0,1)}[X\sigma(X)]\right)^2$.*

Clearly, for normalized activations, the two coefficients coincide, i.e. $\mu = \tilde{\mu}$. For non-affine σ, it can be shown that $\tilde{\mu} \in (0, 1]$. The second quantity is the derivative of the dual activation (see Definition 19) at $\bar{\rho}$, denoted by $\dot{\hat{\sigma}}(\bar{\rho})$. We have $\dot{\hat{\sigma}}(\bar{\rho}) \leq 1$ with strict inequality when $\bar{\rho} < 1$. The following theorem generalizes Theorem 3 to non-affine activations (proof in Appendix C.3):

Theorem 12 *Suppose σ is non-affine. Then, there is a $\bar{\rho} \in [0, 1]$ such that $\lim_{L \to \infty} \bar{K}_{ij} = \bar{\rho}$. If $\bar{\rho} < 1$, or $\bar{\rho} = 1$ and $\dot{\hat{\sigma}}(1) < 1$, then supposing Assumption A holds, there is a constant $L_0 = O_\sigma(\log(\frac{1}{\delta}))$ such that if $L \geq L_0$, then*

$$|\bar{K}_{ij} - \bar{\rho}| \leq \max\left\{1 - \frac{\tilde{\mu}(1-\bar{\rho})}{2}, \dot{\hat{\sigma}}(\bar{\rho})\right\}^{L-L_0}\left(\frac{1+\bar{\rho}}{2}\right)$$

for all $i, j \in [n]$ with $i \neq j$.
If $\bar{\rho} = 1$ and $\dot{\hat{\sigma}}(1) = 1$, then there is a constant L_0' depending on σ, such that if $L > L_0'$, then

$$|\bar{K}_{ij} - 1| \leq \frac{\log(2e\tilde{\mu}(L-L_0'))}{2\tilde{\mu}(L-L_0')}$$

for all $i, j \in [n]$ with $i \neq j$.

3.5. Extension to non-unit length inputs

In this section we extend the result of Section 3.1 to the case when the inputs are do not have to be exactly unit length. To establish these results we require further assumptions on the activation function, which we highlight in the theorem. For a discussion of these assumptions see Appendix D.4.

Theorem 13 *Let σ be a twice-differentiable monotonically increasing odd function which is concave on $\mathbb{R}^+$. There exists a constant α_σ (depending on σ) such that for any two inputs x, y such that $\|x\|^2, \|y\|^2 \geq 0.5$, $\frac{|x^\top y|}{\|x\|\|y\|} \leq (1 - \delta)$, when $L \geq \hat{L} := \alpha_\sigma \log(\max(|\|x\|^2 - 1|, |\|y\|^2 - 1|, \mu/4) \cdot 4/\mu)$, we have $\frac{\bar{k}(x,y)}{\sqrt{\bar{k}(x,x) \cdot \bar{k}(y,y)}} \leq B_{\mu/2}(L - \hat{L}, \delta)$.*

The proof of the above theorem as well as a precise description of the constant α_σ can be found in Appendix D in the supplementary material. Our analysis proceeds by first proving the theorem for the norms of the representations induced by the input. The above theorem formalizes a sufficient condition on the activation function for *global* convergence to the fixed point (i.e. 1) of the length map defined in Poole et al. (2016). In fact, for this part we establish a weaker sufficient condition on the activation than Theorem 13, see Appendix D for details. Poole et al. (2016) informally mention that monotonicity of activations suffices, although counterexamples exist, see Appendix D.4. Furthermore, the theorem generalizes the work of Xiao et al. (2019) which only provides the asymptotics close to the fixed point. Next, we show the monotonicity of the normalized dot-product for the representations. This allows us to leverage our previous analysis for the norm 1 case once the norms have converged.

A similar analysis and theorem can be obtained for the NormReLU activation we propose in this paper (details in Section 7). For the precise theorem statement and proof, see Appendix I.1.

4. Implications for optimization

Suppose we train the network using gradient descent on a loss function $\ell : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$, which defines the empirical loss function $\mathcal{L}(\vec{W}) := \frac{1}{n} \sum_{i=1}^{n} \ell(f_{\vec{W}}(x_i), y_i)$. For the rest of this section we will assume that the loss function ℓ is the square loss, i.e. $\ell(\hat{y}, y) = (\hat{y} - y)^2$. The results presented can appropriately be extended to the setting where the loss function is smooth and strongly convex. Training a finite-width neural network necessitates the study of the conditioning of the finite-width kernel matrices K and $\mathcal{K}$, rather than their infinite-width counterparts. In such settings optimization results typically follow from a simple 2-step modular analysis, where in the first step we show via concentration inequalities that conditioning in the infinite-width case transfers to the finite-width case, and in the second step we show that conditioning is not hurt much in the training process. We now provide a couple of representative optimization results that follow from this type of analysis.

4.1. Training only the top layer

We consider a mode of training where only the top layer weight vector, v, is updated, while keeping $W_1, W_2, \ldots, W_L$ frozen at their randomly initialized values. To highlight this we introduce the notation $\vec{W}_{1:L} = \{W_1 \ldots W_L\}$. Let $\eta > 0$ be a step size, the update rule at iteration t is given by $v_{t+1} = v_t - \eta \cdot \partial_v \mathcal{L}(\{v_t, \vec{W}_{1:L}\}) = v_t - \eta \cdot \frac{1}{n} \sum_{i=1}^{n} 2(v_t \cdot \Phi_{\vec{W}}(x_i) - y_i) \Phi_{\vec{W}}(x_i)$. Note that in this mode of training, the optimization problem is convex in v. We assume that σ satisfies a regularity conditioning, *C-boundedness*, introduced by Daniely et al. (2016), which allows us to apply their concentration bounds. Then, standard convex optimization theory (Nesterov, 2014) gives the following result (precise statements and proof are in Appendix E):

Theorem 14 *Suppose $L = \Theta\left(\frac{\log(n/\delta)}{\mu}\right)$, σ is C-bounded and the width $m = poly(n, \frac{1}{\delta})$. Then for an appropriate choice of η, with high probability over the initialization, gradient descent finds an ϵ sub-optimal point in $O(\log(\frac{1}{\epsilon}))$ steps. The same result holds for stochastic gradient descent as well.*

4.2. Training All The Layers Together

In this section we provide a representative result for the training dynamics when all the layers are trained together with a fixed common learning rate. The dynamics are given by $\vec{W}(t+1) = \vec{W}(t) - \eta \partial_{\vec{W}} \mathcal{L}(\vec{W}(t))$. The analysis in this setting follows from carefully establishing that the NTK

does not change too much during the training procedure allowing for the rest of the analysis to go through. We have the following theorem, using the concentration bounds of Lee et al. (2019) (precise theorem statement and proof are in Appendix E):

Theorem 15 *Suppose σ is smooth, bounded and has bounded derivatives. If the width is a large enough constant (depending on L, n, δ) and $L = \Theta(\frac{\log(n/\delta)}{\mu})$, then gradient descent with high probability finds an ϵ suboptimal point in $O(\log(1/\epsilon))$ iterations.*

5. SQ Learnability of Random Deep Neural Nets

In this section we give a generalization of the recent result of Das et al. (2019) regarding learnability of random neural networks. This work studied randomly initialized deep neural networks with sign activations at hidden units. Motivated from the perspective of complexity of learning, they studied learnability of random neural networks in the popular *statistical query learning* (SQ) framework (Kearns, 1998). Their main result establishes that any algorithm for learning a function that is a randomly initialized deep network with sign activations, requires exponential (in depth) many statistical queries in the worst case.

Here we generalize their result in two ways: (a) our result applies to arbitrary activations (as opposed to just sign activations in Das et al. (2019)) satisfying a subgaussianity assumption for standard Gaussian inputs, and (b) our lower bound shows that a randomly initialized network is hard to learn in the SQ model with *constant* probability, as opposed to just positive probability, in Das et al. (2019). We achieve the stronger lower bound by carefully adapting the lower bound technique of Bshouty and Feldman (2002). The subgaussianity assumption is there exists a constant $\alpha > 0$ such that for all λ, we have $\mathbb{E}_{X \sim \mathcal{N}(0,1)}[e^{\lambda \sigma(X)}] \leq e^{\lambda^2 \alpha^2/2}$ for all λ. All standard activations (such as the sign, ReLU and tanh), when normalized, satisfy this assumption.

For technical reasons, we will work with networks that normalize the output of each layer to unit length via the operation $\Pi : \mathbb{R}^m \to \mathbb{R}^m$, and thus the neural network function is of the form

$$f_{\vec{W}}(x) = \left(v \cdot \frac{1}{\sqrt{m}}\Pi(\sigma(W_L \frac{1}{\sqrt{m}}\Pi(\sigma(W_{L-1}\cdots\frac{1}{\sqrt{m}}\Pi(\sigma(W_1 x)\cdots)))))\right). \tag{3}$$

We will consider learning $\mathrm{sgn}(f_{\vec{W}}(x))$ in the SQ model (Kearns, 1998) where the learning algorithm does not have access to a labeled training set. Instead, for a given target function f and a distribution D over $\mathbb{R}^d$, the algorithm has access to a query oracle $SQ_{f,D}(\psi, \tau)$. The oracle takes as input a query function ψ and a tolerance parameter $\tau \geq 0$, and outputs a value v such that $|\mathbb{E}_D[\psi(x, f(x))] - v| \leq \tau$. The goal of the algorithm is to use the query algorithm to output a function g that is ϵ-correlated with f, i.e., $\Pr_D[g(x)f(x)] \geq \epsilon$, for a given $\epsilon > 0$. Our main result is the following (proofs can be found in Appendix F):

Theorem 16 *Fix any nonlinear activation σ with the coefficient of non-linearity μ that satisfies the subgaussianity assumption. Let $f_{\vec{W}}$ be an L-layer neural network with width $m = \Omega(L\frac{\mu^2}{\delta^2})$ taking inputs of dimension d with weights randomly initialized to standard Gaussians. Any algorithm that makes at most $p(d, L)$ statistical queries with tolerance $1/poly(d, L)$ and outputs a function that is $1/poly(d, L)$-correlated with $sgn(f_{\vec{W}})$ must satisfy $p(d, L) \geq e^{\Omega(L)}$.*

A key component in establishing the above SQ hardness of learning is to show that given two non-collinear unit length vectors, a randomly initialized network of depth h and sufficiently large width

makes, in expectation, the pair nearly orthogonal. In other words, the magnitude of the expected dot product between any pair decreases exponentially with depth. While Das et al. (2019) proved the result for sign activations, we prove the statement for more general activations and then use it to establish hardness of learning in the SQ model.

6. Benign Overfitting in Deep Neural Networks

In this section, we give an application of our conditioning results showing how interpolating classifiers (i.e. classifiers achieving perfect training accuracy) can generalize well in the context of deep neural networks. Specifically, we consider the problem of linear regression with square loss where the feature representation is obtained via a randomly initialized deep network, and an interpolating linear predictor is obtained by training only the top layer (i.e. the v vector). Since there are infinitely many interpolating linear predictors in our overparameterized setting, we focus our attention on the *minimum norm* predictor.

Our result builds on the prior work of Bartlett et al. (2019b), which studies benign overfitting in kernel least-squares regression, where the kernel is externally provided. They prove a benign overfitting result (i.e. generalization error going to 0 with increasing sample size) assuming that the spectrum of the kernel matrix decays at a certain slow rate. In this work, we *construct* the kernel via a randomly initialized deep neural network, and show, via our conditioning results, that the spectrum of the kernel matrix decays slowly enough for a benign overfitting result to hold. While the results of Bartlett et al. (2019b) assume a certain *well-specified* setting for the data to prove their generalization bound, we prove our result in the *misspecified* (or *agnostic*) setting and give an *excess risk* bound.

In our setting, the input space is the d dimensional unit sphere $\mathbb{S}^{d-1}$, the output space $\mathcal{Y} = [-1, 1]$, and samples $(x, y) \in \mathbb{S}^{d-1} \times [-1, 1]$ are drawn from an unknown distribution $\mathcal{D}$. The training set is $S = \{(x_i, y_i) \in \mathbb{S}^{d-1} \times [-1, 1]\}_{i=1}^n$. To simplify the presentation, we work in the infinite width setting, i.e. we learn the minimum norm linear predictor in the RKHS $\mathcal{H}$ corresponding to the kernel function $\bar{k}$ (see Section 3.1). Let $\Phi : \mathbb{S}^{d-1} \to \mathcal{H}$ be the feature map corresponding to $\bar{k}$. The loss of a linear predictor parameterized by $v \in \mathcal{H}$ on an example (x, y) is $(y - v^\top \Phi(x))^2$. We denote by v^* the optimal linear predictor, i.e. a vector in $\arg\min_{v \in \mathcal{H}} \mathbb{E}_{(x,y)}[(y - v^\top \Phi(x))^2]$, and by v_S the minimum norm interpolating linear predictor, if one exists. A key quantity of interest is the function $\Delta : \mathbb{N} \times [0, 1] \to [0, 1]$ defined as follows: if $T = \{x'_1, x'_2, \ldots, x'_m\}$ denotes a sample set of size m drawn i.i.d. from the marginal distribution of $\mathcal{D}$ over the x-coordinate, then

$$\Delta(m, \gamma) := \sup \left\{ \delta : \Pr_T \left[\max_{i,j \in [m]: i \neq j} |x'_i \cdot x'_j| \leq 1 - \delta \right] \geq 1 - \gamma \right\}.$$

With this definition, we have the following excess risk bound (proof in Appendix G):

Theorem 17 *For any $\gamma \in (0, 1/2)$, let $L = \left\lceil \frac{\log(n^2)}{-\log(1 - \frac{\mu}{2})} \right\rceil + L_0(\Delta(n^2, \gamma))$. Then, with probability at least $1 - \gamma$ over the choice of S, there exists an interpolating linear predictor, and we have*

$$\mathbb{E}_{(x,y)} [(y - v_S^\top \Phi(x))^2] - \mathbb{E}_{(x,y)} [(y - v^{*\top} \Phi(x))^2] \leq O\left(\frac{\log(n/\gamma)}{n} \|v^*\|^2 \right).$$

A few caveats about the theorem are in order. Note that the number of layers, L, and therefore $\mathcal{H}$ and the optimal linear predictor v^* depends on the sample size n. Thus, the excess risk goes to 0 when n increases if $\|v^*\| = o(\sqrt{n})$.

10

7. Experiments

In this section we present empirical results supporting our theoretical findings, and evaluate the effectiveness of using normalized activations as a replacement for standard operations such as batch normalization and layer normalization in training deep networks.

Normalized ReLU. Motivated by the practical success of ReLU, we propose a family of normalized ReLU-like functions, parameterized by a scalar c, the location of the kink:

$$\text{NormReLU}_c(x) = \lambda(c) \cdot [\max\{x - c, 0\} + b(c)], \tag{4}$$

where the constants $\lambda(c)$ and $b(c)$ are chosen to normalize the function (i.e. (2) holds). In Appendix I, we derive the following closed form expressions for $b(c)$ and $\lambda(c)$: if $\varphi(x)$ and $\Phi(x)$ are the Gaussian density and cumulative distribution functions respectively, then $b(c) = (1 - \Phi(c))c - \varphi(c)$ and $\lambda(c) = [(1 - \Phi(c))\Phi(c)c^2 + (1 - 2\Phi(c))\varphi(c)c + (1 - \Phi(c) - \varphi(c)^2)]^{-1/2}$.

In our experimental setup we choose $c = -1.5975$ since this gives $\lambda(c) \approx 1.05$, which is the same scaling factor in the SeLU activation Klambauer et al. (2017). For this value of c, we have $b(c) \approx -1.6209$. In the following, we refer to $\text{NormReLU}_{-1.5975}$ simply as NormReLU for convenience. Figure 2 shows this NormReLU activation compared to SeLU. Appendix I has a comparison of the two activations in terms of training and generalization behavior. Appendix I also contains additional experimental details for this section including the choice of hyperparameters and the number of training and evaluation runs.
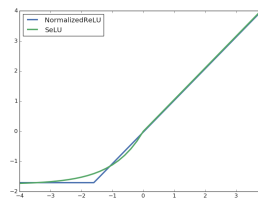

Figure 2: NormReLU vs SeLU.

Effectiveness of NormReLU for training deep neural networks. We first train fully connected feedforward networks of depth 32 and 64 on the CIFAR-10 dataset using either batch normalization, layer normalization or NormReLU. For each method, the best learning rate is chosen via cross validation. Figure 3 below shows how the training and the test accuracy increases with the number of epochs. As predicted by our theory, using NormReLU results in significantly faster optimization. Furthermore, the model trained via NormReLU also generalizes significantly better than using either batch or layer normalization. However, when using NormReLU we observed that we had to use small learning rates, of the order of 10^{-5} to stabilize training. Batch or Layer normalizations on the other hand are less sensitive and can be applied in conjunction with large learning rates.
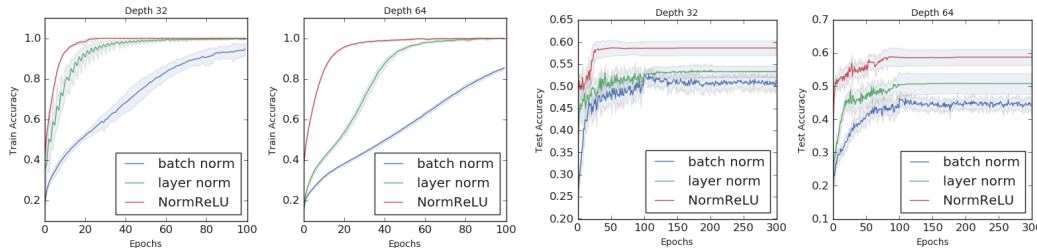

Figure 3: Train and test accuracy vs. the number of epochs. The left plots shows the optimization performance of the NormReLU as compared to standard methods such as batch normalization and layer normalization. The plots are obtained by training depth 32 fully connected networks with 3000 hidden units in each layer. Similarly, the right plot compares the test accuracy of the three methods.

NormReLU as a replacement of batch normalization for other architectures. We train deep convolutional networks with the ResNet architecture He et al. (2016a) using the standard practice of using batch normalization with skip connections and also by replacing batch normalization with NormReLU. We do not use layer normalization since that is not the standard way to train CNNs. Figure 4 shows the train and test accuracies obtained on both network architectures. As can be seen, the use of NormReLU is indeed competitive with batch normalization achieving similar test accuracies and slightly outperforming batch normalization at depth 110.
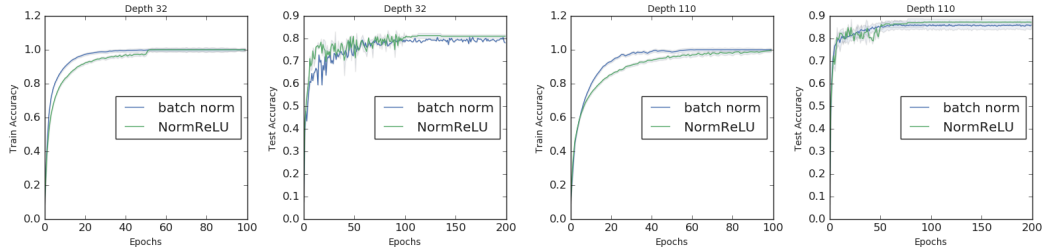

Figure 4: Train and test accuracy vs. the number of epochs. The plots are obtained by training a depth 32 and a depth 110 ResNet architecture.

Comparison with Fixup initialization. We show that on the CIFAR-10 dataset, using NormReLU with standard initialization we can achieve comparable results to those obtained using standard ReLU with the Fixup initialization method of Zhang et al. (2019). We turn on data augmentation and train the same 110 depth architecture as in Zhang et al. (2019). We replace Fixup initialization with standard Gaussian initialization where the kernel weights are initialized with a mean zero Gaussian and a variance of $1/(\text{kernel\_size}^2 \cdot \text{num\_out\_channels})$, and replace the standard ReLU activation with NormReLU. Figure 5 shows the test accuracies achieved by both the methods. Note that training with NormReLU achieves a similar accuracy as with Fixup initialization.
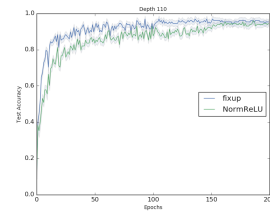

Figure 5: NormReLU vs Fixup on CIFAR-10 with a ResNet 110 architecture.

8. Conclusions and Future Directions

In this work we further elaborated the role of depth in training of modern neural networks by showing that the conditioning of the input data improves exponentially with depth at random initialization. It would be interesting to further rigorously understand how the conditioning behaves during the course of training. An excellent open question is to analyze more realistic parameter regimes (low width in particular). While it is reasonably straightforward to extend our analysis to architectures such as Convolutional Neural Networks and ResNets, extending it to architectures such as Recurrent Networks and Transformers would be quite interesting. Finally, we believe that our main result is quite general and it would be interesting to explore other applications in addition to the ones presented in the current work.

References

Zeyuan Allen-Zhu and Yuanzhi Li. Backward feature correction: How deep learning performs deep learning. *arXiv preprint arXiv:2001.04413*, 2020.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. *arXiv preprint arXiv:1811.03962*, 2018.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *ICML*, pages 242–252, 2019.

Alexandr Andoni, Rina Panigrahy, Gregory Valiant, and Li Zhang. Learning polynomials with neural networks. In *International conference on machine learning*, pages 1908–1916, 2014.

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. *arXiv preprint arXiv:1611.01491*, 2016.

Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. *arXiv preprint arXiv:1810.02281*, 2018a.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *ICML*, pages 244–253, 2018b.

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. *arXiv preprint arXiv:1802.05296*, 2018c.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *CoRR*, abs/1905.13655, 2019a.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems*, pages 7411–7422, 2019b.

Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *NeurIPS*, 2019c.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

Ainesh Bakshi, Rajesh Jayaram, and David P Woodruff. Learning two layer rectified neural networks in polynomial time. *arXiv preprint arXiv:1811.01885*, 2018.

David Balduzzi, Marcus Frean, Lennox Leary, J. P. Lewis, Kurt Wan-Duo Ma, and Brian McWilliams. The shattered gradients problem: If resnets are the answer, then what is the question? In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 342–350. PMLR, 2017.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017.

Peter L Bartlett, David P Helmbold, and Philip M Long. Gradient descent with identity initialization efficiently learns positive-definite linear transformations by deep residual networks. *Neural computation*, 31(3):477–502, 2019a.

Peter L. Bartlett, Philip M. Long, Gabor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *arXiv preprint arXiv:1906.11300*, 2019b.

Mikhail Belkin, Daniel J. Hsu, and Partha Mitra. Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate. In *NeurIPS*, pages 2306–2317, 2018.

Mikhail Belkin, Daniel Hsu, and Ji Xu. Two models of double descent for weak features. *CoRR*, abs/1903.07571, 2019a.

Mikhail Belkin, Alexander Rakhlin, and Alexandre B. Tsybakov. Does data interpolation contradict statistical optimality? In *AISTATS*, pages 1611–1619, 2019b.

Nader H Bshouty and Vitaly Feldman. On using extended statistical queries to avoid membership queries. *Journal of Machine Learning Research*, 2(Feb):359–395, 2002.

Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in neural information processing systems*, pages 3036–3046, 2018.

Amit Daniely. Depth separation for neural networks. In *COLT*, pages 690–696, 2017a.

Amit Daniely. Sgd learns the conjugate kernel class of the network. In *Advances in Neural Information Processing Systems*, pages 2422–2430, 2017b.

Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *Advances In Neural Information Processing Systems*, pages 2253–2261, 2016.

Abhimanyu Das, Sreenivas Gollapudi, Ravi Kumar, and Rina Panigrahy. On the learnability of deep random networks. *CoRR*, abs/1904.03866, 2019. URL http://arxiv.org/abs/1904.03866.

Olivier Delalleau and Yoshua Bengio. Shallow vs. deep sum-product networks. In *Advances in neural information processing systems*, pages 666–674, 2011.

Simon S Du, Jason D Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. *arXiv preprint arXiv:1811.03804*, 2018.

Simon S. Du, Xiyu Zhai, Barnabás Póczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *ICLR*, 2019.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In *COLT*, pages 907–940, 2016.

Rong Ge, Jason D Lee, and Tengyu Ma. Learning one-hidden-layer neural networks with landscape design. *arXiv preprint arXiv:1711.00501*, 2017.

Rong Ge, Rohith Kuditipudi, Zhize Li, and Xiang Wang. Learning two-layer neural networks with symmetric inputs. *arXiv preprint arXiv:1810.06793*, 2018.

T. Gneiting. Strictly and non-strictly positive definite functions on spheres. *Bernoulli*, 19(4): 1327–1349, 2013.

Surbhi Goel and Adam Klivans. Learning neural networks with two nonlinear layers in polynomial time. *arXiv preprint arXiv:1709.06010*, 2017.

Surbhi Goel, Adam Klivans, and Raghu Meka. Learning one convolutional layer with overlapping patches. *arXiv preprint arXiv:1802.02547*, 2018.

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *CoRR*, abs/1903.08560, 2019. URL http://arxiv.org/abs/1903.08560.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016b.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, pages 8580–8589, 2018.

Daniel M Kane and Ryan Williams. Super-linear gate and super-quadratic wire lower bounds for depth-two and depth-three threshold circuits. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 633–643, 2016.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *Advances in neural information processing systems*, pages 971–980, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Holden Lee, Rong Ge, Tengyu Ma, Andrej Risteski, and Sanjeev Arora. On the ability of neural nets to express distributions. In *COLT*, pages 1271–1296, 2017.

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 *NeurIPS*, pages 8570–8581. 2019.

Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *NeurIPS*, pages 8168–8177, 2018.

Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with relu activation. In *Advances in neural information processing systems*, pages 597–607, 2017.

Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel "ridgeless" regression can generalize. *CoRR*, abs/1808.00387, 2018. URL http://arxiv.org/abs/1808.00387.

Tengyuan Liang, Alexander Rakhlin, and Xiyu Zhai. On the risk of minimum-norm interpolants and restricted lower isometry of kernels. *CoRR*, abs/1908.10292, 2019. URL http://arxiv.org/abs/1908.10292.

Philip M Long and Hanie Sedghi. Size-free generalization bounds for convolutional neural networks. *arXiv preprint arXiv:1905.12600*, 2019.

Eran Malach and Shai Shalev-Shwartz. Is deeper better only when shallow is good? In *Advances in Neural Information Processing Systems*, pages 6426–6435, 2019.

James Martens and Venkatesh Medabalimi. On the expressive efficiency of sum product networks. *arXiv preprint arXiv:1411.7717*, 2014.

Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and double descent curve. *CoRR*, abs/1908.05355, 2019.

Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

Vaishnavh Nagarajan and J Zico Kolter. Deterministic pac-bayesian generalization bounds for deep networks via generalizing noise-resilience. *arXiv preprint arXiv:1905.13344*, 2019.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Publishing Company, Incorporated, 1 edition, 2014. ISBN 1461346916.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401, 2015.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

Samet Oymak and Mahdi Soltanolkotabi. Towards moderate overparameterization: global convergence guarantees for training shallow neural networks. *CoRR*, abs/1902.04674, 2019.

Abhishek Panigrahi, Abhishek Shetty, and Navin Goyal. Effect of activation functions on the training of overparametrized neural nets. In *ICLR*, 2020.

Jeffrey Pennington, Samuel S. Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. In *NeurIPS*, pages 4785–4795, 2017.

Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *NeurIPS*, pages 3360–3368, 2016.

Maithra Raghu, Ben Poole, Jon M. Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *ICML*, pages 2847–2854, 2017.

Grant Rotskoff and Eric Vanden-Eijnden. Parameters as interacting particles: long time convergence and asymptotic error scaling of neural networks. In *Advances in neural information processing systems*, pages 7146–7155, 2018.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *NeurIPS*, pages 2488–2498, 2018.

I. J. Schoenberg. Positive definite functions on spheres. *Duke Math. J.*, 9(1):96–108, 03 1942. doi: 10.1215/S0012-7094-42-00908-6. URL https://doi.org/10.1215/S0012-7094-42-00908-6.

Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. In *ICLR*, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks. *arXiv preprint arXiv:1805.01053*, 2018.

Mahdi Soltanolkotabi, Adel Javanmard, and Jason D Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 65(2):742–769, 2018.

Le Song, Santosh Vempala, John Wilmes, and Bo Xie. On the complexity of learning neural networks. In *Advances in neural information processing systems*, pages 5514–5522, 2017.

Lili Su and Pengkun Yang. On learning over-parameterized neural networks: A functional approximation perspective. In *NeurIPS*, pages 2637–2646, 2019.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

Matus Telgarsky. Benefits of depth in neural networks. In *COLT*, pages 1517–1539, 2016.

Joel A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends in Machine Learning*, 8(1-2):1–230, 2015.

Santosh Vempala and John Wilmes. Gradient descent for one-hidden-layer neural networks: Polynomial convergence and sq lower bounds. *arXiv preprint arXiv:1805.02677*, 2018.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Colin Wei, Jason D Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets vs their induced kernel. In *Advances in Neural Information Processing Systems*, pages 9709–9721, 2019.

Lechao Xiao, Jeffrey Pennington, and Samuel S. Schoenholz. Disentangling trainability and generalization in deep learning. *CoRR*, abs/1912.13053, 2019.

Greg Yang. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *CoRR*, abs/1902.04760, 2019.

Greg Yang, Jeffrey Pennington, Vinay Rao, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. A mean field theory of batch normalization. In *ICLR*, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

Hongyi Zhang, Yann N Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. *arXiv preprint arXiv:1901.09321*, 2019.

Difan Zou and Quanquan Gu. An improved analysis of training over-parameterized deep neural networks. In *NeurIPS*, 2019.

Appendix A. Related Work

Representational Benefits of Depth.   Analogous to depth hierarchy theorems in circuit complexity, many recent works have aimed to characterize the representational power of deep neural networks when compared to their shallow counterparts. The work of Delalleau and Bengio (2011) studies sum-product networks and constructs examples of functions that can be efficiently represented by depth 4 or higher networks and require exponentially many neurons for representation with depth one networks. The works of Martens and Medabalimi (2014) and Kane and Williams (2016) study networks of linear threshold gates and provide similar separation results. Eldan and Shamir (2016) show that for many popular activations such as sigmoid, ReLU etc. there are simple functions that can be computed by depth 3 feed forward networks but require exponentially (in the input dimensionality) many neurons to represent using two layer feed forward networks. Telgarsky (2016) generalizes this to construct, for any integer k, a family of functions that can be approximated by $\Theta(k^3)$ layers and $\Theta(k^3)$ size and require exponential in k neurons to represent with $O(k)$ depth.

Optimization Benefits of Depth.   While the benefits of depth are well understood in terms of the representation power using a small number of neurons, the question of whether increasing depth helps with optimization is currently poorly understood. The recent work of Arora et al. (2018b) aims to understand this question for the special case of linear neural networks. For the case of ℓ_p regression, they show that gradient descent updates on a depth 2 linear network correspond to accelerated gradient descent type updates on the original weight vector. Similarly, they derive the form of the weight updates for a general over parameterized deep linear neural network and show that these updates can be viewed as performing gradient descent on the original network but with a preconditioning operation applied to the gradient at each step. Empirically this leads to faster convergence. The works of Bartlett et al. (2019a) and Arora et al. (2018a) study the convergence of gradient descent on linear regression problems when solved via an over parameterized deep linear network. These works establish that under suitable assumptions on the initialization, gradient descent on the over parameterized deep linear networks enjoys the same rate of convergence as performing linear regression in the original parameter space which is a smooth and strongly convex problem.

In a similar vein, the recent work of Arora et al. (2019b) analyzes over parameterized deep linear networks for solving matrix factorization, and shows that the solution to the gradient flow equations approaches the minimum nuclear norm solution at a rate that increases with the depth of the network. The recent work of Malach and Shalev-Shwartz (2019) studies depth separation between shallow and deeper networks over distributions that have a certain fractal structure. In certain regimes of the parameters of the distribution the authors show that, surprisingly, the stronger the depth separation is, the harder it becomes to learn the distribution via a deep network using gradient based algorithms.

Optimization of Neural Networks via Gradient Descent   In recent years there has been a large body of work in analyzing the convergence of gradient descent and stochastic gradient descent (SGD) on over parameterized neural networks. The work of Andoni et al. (2014) shows that depth one neural networks with quadratic activations can efficiently represent low degree polynomials and performing gradient descent on the network starting with random initialization can efficiently learn such classes. The work of Li and Yuan (2017) shows convergence of gradient descent on the population loss and under Gaussian input distribution, of a two layer feed forward network with relu activations and the identity mapping mimicking the ResNet architecture. Under similar assumptions the work of

Soltanolkotabi et al. (2018) analyzes SGD for two layer neural networks with quadratic activations. The work of Li and Liang (2018) extends these results to more realistic data distributions.

Building upon the work of Daniely et al. (2016), Daniely (2017b) shows that SGD when run on over parameterized neural networks achieves at most ϵ excess loss (on the training set) over the best predictor in the conjugate kernel class at the rate that depends on $1/\epsilon^2$ and M, the norm of the best predictor. This result is extended in the work of Du et al. (2019) showing that by running SGD on a randomly initialized two layer over parameterized networks with relu activations, one can get ϵ loss on the training data at the rate that depends on $\log(1/\epsilon)$ and the smallest eigenvalue of a certain kernel matrix. While the authors show that this eigenvalue is positive, no explicit bound is provided. These results are extended to higher depth in (Du et al., 2018) at the expense of an exponential dependence on the depth on the amount of over parameterization needed. In (Allen-Zhu et al., 2018) the authors provide an alternate analysis under the weaker Assumption A and at the same time obtain convergence rates that depend on $\log(1/\epsilon)$ and only polynomially in the depth of the network. A few recent papers (Oymak and Soltanolkotabi, 2019; Zou and Gu, 2019; Su and Yang, 2019) provide an improved analysis with better dependence on the parameters. We would like to point out that all the above works fail to explain the optimization benefits of depth, and in fact the resulting bounds degrade as the network gets deeper.

The work of Jacot et al. (2018) proposed the Neural Tangent Kernel (NTK) that is associated with a randomly initialized neural network in the infinite width regime. The authors show that in this regime performing gradient descent on the parameters of the network is equivalent to kernel regression using the NTK. The work of Lee et al. (2019) and Yang (2019) generalizes this result and the recent work of Arora et al. (2019c) provides a non-asymptotic analysis and an algorithm for exact computation of the NTK for feed forward and convolutional neural networks. There have also been works analyzing the mean field dynamics of SGD on infinite width neural networks (Mei et al., 2018; Chizat and Bach, 2018; Rotskoff and Vanden-Eijnden, 2018; Sirignano and Spiliopoulos, 2018) as well as works designing provable learning algorithms for shallow neural networks under certain assumptions (Arora et al., 2016; Ge et al., 2017; Goel and Klivans, 2017; Ge et al., 2018; Goel et al., 2018; Bakshi et al., 2018; Vempala and Wilmes, 2018). Recent works have also explored the question of providing sample complexity based separation between training via the NTK vs. training all the layers (Wei et al., 2019; Allen-Zhu and Li, 2020).

SQ Learnability of Neural Networks.   Several recent works have studied the statistical query (SQ) framework of Kearns (1998) to provide lower bounds on the number of queries needed to learn neural networks with a certain structure (Song et al., 2017; Vempala and Wilmes, 2018; Das et al., 2019). The closest to us is the recent work of Das et al. (2019) that shows that learning a function that is a randomly initialized deep neural network with sign activations requires exponential in depth many statistical queries. A crucial part of their analysis requires showing that for randomly initialized neural networks with sign activations, the pairwise (normalized) dot products decrease exponentially fast with depth. Our main result in Theorem 3 strictly generalizes this result for arbitrary non-linear activations (under mild assumptions) thereby implying exponential SQ lower bounds for networks with arbitrary non linear activations. In particular, we show any algorithm that works in the statistical query framework, and learns (with high probability) a sufficiently deep randomly initialized network with an arbitrary non-linear activation, must necessarily use exponentially (in depth) many queries in the worst case. The only requirement we impose on the non-linear activations is that they satisfy

subgaussianity (see Section F), a condition satisfied by popular activations such as relu, sign, and tanh.

Generalization in Neural Networks. It has been observed repeatedly that modern deep neural networks have sufficient capacity to perfectly memorize the training data, yet generalize to test data very well (see, e.g., (Zhang et al., 2017)). This observation flies in the face of conventional statistical learning theory which indicates that such overfitting should lead to poor generalization. Since then there has been a line of work providing generalization bounds for neural networks that depend on compressibility of the network (Arora et al., 2018c), norm based bounds (Neyshabur et al., 2015; Bartlett et al., 2017), bounds via PAC-bayes analysis (Neyshabur et al., 2017; Dziugaite and Roy, 2017; Nagarajan and Kolter, 2019) and bounds that depend on the distance to initialization (Long and Sedghi, 2019). Since randomly initialized neural networks are interpolating classifiers, i.e., they achieve zero error on the training set, there have also been recent works (e.g. (Belkin et al., 2018, 2019b; Liang and Rakhlin, 2018; Liang et al., 2019; Bartlett et al., 2019b; Belkin et al., 2019a; Hastie et al., 2019; Mei and Montanari, 2019)) that study the generalization phenomenon in the context of specific *interpolating* methods (i.e. methods which perfectly fit the training data) and show how the obtained predictors can generalize well.

Appendix B. Proof of Lemma 1

In this section, we prove Lemma 1, restated here for convenience:

Lemma 18 *If the neural network in* (1) *incorporates batch normalization in each layer, then the network output is the same regardless of whether the activation σ is normalized or not. The same holds if instead layer normalization is employed, but on the post-activation outputs rather than pre-activation inputs.*

Proof Let σ be an arbitrary activation function, and $\tilde{\sigma}$ be its normalized version. Thus $\tilde{\sigma}(u) = \frac{\sigma(u) - c}{s}$ for some constants c and s.

Invariance with Batch Normalization. For a batch size b, the vanilla Batch Normalization operation $\mathrm{BN} : \mathbb{R}^b \to \mathbb{R}^b$ is defined as follows:

$$\mathrm{BN}(v) = \frac{1}{\nu}(v - \mu \vec{1}),$$

where $\vec{1}$ is the d-dimensional all ones vector, $\mu = \frac{1}{b} v \cdot \vec{1}$, and $\nu = \sqrt{\frac{1}{b} \|v - \mu \vec{1}\|_2^2}$.

Now, fix a particular layer in the neural network, and a particular hidden unit in that layer. Let w be the weight vector corresponding to that hidden unit. Thus if x is the pre-activation input to the previous layer, then $w \cdot \frac{1}{\sqrt{m}} \sigma(x)$ is the pre-activation value[4] for the hidden unit in question. Suppose the batch size is b, and let $x_1, x_2, \ldots, x_b$ be the pre-activation inputs to the previous layer. Define $X = [x_1, x_2, \ldots, x_b]$, and $\sigma(X) = [\sigma(x_1), \sigma(x_2), \ldots, \sigma(x_b)]$. Then $v = w^\top \frac{1}{\sqrt{m}} \sigma(X)$ is the vector of pre-activation inputs to the hidden unit for the batch. Now, we claim that

$$\mathrm{BN}(w^\top \tfrac{1}{\sqrt{m}} \sigma(X)) = \mathrm{BN}(w^\top \tfrac{1}{\sqrt{m}} \tilde{\sigma}(X)), \tag{5}$$

---

4. We can handle the input layer of the network by simply setting σ to be the identity multiplied by $\sqrt{m}$.

which establishes the desired invariance. Let $\tilde{v} = w^\top \frac{1}{\sqrt{m}}\tilde{\sigma}(X)$. Define $\mu = \frac{1}{b}v \cdot \vec{1}$, $\nu = \sqrt{\frac{1}{b}\|v - \mu\vec{1}\|_2^2}$, $\tilde{\mu} = \frac{1}{b}\tilde{v} \cdot \vec{1}$ and $\tilde{\nu} = \sqrt{\frac{1}{b}\|\tilde{v} - \tilde{\mu}\vec{1}\|_2^2}$. Then by direct calculation we have $\tilde{\mu} = \frac{\mu - c}{s}$, and $\tilde{\nu} = \frac{\nu}{s}$. These facts imply the claimed equality (5).

Finally, the standard Batch Normalization operation also includes constants γ and β so that the final output is $\gamma \mathrm{BN}(\cdot) + \beta$. The above analysis immediately implies that

$$\gamma \mathrm{BN}(w^\top \tfrac{1}{\sqrt{m}}\sigma(X)) + \beta = \gamma \mathrm{BN}(w^\top \tfrac{1}{\sqrt{m}}\tilde{\sigma}(X)) + \beta,$$

establishing invariance for the standard Batch Normalization operation as well.

Invariance with post-activation Layer Normalization. For a layer with m hidden units, the Layer Normalization operation $\mathrm{LN} : \mathbb{R}^m \to \mathbb{R}^m$ is defined as follows:

$$\mathrm{LN}(v) = \frac{1}{\nu}(v - \mu\vec{1}),$$

where $\vec{1}$ is the m-dimensional all ones vector, $\mu = \frac{1}{m}v \cdot \vec{1}$, and $\nu = \sqrt{\frac{1}{m}\|v - \mu\vec{1}\|_2^2}$.

Now suppose we apply Layer Normalization to the post-activation outputs (i.e. on $v = \frac{1}{\sqrt{m}}\tilde{\sigma}(x)$, for a pre-activation input $x \in \mathbb{R}^m$) the rather than pre-activation inputs. Then, we claim that

$$\mathrm{LN}(\tfrac{1}{\sqrt{m}}\sigma(x)) = \mathrm{LN}(\tfrac{1}{\sqrt{m}}\tilde{\sigma}(x)), \tag{6}$$

which establishes the desired invariance. Let $\tilde{v} = \frac{1}{\sqrt{m}}\tilde{\sigma}(x)$. Define $\mu = \frac{1}{m}v \cdot \vec{1}$, $\nu = \sqrt{\frac{1}{m}\|v - \mu\vec{1}\|_2^2}$, $\tilde{\mu} = \frac{1}{m}\tilde{v} \cdot \vec{1}$ and $\tilde{\nu} = \sqrt{\frac{1}{m}\|\tilde{v} - \tilde{\mu}\vec{1}\|_2^2}$. Then by direct calculation we have $\tilde{\mu} = \frac{\mu - c}{s}$, and $\tilde{\nu} = \frac{\nu}{s}$. These facts imply the claimed equality (6). ∎

Appendix C. Conditioning Analysis

Recall the notion of the dual activation $\hat{\sigma}$ for the activation σ:

Definition 19 *For $\rho \in [-1, 1]$, define matrix $\Sigma_\rho = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$. Define the conjugate activation function $\hat{\sigma} : [-1, 1] \to [-1, 1]$ as follows:*

$$\hat{\sigma}(\rho) := \mathop{\mathbb{E}}_{(X,X')\sim\mathcal{N}(0,\Sigma_\rho)}[\sigma(X)\sigma(X')].$$

The following facts can be found in Daniely et al. (2016):

1. Let $x, x' \in \mathbb{R}^d$ such that $\|x\| = \|x'\| = 1$. Then

$$\mathop{\mathbb{E}}_{w\sim\mathcal{N}(0,I_d)}[\sigma(w \cdot x)\sigma(w \cdot x')] = \hat{\sigma}(x \cdot x').$$

2. Since $\mathbb{E}_{X\sim\mathcal{N}(0,1)}[\sigma^2(X)] = 1$, σ is square integrable w.r.t. the Gaussian measure. The (probabilitist's) Hermite polynomials $h_0, h_1, \ldots$ form an orthogonal basis for the Hilbert space of square integrable functions w.r.t. the Gaussian measure, and hence σ can be written as $\sigma(u) = \sum_{i=0}^{\infty} a_i h_i(u)$, where $a_i = \mathbb{E}_{X\sim\mathcal{N}(0,1)}[\sigma(X)h_i(X)]$. This expansion is known as the Hermite expansion for σ.

3. We have $\hat{\sigma}(\rho) = \sum_{i=0}^{\infty} a_i^2 \rho^i$.

4. The normalization (2) has the following consequences. Since $\mathbb{E}_{X \sim \mathcal{N}(0,1)}[\sigma(X)] = 0$, we have $a_0 = 0$, and since $\mathbb{E}_{X \sim \mathcal{N}(0,1)}[\sigma^2(X)] = 1$ we have $\sum_{i=1}^{\infty} a_i^2 = 1$.

5. If $\dot{\sigma}$ denotes the derivative of σ, then $\hat{\dot{\sigma}} = \dot{\hat{\sigma}}$.

The above facts imply the following simple bound on the coefficient of non-linearity μ:

Lemma 20 *For any normalized non-linear activation function σ, we have $0 < \mu \le 1$.*

Proof The degree 1 Hermite polynomial is $h_1(u) = u$, so $a_1 = \mathbb{E}_{X \sim \mathcal{N}(0,1)}[\sigma(X)X]$. Since σ is non-linear, for at least one $i \ne 1$, we have $a_i \ne 0$. This, coupled with the fact that $\sum_{i=1}^{\infty} a_i^2 = 1$ implies that $a_1 \in (-1, 1)$, which implies that $\mu = 1 - a_1^2 \in (0, 1]$. ∎

The random initialization of the neural network induces a feature representation of the input vectors at every depth l in the neural network: $\Phi_{\tilde{W}}^{(l)}(x) := \frac{1}{\sqrt{m}}\sigma(W_l \frac{1}{\sqrt{m}}\sigma(W_{l-1} \cdots \frac{1}{\sqrt{m}}\sigma(W_1 x)\cdots))$. This feature representation naturally yields a kernel function $k^{(l)}(x, x') := \Phi_{\tilde{W}}^{(l)}(x) \cdot \Phi_{\tilde{W}}^{(l)}(x')$. In particular, after the first layer, the kernel function $k^{(1)}(x, x') = \frac{1}{m}\sigma(W_1 x) \cdot \sigma(W_1 x')$. The central limit theorem implies that as the width m goes to infinity, this kernel function tends to a deterministic value, viz. its expectation, which is $\mathbb{E}_{w \sim \mathcal{N}(0, I_d)}[\sigma(w \cdot x)\sigma(w \cdot x')]$, which equals $\hat{\sigma}(x \cdot x')$ if x and x' are unit vectors. Furthermore, the normalization $\mathbb{E}_{X \sim \mathcal{N}(0,1)}[\sigma^2(X)] = 1$ implies that the feature representation is itself normalized in the sense for any unit vector x, that as $m \to \infty$, we have $\|\Phi_{\tilde{W}}^{(1)}(x)\|_2^2 = k^{(1)}(x, x) \to 1$. Applying these observations recursively, we get Lemma 21, which was also proved by Daniely et al. (2016).

Lemma 21 *Suppose $\|x\|_2 = \|x'\|_2 = 1$. Then for any depth l, as $m \to \infty$*

$$k^{(l)}(x, x') \xrightarrow{a.s.} \hat{\sigma}^{(l)}(x \cdot x'),$$

where $\hat{\sigma}^{(l)}$ denotes the l-fold composition of $\hat{\sigma}$ with itself.

The following technical lemma shows how one application of $\hat{\sigma}$ behaves:

Lemma 22 *Let $\delta \in [0, 1]$. Then*

$$|\hat{\sigma}(-(1-\delta))| \le \hat{\sigma}(1-\delta) \le \begin{cases} 1 - (1 + \frac{\mu}{2})\delta & \text{if } \delta \le \frac{1}{2} \\ (1 - \frac{\mu}{2})(1 - \delta) & \text{if } \delta > \frac{1}{2}. \end{cases}$$

Proof The fact that $|\hat{\sigma}(-(1-\delta))| \le \hat{\sigma}(1-\delta)$ follows from the fact that the power series $\hat{\sigma}(\rho) = \sum_{i=1}^{\infty} a_i^2 \rho^i$ has only non-negative coefficients. Next, we have

$$\hat{\sigma}(1-\delta) = \sum_{i=1}^{\infty} a_i^2 (1-\delta)^i \le a_1^2(1-\delta) + \sum_{i=2}^{\infty} a_i^2 (1-\delta)^2 = a_1^2(1-\delta) + (1 - a_1^2)(1-\delta)^2 = (1-\delta)(1-\mu\delta).$$

Now if $\delta > \frac{1}{2}$, we have $(1-\delta)(1-\mu\delta) \le (1 - \frac{\mu}{2})(1-\delta)$. If $\delta \le \frac{1}{2}$, we have $(1-\delta)(1-\mu\delta) = 1 - (1+\mu)\delta + \mu\delta^2 \le 1 - (1+\frac{\mu}{2})\delta$. ∎

23

Recall the definition of $B_\nu(L, \delta)$: for any $\nu \in (0, 1]$, $\delta \in (0, 1)$ and a positive integer L, let $L_0(\delta) = \max\left\{\left\lceil \frac{\log(\frac{1}{2\delta})}{\log(1 + \frac{\nu}{2})} \right\rceil, 0\right\} = O\left(\frac{\log(1/\delta)}{\nu}\right)$, and define

$$B_\nu(L, \delta) := \begin{cases} 1 - \delta(1 + \frac{\nu}{2})^L & \text{if } L \le L_0(\delta) \\ \frac{1}{2}(1 - \frac{\nu}{2})^{L - L_0(\delta)} & \text{if } L > L_0(\delta). \end{cases}$$

Remember, we denote by $B(L, \delta)$, the quantity $B_\mu(L, \delta)$.

The following lemma is an immediate consequnce via repeated application of Lemma 22:

Lemma 23 (Correlation decay lemma) *Suppose $|\rho| \le 1 - \delta$ for some $\delta \in (0, 1]$. Then*

$$|\hat{\sigma}^{(L)}(\rho)| \le B(L, \delta).$$

The final technical ingredient we need is the following linear-algebraic lemma which gives a lower bound on the smallest eigenvalue of a matrix obtained by the application of a given function to all entries of another positive definite matrix:

Lemma 24 (Eigenvalue lower bound lemma) *Let $f : [-1, 1] \to \mathbb{R}$ be an arbitrary function whose power series $f(\rho) = \sum_{i=0}^\infty a_i \rho^i$ converges everywhere in $[-1, 1]$ and has non-negative coefficients $a_i \ge 0$. Let $K \in \mathbb{R}^{n \times n}$ be a positive definite matrix with $K \succeq \delta I_n$ for some $\delta > 0$, and all diagonal entries equal to 1. Let $f[K]$ be matrix obtained by entrywise application of f. Then we have*

$$f[K] \succeq (f(1) - f(1 - \delta))I_n.$$

Proof We have $f[K] = \sum_{i=1}^\infty a_i K^{\odot i}$, where $K^{\odot i}$ denotes the i-fold Hadamard (i.e. entrywise) product of K with itself. Since all diagonal entries of K equal 1, we can also write $K^{\odot i}$ as

$$K^{\odot i} = (K - \delta I_n)^{\odot i} + (1 - (1 - \delta)^i)I_n.$$

By assumption, $K - \delta I_n \succeq 0$. Since the Hadamard product of positive semidefinite matrices is also positive semidefinite, we have $(K - \delta I_n)^{\odot i} \succeq 0$. Thus, $K^{\odot i} \succeq (1 - (1 - \delta)^i)I_n$. Thus, we have

$$f[K] = \sum_{i=0}^\infty a_i K^{\odot i} \succeq \sum_{i=0}^\infty a_i(1 - (1 - \delta)^i)I_n = (f(1) - f(1 - \delta))I_n,$$

as required. ∎

C.1. Top Layer Kernel Matrix

We can now prove Theorem 3 and Theorem 7: which we restate here in a combined form for convenience:

Theorem 25 *The following bounds hold:*

1. *Under Assumption A, we have $|\bar{K}_{ij}| \le B(L, \delta)$ for all $i, j \in [n]$ with $i \ne j$.*

2. *Under Assumption B, we have $\lambda_{\min}(\bar{K}) \ge 1 - B(L, \delta)$.*

Proof Part 1 follows directly from Lemma 23.

As for part 2, Assumption B implies that $\underline{K} \geq \delta I_n$. Since the function $\hat{\sigma}^{(L)} : [-1, 1] \to \mathbb{R}$ defines a kernel on the unit sphere, by Schoenberg's theorem (Schoenberg, 1942), its power series expansion has only non-negative coefficients, so Lemma 24 applies to $\bar{K} = \hat{\sigma}^{(L)}[\underline{K}]$, and we have

$$\bar{K} \geq (\hat{\sigma}^{(L)}(1) - \hat{\sigma}^{(L)}(1 - \delta))I_n \geq (1 - B(L, \delta))I_n,$$

using Lemma 23 and the fact that $\hat{\sigma}^{(L)}(1) = 1$. ∎

We can now prove Corollary 4 and Corollary 8, restated here in a combined form for convenience:

Corollary 26 *The following bounds on $\kappa(\bar{K})$ hold:*

1. *Under Assumption A, if $L \geq L_1(\delta) := \left\lceil \frac{\log(n)}{-\log(1 - \frac{\mu}{2})} \right\rceil + L_0(\delta)$, then $\kappa(\bar{K}) \leq 1 + 2n(1 - \frac{\mu}{2})^{L - L_1(\delta)}$.*

2. *Under Assumption B, we have $\kappa(\bar{K}) \leq 1 + \frac{n}{\delta}(1 + \frac{\mu}{2})^{-L}$.*

Proof To prove part 1, note that the normalization (2) implies that $\bar{K}_{ii} = 1$ for all $i \in [n]$. This fact, coupled with Theorem 3 (part 1) and the Gershgorin circle theorem implies the following bounds on the largest and smallest eigenvalues of $\bar{K}$: we have $\lambda_{\max}(\bar{K}) \leq 1 + (n-1)B(L, \delta)$ and $\lambda_{\min}(\bar{K}) \geq 1 - (n-1)B(L, \delta)$, which implies that $\kappa(\bar{K}) \leq \frac{1 + (n-1)B(L,\delta)}{1 - (n-1)B(L,\delta)}$. Since $L \geq L_1(\delta) = \left\lceil \frac{\log(n)}{-\log(1 - \frac{\mu}{2})} \right\rceil + L_0(\delta)$, we have $(n-1)B(L, \delta) \leq \frac{1}{2}$, and then using the inequality $\frac{1+x}{1-x} \leq 1 + 4x$ for $x \in [0, \frac{1}{2}]$, the bound on the condition number follows.

As for part 2, using Theorem 3 (part 2) and the bound $\lambda_{\max}(\bar{K}) \leq 1 + (n-1)B(L, \delta)$, we have $\kappa(\bar{K}) \leq \frac{1 + (n-1)B(L,\delta)}{1 - B(L,\delta)}$. Now if $L \leq L_0(\delta)$, using the definition of $B(L, \delta)$, we have

$$\kappa(\bar{K}) - 1 \leq \frac{n(1 - \delta(1 + \frac{\mu}{2})^L)}{1 - (1 - \delta(1 + \frac{\mu}{2})^L)} \leq \frac{n}{\delta}(1 + \frac{\mu}{2})^{-L}.$$

If $L > L_0(\delta)$, then we have

$$\kappa(\bar{K}) - 1 \leq \frac{\frac{n}{2}(1 - \frac{\mu}{2})^{L - L_0(\delta)}}{1 - \frac{1}{2}(1 - \frac{\mu}{2})^{L - L_0(\delta)}} \leq n(1 - \frac{\mu}{2})^{L - L_0(\delta)} \leq n(1 + \frac{\mu}{2})^{-L}(1 + \frac{\mu}{2})^{L_0(\delta)} \leq \frac{n}{\delta}(1 + \frac{\mu}{2})^{-L},$$

as required. ∎

C.2. Neural Tangent Kernel Matrix

The following formula for the NTK was given by Arora et al. (2019c): defining $\rho := x_i \cdot x_j$, we have

$$\bar{\mathcal{K}}_{ij} = \sum_{h=1}^{L+1} \hat{\sigma}^{(h-1)}(\rho) \left(\prod_{h'=h}^{L} \dot{\hat{\sigma}}(\hat{\sigma}^{(h')}(\rho)) \right). \tag{7}$$

Using this formula, we have the following bound:

Lemma 27 *For any $\rho \in [-1, 1]$, we have $\frac{\dot{\hat{\sigma}}(\rho)}{\dot{\hat{\sigma}}(1)} \leq 1 - \mu(1 - |\rho|)$.*

Proof We have $\hat{\sigma}(\rho) = \dot{\hat{\sigma}}(\rho) = \sum_{i=1}^{\infty} i \cdot a_i^2 \cdot \rho^{i-1}$. This implies that $\dot{\hat{\sigma}}(\rho) \leq \dot{\hat{\sigma}}(|\rho|)$, so it suffices to prove the bound for $\rho \geq 0$. Also note that by definition $\dot{\hat{\sigma}}(\rho)$ is non-negative for all $\rho \geq 0$ as well as an increasing function over $\rho \geq 0$. Thus, using the fact that $\sum_{i=2}^{\infty} a_i^2 = 1 - a_1^2 = \mu$, we have

$$\frac{\dot{\hat{\sigma}}(\rho)}{\dot{\hat{\sigma}}(1)} = \frac{a_1^2 + \sum_{i=2}^{\infty} i \cdot a_i^2 \cdot \rho^i}{a_1^2 + \sum_{i=2}^{\infty} i \cdot a_i^2} \leq \frac{a_1^2 + \sum_{i=2}^{\infty} i \cdot a_i^2 \cdot \rho}{a_1^2 + \sum_{i=2}^{\infty} i \cdot a_i^2} = 1 - \left(\frac{\sum_{i=2}^{\infty} i \cdot a_i^2}{a_1^2 + \sum_{i=2}^{\infty} i \cdot a_i^2}\right)(1 - \rho) \leq 1 - \mu(1 - \rho),$$

as required. ∎

We can now prove Theorem 5 and Theorem 9, which we restate here in a combined form for convenience:

Theorem 28 *The diagonal entries of $\bar{\mathcal{K}}$ are all equal. Furthermore, the following bounds hold if $L \geq 2L_0(\delta)$:*

1. *Under Assumption A, we have $|\bar{\mathcal{K}}_{ij}| \leq 2B(L/2, \delta) \cdot \bar{\mathcal{K}}_{11}$ for all $i, j \in [n]$ with $i \neq j$.*

2. *Under Assumption B, we have $\lambda_{\min}(K) \geq (1 - 2B(L/2, \delta)) \bar{\mathcal{K}}_{11}$.*

Proof First, we show that all diagonal values of $\bar{\mathcal{K}}$ are equal. For every i, we have $x_i \cdot x_i = 1$, and since $\hat{\sigma}^{(h)}(1) = 1$ for any h, we have from (7),

$$\bar{\mathcal{K}}_{ii} = \sum_{h=1}^{L+1} \hat{\sigma}^{(h-1)}(1) \left(\prod_{h'=h}^{L} \dot{\hat{\sigma}}(\hat{\sigma}^{(h')}(1))\right) = \sum_{h=1}^{L+1} \left(\prod_{h'=h}^{L} \dot{\hat{\sigma}}(1)\right) = \frac{\dot{\hat{\sigma}}(1)^{L+1} - 1}{\dot{\hat{\sigma}}(1) - 1},$$

which is a fixed constant.

To prove part 1, let $\rho := x_i \cdot x_j$. It is easy to show (say, via the Hermite expansion of σ) that $\dot{\hat{\sigma}}(1) > 0$. Thus, we have

$$\frac{\bar{\mathcal{K}}_{ij}}{\bar{\mathcal{K}}_{11}} = \frac{\sum_{h=1}^{L+1} \hat{\sigma}^{(h-1)}(\rho) \left(\prod_{h'=h}^{L} \dot{\hat{\sigma}}(\hat{\sigma}^{(h')}(\rho))\right)}{\sum_{h=1}^{L+1} \left(\prod_{h'=h}^{L} \dot{\hat{\sigma}}(1)\right)} \leq \max_{h \in [L+1]} \hat{\sigma}^{(h-1)}(\rho) \cdot \prod_{h'=h}^{L} \frac{\dot{\hat{\sigma}}(\hat{\sigma}^{(h')}(\rho))}{\dot{\hat{\sigma}}(1)}$$

$$\leq \max_{h \in [L+1]} |\hat{\sigma}^{(h-1)}(\rho)| \cdot \prod_{h'=h}^{L} (1 - \mu(1 - |\hat{\sigma}^{(h')}(\rho)|)) \leq \max_{h \in [L+1]} B(h-1, \delta) \cdot \prod_{h'=h}^{L} (1 - \mu(1 - B(h', \delta))),$$

where the penultimate inequality follows Lemma 27 and the final one from Lemma 23. We now show that since $L \geq 2L_0(\delta)$, for any any $h \in [L+1]$, we have

$$B(h-1, \delta) \cdot \prod_{h'=h}^{L} (1 - \mu(1 - B(h', \delta))) \leq 2B(L/2, \delta),$$

which gives the bound of part 1. We do this in two cases: if $h - 1 \geq L/2$, then $B(h-1, \delta) \leq B(L/2, \delta)$, which gives the required bound since all terms in the product are at most 1. Otherwise, if $h - 1 < L/2$, then there are at least $L/2 - L_0(\delta)$ values of h' in $\{h, h+1, \ldots, L\}$ which are larger than $L_0(\delta)$, and for these values of h', we have $B(h', \delta) \leq \frac{1}{2}$, so $1 - \mu(1 - B(h', \delta)) \leq 1 - \frac{\mu}{2}$. The product of these terms is therefore at most $(1 - \frac{\mu}{2})^{L/2 - L_0(\delta)} = 2B(L/2, \delta)$, which gives the required bound in this case.

To prove part 2, define $f : [-1, 1] \to \mathbb{R}$ as $f(\rho) = \sum_{h=1}^{L+1} \hat{\sigma}^{(h-1)}(\rho) \left(\prod_{h'=h}^{L} \dot{\hat{\sigma}}(\hat{\sigma}^{(h')}(\rho))\right)$. Equation (7) shows that this defines a kernel on the unit sphere, and so by Schoenberg's theorem

([Schoenberg, 1942](#)), its power series expansion has only non-negative coefficients. Thus, applying [Lemma 24](#) to $\bar{\mathcal{K}} = f[\underline{K}]$, we conclude that

$$\bar{\mathcal{K}} \succeq (f(1) - f(1-\delta))I_n \succeq (1 - 2B(L/2, \delta))f(1)I_n,$$

using the calculations in part 1. Since $f(1) = \bar{\mathcal{K}}_{11}$, the bound of part 2 follows. ∎

Finally, we prove [Corollary 6](#) and [Corollary 10](#), restated here in a combined form:

Corollary 29 *The following bounds on the condition number $\kappa(\bar{\mathcal{K}})$ hold:*

1. *Under [Assumption A](#), if $L \geq L_2(\delta) := \left\lceil \frac{2\log(2n)}{-\log(1-\frac{\mu}{2})} \right\rceil + 2L_0(\delta)$, then $\kappa(\bar{\mathcal{K}}) \leq 1 + 4n(1 - \frac{\mu}{2})^{L/2 - L_2(\delta)}$.*

2. *Under [Assumption B](#), if $L \geq 4L_0(\delta)$, then $\kappa(\bar{\mathcal{K}}) \leq 1 + \frac{2n}{\delta}(1 + \frac{\mu}{2})^{-L/2}$.*

Proof To prove part 1, note that [Theorem 5](#) (part 1) and the Gershgorin circle theorem implies the following bounds on the largest and smallest eigenvalues of $\bar{\mathcal{K}}$: we have $\lambda_{\max}(\bar{\mathcal{K}}) \leq (1 + 2(n-1)B(L/2, \delta))\bar{\mathcal{K}}_{11}$ and $\lambda_{\min}(\bar{\mathcal{K}}) \geq (1 - 2(n-1)B(L/2, \delta))\bar{\mathcal{K}}_{11}$, which implies that $\kappa(\bar{\mathcal{K}}) \leq \frac{1+2(n-1)B(L/2,\delta)}{1-2(n-1)B(L/2,\delta)}$. Since $L \geq L_2(\delta) = \left\lceil \frac{2\log(2n)}{-\log(1-\frac{\mu}{2})} \right\rceil + 2L_0(\delta)$, we have $2(n-1)B(L/2, \delta) \leq \frac{1}{2}$, and then using the inequality $\frac{1+x}{1-x} \leq 1 + 4x$ for $x \in [0, \frac{1}{2}]$, the bound on the condition number follows.

As for part 2, using [Theorem 5](#) (part 2) and the bound $\lambda_{\max}(\bar{\mathcal{K}}) \leq (1 + 2(n-1)B(L/2, \delta))\bar{\mathcal{K}}_{11}$, we have $\kappa(\bar{\mathcal{K}}) \leq \frac{1+2(n-1)B(L/2,\delta)}{1-2B(L/2,\delta)}$. Thus,

$$\kappa(\bar{\mathcal{K}}) - 1 \leq \frac{n(1 - \frac{\mu}{2})^{L/2 - L_0(\delta)}}{1 - (1 - \frac{\mu}{2})^{L/2 - L_0(\delta)}} \leq 2n(1 - \frac{\mu}{2})^{L/2 - L_0(\delta)} \leq 2n(1 + \frac{\mu}{2})^{-L/2}(1 + \frac{\mu}{2})^{L_0(\delta)} \leq \frac{2n}{\delta}(1 + \frac{\mu}{2})^{-L/2},$$

the second inequality follows since $L/2 - L_0(\delta) \geq L_0(\delta)$, and so $(1 - \frac{\mu}{2})^{L/2 - L_0(\delta)} \leq (1 - \frac{\mu}{2})^{L_0(\delta)} \leq \frac{1}{2}$. ∎

C.3. Evolution of dot products for uncentered activations

In this section, we generalize the analysis in the beginning of [Appendix C](#) to activations that need not be normalized ([2](#)). Specifically, we only assume the following condition on the activation σ:

$$\underset{X \sim \mathcal{N}(0,1)}{\mathbb{E}}[\sigma^2(X)] = 1, \tag{8}$$

i.e., we allow $\mathbb{E}_{X \sim \mathcal{N}(0,1)}[\sigma(X)]$ to be non-zero. In this case, we will show that there is a certain value $\bar{\rho} \in [0, 1]$ such that dot products of input representations at the top layer converge to $\bar{\rho}$ as the depth increases. Furthermore, when $\bar{\rho} < 1$, this convergence is at an *exponential* rate governed by the following notion of the coefficient of non-affinity which accounts for the non-zero expectation of σ under standard Gaussian inputs:

Definition 30 *The* coefficient of non-affinity *of the activation function σ is defined to be $\tilde{\mu} := 1 - \left(\mathbb{E}_{X \sim \mathcal{N}(0,1)}[\sigma(X)]\right)^2 - \left(\mathbb{E}_{X \sim \mathcal{N}(0,1)}[X\sigma(X)]\right)^2.$*

Clearly, for normalized activations, the two coefficients coincide, i.e. $\mu = \tilde{\mu}$. Let $\sigma(u) = \sum_{i=0}^{\infty} a_i h_i(u)$ as usual, and then the dual activation is $\hat{\sigma}(\rho) = \sum_{i=0}^{\infty} a_i^2 \rho^i$. The normalization (8) implies that $\sum_{i=0}^{\infty} a_i^2 = 1$. Furthermore, since $h_0(u) \equiv 1$ and $h_1(u) = u$, we have

$$a_0 = \underset{X \sim \mathcal{N}(0,1)}{\mathbb{E}} [\sigma(X)] \quad \text{and} \quad a_1 = \underset{X \sim \mathcal{N}(0,1)}{\mathbb{E}} [X\sigma(X)],$$

and so $\tilde{\mu} = 1 - a_0^2 - a_1^2$. We call an activation function σ "non-affine" if it cannot be written as $u \mapsto a + bu$ for constants a and b. It is easy to see from the Hermite expansion of σ that this is equivalent to saying that for some $i > 1$, $a_i \neq 0$. We then have the following analogue of Lemma 20:

Lemma 31 *For any non-affine activation function σ, we have $0 < \tilde{\mu} \leq 1$.*

Proof Since σ is non-affine, for at least one $i > 1$, we have $a_i \neq 0$. This, coupled with the fact that $\sum_{i=0}^{\infty} a_i^2 = 1$ implies that $\tilde{\mu} = 1 - a_0^2 - a_1^2 \in (0, 1]$. ∎

The following property of $\hat{\sigma}$ immediately follows from the formula $\hat{\sigma}(\rho) = \sum_{i=0}^{\infty} a_i^2 \rho^i$:

Lemma 32 *The dual activation $\hat{\sigma}$ is non-decreasing and convex on $[0, 1]$.*

Note that we always have $\hat{\sigma}(1) = 1$. We now have the following important property regarding the other fixed points of $\hat{\sigma}$:

Lemma 33 *Let $\bar{\rho} \in [0, 1]$ such that $\hat{\sigma}(\bar{\rho}) = \bar{\rho}$. Then for any $\lambda \in [0, 1]$, if $\rho_\lambda := \lambda + (1 - \lambda)\bar{\rho}$, then*

$$\rho_\lambda - \hat{\sigma}(\rho_\lambda) \geq \tilde{\mu}\lambda(1 - \lambda)(1 - \bar{\rho})^2.$$

Thus, if σ is non-affine, then there can be at most one fixed point of $\hat{\sigma}$ in $[0, 1)$, i.e. a point $\bar{\rho} \in [0, 1)$ such that $\hat{\sigma}(\bar{\rho}) = \bar{\rho}$.

Proof First, since $\bar{\rho}$ and 1 are both fixed points of $\hat{\sigma}$, we have

$$\rho_\lambda = \lambda + (1 - \lambda)\bar{\rho} = \lambda\hat{\sigma}(1) + (1 - \lambda)\hat{\sigma}(\bar{\rho}) = \sum_{i=0}^{\infty} a_i^2(\lambda + (1 - \lambda)\bar{\rho}^i), \tag{9}$$

whereas

$$\hat{\sigma}(\rho_\lambda) = \sum_{i=0}^{\infty} a_i^2(\lambda + (1 - \lambda)\bar{\rho})^i. \tag{10}$$

Now, we show that the sequence $(\lambda + (1-\lambda)\bar{\rho}^i) - (\lambda + (1-\lambda)\bar{\rho})^i$ for $i = 0, 1, 2, \ldots$ is non-decreasing. This is because by direct calculation,

$$[(\lambda + (1 - \lambda)\bar{\rho}^{i+1}) - (\lambda + (1 - \lambda)\bar{\rho})^{i+1}] - [(\lambda + (1 - \lambda)\bar{\rho}^i) - (\lambda + (1 - \lambda)\bar{\rho})^i]$$
$$= (1 - \lambda)(1 - \bar{\rho})((\lambda + (1 - \lambda)\bar{\rho})^i - \bar{\rho}^i)$$
$$\geq 0,$$

since $\bar{\rho} \leq \lambda + (1 - \lambda)\bar{\rho}$. Thus, for all $i \geq 2$, we have

$$(\lambda + (1 - \lambda)\bar{\rho}^i) - (\lambda + (1 - \lambda)\bar{\rho})^i \geq (\lambda + (1 - \lambda)\bar{\rho}^2) - (\lambda + (1 - \lambda)\bar{\rho})^2 = \lambda(1 - \lambda)(1 - \bar{\rho})^2. \tag{11}$$

Hence, using (9) and (10), we have

$$
\begin{aligned}
\rho_\lambda - \hat{\sigma}(\rho_\lambda) &= \sum_{i=0}^{\infty} a_i^2 \big[(\lambda + (1-\lambda)\bar{\rho}^i) - (\lambda + (1-\lambda)\bar{\rho})^i \big] \\
&= \sum_{i=2}^{\infty} a_i^2 \big[(\lambda + (1-\lambda)\bar{\rho}^i) - (\lambda + (1-\lambda)\bar{\rho})^i \big] \\
&\geq \sum_{i=2}^{\infty} a_i^2 \lambda (1-\lambda)(1-\bar{\rho})^2 \qquad\qquad \text{(Using (11))} \\
&= \tilde{\mu}\lambda(1-\lambda)(1-\bar{\rho})^2,
\end{aligned}
$$

as required. ∎

Since $\hat{\sigma}(1) = 1$, 1 is a fixed point of $\hat{\sigma}$. The above lemma shows that for non-affine σ, $\hat{\sigma}$ can have at most one more fixed point in $[0,1]$. This fact also gives us the following useful consequence:

Lemma 34 *Let σ be a non-affine activation, and let $\bar{\rho} \in [0,1]$ be the smallest fixed point of $\hat{\sigma}$. Then, if $\bar{\rho} < 1$, then $\dot{\hat{\sigma}}(\bar{\rho}) < 1$. If $\bar{\rho} = 1$, then $\dot{\hat{\sigma}}(1) \leq 1$.*

Proof Suppose $\bar{\rho} < 1$. If $\dot{\hat{\sigma}}(\bar{\rho}) \geq 1$, then since $\hat{\sigma}(1) = 1$, the mean value theorem and the convexity of $\hat{\sigma}$ imply that $\dot{\hat{\sigma}}(\bar{\rho}) = 1$, which implies that all $\rho \in [\bar{\rho}, 1]$ are fixed points of $\hat{\sigma}$, which is a contradiction by Lemma 32.

Next, if $\bar{\rho} = 1$, then there is no fixed point of $\hat{\sigma}$ in $[0,1)$. If $\dot{\hat{\sigma}}(1) > 1$, then there exists a value $\rho < 1$ such that $\hat{\sigma}(\rho) < \rho$. Since $\hat{\sigma}(0) = a_0^2 \geq 0$, and $\hat{\sigma}$ is continuous, we conclude that there exists a value $\rho' \in [0, \rho)$ such that $\hat{\sigma}(\rho') = \rho'$, a contradiction. ∎

We can now prove an analogue of Lemma 22:

Lemma 35 *Suppose σ is a non-affine activation. Let $\bar{\rho}$ be the smallest fixed point of $\hat{\sigma}$ in $[0,1]$, and $\delta \in (0,1]$. Then, the following bounds hold:*

- *If $\rho \in [\frac{1+\bar{\rho}}{2}, 1]$, then*

$$
\hat{\sigma}(\rho) \leq \rho - \frac{\tilde{\mu}}{2}(1-\bar{\rho})(1-\rho).
$$

- *If $\rho \in [\bar{\rho}, \frac{1+\bar{\rho}}{2})$, then*

$$
|\hat{\sigma}(\rho) - \bar{\rho}| \leq (1 - \tfrac{\tilde{\mu}}{2}(1-\bar{\rho}))|\rho - \bar{\rho}|.
$$

- *If $\rho \in [0, \bar{\rho})$, then*

$$
|\hat{\sigma}(\rho) - \bar{\rho}| \leq \dot{\hat{\sigma}}(\bar{\rho})|\rho - \bar{\rho}|.
$$

- *If $\rho \in [-1, 0)$, then $|\hat{\sigma}(\rho)| \leq \hat{\sigma}(-\rho)$, and*

$$
\hat{\sigma}(\rho) \geq \rho + \hat{\sigma}(0).
$$

Proof We analyze the four cases separately:

Case 1: $\rho \in \left[\frac{1+\bar\rho}{2}, 1\right]$. In this case, writing ρ as a convex combination of $\frac{1+\bar\rho}{2}$ and 1 and using the convexity of $\hat\sigma$ in $[0,1]$, we have

$$\hat\sigma(\rho) \le \frac{2-2\rho}{1-\bar\rho}\hat\sigma\left(\frac{1+\bar\rho}{2}\right) + \frac{2\rho-1-\bar\rho}{1-\bar\rho}\hat\sigma(1) \le \frac{2-2\rho}{1-\bar\rho}\left(\frac{1+\bar\rho}{2} - \frac{\tilde\mu}{4}(1-\bar\rho)^2\right) + \frac{2\rho-1-\bar\rho}{1-\bar\rho} = \rho - \frac{\tilde\mu}{2}(1-\bar\rho)(1-\rho).$$

The second inequality above follows from Lemma 33.

Case 2: $\rho \in \left[\bar\rho, \frac{1+\bar\rho}{2}\right)$. Again, writing ρ as a convex combination of $\bar\rho$ and $\frac{1+\bar\rho}{2}$ and using the convexity of $\hat\sigma$ in $[0,1]$, we have

$$\hat\sigma(\rho) \le \frac{1+\bar\rho-2\rho}{1-\bar\rho}\hat\sigma(\bar\rho) + \frac{2\rho-2\bar\rho}{1-\bar\rho}\hat\sigma\left(\frac{1+\bar\rho}{2}\right) \le \frac{1+\bar\rho-2\rho}{1-\bar\rho}\cdot\bar\rho + \frac{2\rho-2\bar\rho}{1-\bar\rho}\left(\frac{1+\bar\rho}{2} - \frac{\tilde\mu}{4}(1-\bar\rho)^2\right) = \rho - \frac{\tilde\mu}{2}(1-\bar\rho)(\rho-\bar\rho).$$

The second inequality above follows from Lemma 33. Finally, since $\hat\sigma$ is non-decreasing, and $\rho \ge \bar\rho$, we must have $\hat\sigma(\rho) \ge \hat\sigma(\bar\rho) = \bar\rho$. Hence, we have

$$|\hat\sigma(\rho) - \bar\rho| = \hat\sigma(\rho) - \bar\rho \le \rho - \frac{\tilde\mu}{2}(1-\bar\rho)(\rho-\bar\rho) - \bar\rho = \left(1 - \frac{\tilde\mu}{2}(1-\bar\rho)\right)|\rho - \bar\rho|,$$

as required.

Case 3: $\rho \in [0, \bar\rho)$. In this case, the convexity of $\hat\sigma$ in $[0,1]$ implies that

$$\hat\sigma(\rho) \ge \hat\sigma(\bar\rho) + \dot{\hat\sigma}(\bar\rho)(\rho-\bar\rho) = \bar\rho + \dot{\hat\sigma}(\bar\rho)(\rho-\bar\rho),$$

or in other words,

$$\bar\rho - \hat\sigma(\rho) \le \dot{\hat\sigma}(\bar\rho)(\bar\rho-\rho).$$

Since $\hat\sigma$ is non-decreasing, and $\rho < \bar\rho$, we must have $\hat\sigma(\rho) \le \hat\sigma(\bar\rho) = \bar\rho$. Hence, we have

$$|\hat\sigma(\rho) - \bar\rho| = \bar\rho - \hat\sigma(\rho) \le \dot{\hat\sigma}(\bar\rho)(\bar\rho-\rho) = \dot{\hat\sigma}(\bar\rho)|\rho-\bar\rho|,$$

as required.

Case 4: $\rho \in [-1, 0)$. The bound $|\hat\sigma(\rho)| \le \hat\sigma(-\rho)$ is obvious from the fact that $\hat\sigma(\rho) = \sum_{i=0}^\infty a_i^2\rho^i$. Next, we have

$$\hat\sigma(\rho) - \rho = \sum_{i=0}^\infty a_i^2\rho^i - \sum_{i=0}^\infty a_i^2\rho = a_0^2(1-\rho) + \sum_{i=1}^\infty a_i^2(\rho^i-\rho) \ge a_0^2 = \hat\sigma(0).$$

The last inequality follows because $1 - \rho \ge 0$ and $\rho^i - \rho \ge 0$. ∎

Finally, we now have the following generalization of Lemma 23 showing exponentially fast convergence of dot products to the smallest fixed point of $\hat\sigma$ in $[0,1]$. We focus on the case when $\hat\sigma(0) > 0$, since the case when $\hat\sigma(0) = 0$ exactly corresponds to normalized activations, which we have already analyzed.

Lemma 36 (Correlation convergence lemma) *Suppose σ is a non-affine activation with $\hat\sigma(0) > 0$. Let $\bar\rho$ be the smallest fixed point of $\hat\sigma$ in $[0,1]$, and $\delta \in (0,1]$. If $\bar\rho = 1$, then assume further that $\dot{\hat\sigma}(1) < 1$. Then, after $L \ge L_0 := \max\left\{\left\lceil\frac{\log(\frac{1-\bar\rho}{2\delta})}{\log(1+\frac{\tilde\mu(1-\bar\rho)}{2})}\right\rceil, \left\lceil\frac{1}{\hat\sigma(0)}\right\rceil\right\}$ layers, for any ρ such that $|\rho| \le 1 - \delta$, we have*

$$|\hat\sigma^{(L)}(\rho) - \bar\rho| \le \max\left\{1 - \frac{\tilde\mu(1-\bar\rho)}{2}, \dot{\hat\sigma}(\bar\rho)\right\}^{L-L_0}\left(\frac{1+\bar\rho}{2}\right).$$

Finally, if $\bar\rho = 1$ and $\dot{\hat\sigma}(1) = 1$, then

30

Proof If $\rho \in (\frac{1+\bar{\rho}}{2}, 1-\delta]$, then using case 1 of Lemma 35, after $L_1 := \left\lceil \frac{\log(\frac{1-\bar{\rho}}{2\delta})}{\log(1+\frac{\tilde{\mu}(1-\bar{\rho})}{2})} \right\rceil$ layers, we have $\hat{\sigma}^{(L_1)}(\rho) \leq \frac{1+\bar{\rho}}{2}$. Similarly, if $\rho < 0$, then using case 4 of Lemma 35, after $L_2 := \left\lceil \frac{1}{\hat{\sigma}(0)} \right\rceil$ layers, we have $\hat{\sigma}^{(L_2)}(\rho) \geq 0$. Finally, when $\rho \in [0, \frac{1+\bar{\rho}}{2}]$, then using cases 2 and 3 of Lemma 35, we conclude that

$$|\hat{\sigma}(\rho) - \bar{\rho}| \leq \max\left\{1 - \frac{\tilde{\mu}(1-\bar{\rho})}{2}, \dot{\hat{\sigma}}(\bar{\rho})\right\}|\rho - \bar{\rho}|.$$

The statement of the lemma follows from these observations. ∎

The only setting not covered by Lemma 36 is when $\bar{\rho} = 1$ and $\dot{\hat{\sigma}}(1) = 1$. This case is handled separately in the lemma below:

Lemma 37 *Suppose σ is a non-affine activation with $\hat{\sigma}(0) > 0$. Suppose 1 is the unique fixed point of $\hat{\sigma}$ in $[0, 1]$ and also $\dot{\hat{\sigma}}(1) = 1$. Then, after $L \geq L_0' := \left\lceil \frac{1}{\hat{\sigma}(0)} \right\rceil$ layers, for any $\rho \in [-1, 1]$, we have*

$$1 - \hat{\sigma}^{(L)}(\rho) \leq \frac{\log(2e\tilde{\mu}(L-L_0'))}{2\tilde{\mu}(L-L_0')}.$$

Proof First, we note that since 1 is the unique fixed point of $\hat{\sigma}$ in $[0, 1]$, we must have $\hat{\sigma}(\rho) > \rho$ for all $\rho \in [0, 1)$. Also, for any $\rho < 0$, as in case 4 of Lemma 35, we have $\hat{\sigma}(\rho) \geq \rho + \hat{\sigma}(0)$. So for any ρ, an application of $\hat{\sigma}$ never decreases its value.

Since $\dot{\hat{\sigma}}(1) = 1$, we have $\sum_{i=1}^{\infty} ia_i^2 = 1$. Hence, for any $\alpha \in [0, 1]$, we have

$$1 - \dot{\hat{\sigma}}(1-\alpha) = \sum_{i=1}^{\infty} ia_i^2(1 - (1-\alpha)^{i-1}) = \sum_{i=2}^{\infty} ia_i^2(1 - (1-\alpha)^{i-1}) \geq \sum_{i=2}^{\infty} 2a_i^2(1 - (1-\alpha)) = 2\alpha\tilde{\mu}.$$

Next, since $\hat{\sigma}$ is convex in $[0, 1]$ and $\dot{\hat{\sigma}}(1) = 1$, for any $\rho \in [0, 1]$, we have $\dot{\hat{\sigma}}(\rho) \leq 1$. Thus, $\hat{\sigma}(1-\alpha) \geq \hat{\sigma}(1) - \alpha = 1 - \alpha$. Again using the convexity of $\hat{\sigma}$ in $[0, 1]$, for any $\rho \in [0, 1-\alpha]$, we have

$$\hat{\sigma}(\rho) \geq \hat{\sigma}(1-\alpha) + \dot{\hat{\sigma}}(1-\alpha) \cdot (\rho - (1-\alpha)) \geq 1 - \alpha + (1 - 2\alpha\tilde{\mu})(\rho - (1-\alpha)).$$

The second inequality above uses the facts that $\hat{\sigma}(1-\alpha) \geq 1-\alpha$, $\dot{\hat{\sigma}}(1-\alpha) \leq 1-2\alpha\tilde{\mu}$, and $\rho - (1-\alpha) \leq 0$. Simplifying and rearranging, we have

$$1 - \hat{\sigma}(\rho) \leq 2\alpha^2\tilde{\mu} + (1 - 2\alpha\tilde{\mu})(1 - \rho).$$

Via a simple induction on the number of layers L, if $\rho \in [0, 1]$, the above bound implies that

$$1 - \hat{\sigma}^{(L)}(\rho) \leq \alpha + (1 - 2\alpha\tilde{\mu})^L(1 - \rho) \leq \alpha + e^{-2\alpha\tilde{\mu}L}.$$

The RHS above is minimized for $\alpha = \frac{\log(2\tilde{\mu}L)}{2\tilde{\mu}L}$, which yields the bound

$$1 - \hat{\sigma}^{(L)}(\rho) \leq \frac{\log(2e\tilde{\mu}L)}{2\tilde{\mu}L}.$$

Finally, if $\rho < 0$, then after $L_0' := \left\lceil \frac{1}{\hat{\sigma}(0)} \right\rceil$ layers, we reach a non-negative value, at which point the above analysis applies. The lemma follows by accounting for these L_0' layers. ∎

Theorem 12 now follows immediately from Lemma 36 and Lemma 37.

C.4. Calculations of the coefficient of non-linearity

For standard activation functions such as ReLU, it is easy to compute the coefficient of non-linearity of their normalized versions from their Hermite expansion. Specifically, if $\sigma(u) = \sum_{i=0}^{\infty} a_i h_i(u)$ is the Hermite expansion of an activation function σ, then the normalized version of σ, denoted $\bar{\sigma}$, is given by

$$\bar{\sigma}(u) = \frac{\sum_{i=1}^{\infty} a_i h_i(u)}{\sqrt{\sum_{i=1}^{\infty} a_i^2}}.$$

Thus, the coefficient of non-linearity of $\bar{\sigma}$ is given by (see the proof of Lemma 20):

$$\mu = 1 - \frac{a_1^2}{\sum_{i=1}^{\infty} a_i^2}.$$

Since the dual activation of σ, $\hat{\sigma}(\rho) = \sum_{i=0}^{\infty} a_i^2 \rho^i$, can be written as $\hat{\sigma}(\rho) = \sum_{i=0}^{\infty} a_i^2 \rho^i$, we can also write the above formula for the coefficient of non-linearity as:

$$\mu = 1 - \frac{\hat{\sigma}'(0)}{\hat{\sigma}(1) - \hat{\sigma}(0)}.$$

This latter formula easily allows us to compute the coefficient of non-linearity for various activations. For example, using Table 1 from (Daniely et al., 2016), we get the following calculations of μ:

Activation	$\sigma(u)$	$\hat{\sigma}(\rho)$	μ
Identity	u	ρ	0
2nd Hermite	$\frac{u^2-1}{\sqrt{2}}$	ρ^2	1
ReLU	$\max\{u,0\}$	$\frac{1}{2\pi} + \frac{\rho}{4} + \frac{\rho^2}{2\pi} + \frac{\rho^4}{48\pi} + \cdots = \frac{\sqrt{1-\rho^2}+(\pi-\cos^{-1}(\rho))\rho}{2\pi}$	$\frac{\pi-2}{2\pi-2}$
Step	$\mathbf{1}[u \geq 0]$	$\frac{1}{4} + \frac{\rho}{2\pi} + \frac{\rho^3}{12\pi} + \frac{3\rho^5}{80\pi} + \cdots = \frac{\pi-\cos^{-1}(\rho)}{2\pi}$	$\frac{\pi-2}{\pi}$
Exponential	e^u	$e^3(1 + \rho + \frac{\rho^2}{2} + \frac{\rho^3}{6} + \cdots) = e^{\rho+3}$	$\frac{e-2}{e-1}$

For the activation function NormReLU_c (defined in Section 7), it is easier to directly compute the coefficient of non-linearity as follows. First, note that NormReLU_c is already normalized, so it suffices to compute the coefficient a_1 in its Hermite expansion. We have

$$
\begin{aligned}
a_1 &= \mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,1)} \left[X \text{NormReLU}_c(X) \right] \\
&= \mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,1)} \left[X \cdot \lambda(c) \cdot \left[\max\{X - c, 0\} + b(c) \right] \right] \\
&= \mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,1)} \left[\lambda(c) \cdot \mathbf{1}[X \geq c] \right] \\
&= \lambda(c)(1 - \Phi(c)),
\end{aligned}
$$

where the third step uses Stein's lemma, and in the fourth step, $\Phi(\cdot)$ is the Gaussian cumulative distribution function. Thus, the coefficient of non-linearity for NormReLU_c is

$$\mu = 1 - \lambda(c)^2 (1 - \Phi(c))^2.$$

For the value of c used in our experiments, i.e. $c = -1.5975$, we have $\lambda(c) \approx 1.05$, and $\Phi(c) \approx 0.0551$, and thus $\mu \approx 0.0156$.

Appendix D. Analysis for Non-Unit Length Inputs - Proof of Theorem 13

In this section we provide a proof for Theorem 13.

D.1. Analyis for Norms

As stated earlier our analysis begins first by analyzing and understanding the evolution of the norm of the representations across the network. To this end we prove the following theorem.

Theorem 38 *Let σ be a twice-differentiable non-decreasing activation function which satisfies the following properties:*

- $\sigma(0) = 0$

- σ *is concave on $\mathbb{R}^+$ and σ is convex on $\mathbb{R}^-$*

Furthermore define

$$\alpha_\sigma := \min\left(2 \mathop{\mathbb{E}}_{x \in \mathcal{N}(0,1)}\left[\sigma^2\left(\sqrt{0.5}x\right)\right] - 1, 1 - \mathop{\mathbb{E}}_{x \in \mathcal{N}(0,1)}\left[\sigma(x)\sigma'(x)x\right]\right).$$

We have that if $\|x\|^2 \in [0.5, \infty)$ then for any $L \geq 0$

$$\left|\|\bar{k}(x,x)\|^2 - 1\right| \leq (1 - \alpha_\sigma)^L \cdot \|x\|^2 - 1|^2$$

Remark 39 *From the proof of the theorem it will be evident under our assumptions on σ that $\alpha_\sigma \in [0,1]$. Furthermore it will also be evident that the choice of 0.5 is arbitrary and can be replaced by any constant > 0, and the definition of α_σ changes appropriately.*

In the rest of the section we prove Theorem 38. Firstly note that it is sufficient to prove the theorem for $L = 1$. The general case follows inductively easily. To prove the base case consider the following function $\hat{\sigma}_l : \mathbb{R}^+ \to \mathbb{R}^+$

$$\hat{\sigma}_l(\gamma) := \mathop{\mathbb{E}}_{z \sim N(0,\gamma)}\left[\sigma^2(z)\right] = \mathop{\mathbb{E}}_{z \sim N(0,1)}\left[\sigma^2(\sqrt{\gamma}z)\right]$$

A simple application of successive central limit theorems across layers gives us that for any l

$$\bar{k}^{(l)}(x,x) = \hat{\sigma}_l^l(\|x\|^2)$$

Therefore all we are required to show is that for all $\gamma \geq 0.5$ we have that

$$|\hat{\sigma}_l(\gamma) - 1| \leq (1 - \alpha_\sigma)|\gamma - 1| \tag{12}$$

As a reminder note that our assumption on σ implies that $\hat{\sigma}_l(1) = 1$. To establish (12) we begin with the following lemma characterizing the behaviour of the map $\hat{\sigma}_l$.

Lemma 40 *Let σ be a twice-differentiable monotonically increasing activation function which satisfies the following properties:*

- $\sigma(0) = 0$

33

- σ is concave on $\mathbb{R}^+$ and σ is convex on $\mathbb{R}^-$

Then $\hat{\sigma}_l(\gamma)$ is a twice-differentiable non-decreasing concave function on $\mathbb{R}^+$.

We now prove (12) using Lemma 40. We divide the analysis in two cases. Suppose $\gamma \in [0.5, 1]$. Note that since $\hat{\sigma}_l(1) = 1$ and σ is monotonic we have that $\hat{\sigma}_l(\gamma) \leq 1$. Furthermore, since $\hat{\sigma}_l$ is a concave function and $\hat{\sigma}_l(1) = 1$, we have that

$$\hat{\sigma}_l(\gamma) \geq 2(1 - \gamma)\hat{\sigma}_l(0.5) + 2(\gamma - 0.5)$$

It now follows that

$$1 - \hat{\sigma}_l(\gamma) \leq (1 - (2\hat{\sigma}_l(0.5) - 1))(1 - \gamma) \leq (1 - \alpha_\sigma)(1 - \gamma). \tag{13}$$

Note that the concavity and monotonicity of $\hat{\sigma}_l$ also establish that $2\hat{\sigma}_l(0.5) - 1 \in [0, 1]$.

For the case of $\gamma \geq 1$. Note that since $\hat{\sigma}_l$ is concave and it is easy to see that

$$\hat{\sigma}_l(\gamma) \leq \hat{\sigma}_l(1) + \hat{\sigma}_l'(1)(\gamma - 1),$$

which by noting that $\hat{\sigma}_l(1) = 1$ implies that

$$\hat{\sigma}_l(\gamma) - 1 \leq \hat{\sigma}_l'(1)(\gamma - 1) \leq (1 - \alpha_\sigma)(\gamma - 1).$$

The last inequality follows by noting that $\hat{\sigma}_l'(1) = \mathbb{E}_{x \in \mathcal{N}(0,1)}\left[\sigma\left(x\right)\sigma'(x)x\right]$. Again by concavity of $\hat{\sigma}$ and the conditions that $\hat{\sigma}(0) = 0$ and $\hat{\sigma}(1) = 1$, it can be readily seen that $\hat{\sigma}'(1) \leq 1$.

This finishes the proof of Theorem 38. We finish this section by provide the proof of Lemma 40
Proof [Proof of Lemma 40] In the rest of the proof we assume $\gamma > 0$. The calculation for $\gamma = 0$ case can be done analogously. The twice differentiability of $\hat{\sigma}_l$ can be seen easily from the definition.

Further note that

$$\frac{\partial \hat{\sigma}(\gamma)}{\partial \gamma} = \mathbb{E}_{z \sim N(0,1)}\left[\sigma(\sqrt{\gamma}z)\sigma'(\sqrt{\gamma}z)\frac{z}{\sqrt{\gamma}}\right] \geq 0$$

The inequality follows since $\sigma(0) = 0$ and σ is non-decreasing (and hence $\sigma(x) \cdot x \geq 0$). Furthermore consider the computation for the second derivative

$$\frac{\partial^2 \hat{\sigma}(\gamma)}{\partial \gamma^2} = \frac{\mathbb{E}_{z \sim N(0,1)}\left[(\sigma'(\sqrt{\gamma}z))^2 z^2\right]}{2\gamma} + \frac{\mathbb{E}_{z \sim N(0,1)}\left[\sigma(\sqrt{\gamma}z)\sigma''(\sqrt{\gamma}z)z^2\right]}{2\gamma} - \frac{\mathbb{E}_{z \sim N(0,1)}\left[\sigma(\sqrt{\gamma}z)\sigma'(\sqrt{\gamma}z)z\right]}{2\gamma^{3/2}}$$

We can now analyze by considering every term. Notice that the second term

$$\frac{\mathbb{E}_{z \sim N(0,1)}\left[\sigma(\sqrt{\gamma}z)\sigma''(\sqrt{\gamma}z)z^2\right]}{2\gamma} \leq 0$$

because under the assumptions $\sigma(x)$ and $\sigma''(x)$ always have opposite signs. We will now analyze the sum of the first and third terms.

$$\frac{\mathbb{E}_{z \sim N(0,1)}\left[(\sigma'(\sqrt{\gamma}z))^2 z^2\right]}{2\gamma} - \frac{\mathbb{E}_{z \sim N(0,1)}\left[\sigma(\sqrt{\gamma}z)\sigma'(\sqrt{\gamma}z)z\right]}{2\gamma^{3/2}}$$

$$= \frac{\mathbb{E}_{z \sim N(0,1)}\left[(\sigma'(\sqrt{\gamma}z))^2 z^2 \gamma - \sigma(\sqrt{\gamma}z)\sigma'(\sqrt{\gamma}z)z\sqrt{\gamma}\right]}{2\gamma^2}$$

$$= \frac{\mathbb{E}_{z \sim N(0,\gamma)}\left[(\sigma'(z))^2 z^2 - \sigma(z)\sigma'(z)z\right]}{2\gamma^2}$$

We now show that $\forall z, (\sigma'(z))^2 z^2 - \sigma(z)\sigma'(z)z \geq 0$. To this end note that $\forall z \geq 0$

$$(\sigma'(z))^2 z^2 - \sigma(z)\sigma'(z)z = z\sigma'(z)(\sigma'(z)z - \sigma(z)) \leq 0$$

The inequality follows by noting monotonicity and concavity of σ for $z \geq 0$. Similarly for $z \leq 0$,

$$(\sigma'(z))^2 z^2 - \sigma(z)\sigma'(z)z = z\sigma'(z)(\sigma'(z)z - \sigma(z)) \leq 0$$

which follows by noting monotonicity and convexity of σ for $z \leq 0$. ∎

D.2. Analysis for Dot-Products - Preliminaries

Having established that the norms converge to 1 we focus on the normalized dot-product between the representations and show that for odd functions it never increases and once the norms have converged close to 1 it decreases rapidly. To this end we will first need to define the following function which is a generalization of the $\hat{\sigma}$ function defined in Appendix C. For any two vectors x, y define $\hat{\sigma}_c(\cdot, \cdot)$ as

$$\hat{\sigma}_c(x, y) := \mathbb{E}_{w \sim \mathcal{N}(0, I)}[\sigma(w^\top x)\sigma(w^\top y)]$$

A simple parametrization shows that this is equivalent to the following quantity

$$\hat{\sigma}_c(x, y) = \mathbb{E}_{z \sim N(0, \Sigma(x,y))}[\sigma(z_1)\sigma(z_2)] \quad \text{where} \quad \Sigma(x, y) = \begin{bmatrix} \|x\|^2 & x \cdot y \\ x \cdot y & \|y\|^2 \end{bmatrix}$$

To analyze the above quantity we use a general notion of (probabilist's) Hermite polynomials $h_j^{\{\gamma\}}(x)$ defined for any γ, defined to be the Hermite polynomials corresponding to the base distribution being $\mathcal{N}(0, \gamma)$. We use the following specific definition derived from O'Donnell (2014).

Consider for any γ the quantity $exp(tz - \frac{\gamma^2 t^2}{2})$. Considering the power series we get that the coefficient in front of t^j is a polynomial in z (with coefficents depending on γ). Defining this polnomial as $H_j^{\{\gamma\}}(z)$ we get that

$$exp(tz - \frac{\gamma^2 t^2}{2}) = \sum \frac{1}{j!} H_j^{\{\gamma\}}(z) \cdot t^j \tag{14}$$

We can now define the hermite polynomials for any $\gamma \geq 0$ and $j \in \{0, 1, \ldots\}$ formally as

$$h_j^{\{\gamma\}}(z) := \frac{H_j^{\{\gamma\}}(z)}{\sqrt{\gamma^j \cdot j!}}$$

We show the following simple lemma about these polynomials (which also establishes that these polynomials form a basis under the distribution $\mathcal{N}(0, \gamma)$).

Lemma 41 *Given three numbers $\gamma_1 > 0, \gamma_2 > 0, \gamma_3$ with $\gamma_3^2 \leq \gamma_1\gamma_2$, define*

$$\Sigma := \begin{bmatrix} \gamma_1 & \gamma_3 \\ \gamma_3 & \gamma_2 \end{bmatrix}$$

We have that the polynomial family $\{h_j^{\{\gamma\}}\}$ satisfies the following condition

$$\mathbb{E}_{z \sim \mathcal{N}(0, \Sigma)}[h_i^{\gamma_1}(z_1)h_j^{\gamma_2}(z_2)] = \begin{cases} \left(\frac{\gamma_3}{\sqrt{\gamma_1\gamma_2}}\right)^j & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

35

Note that setting $\gamma_1 = \gamma_2 = \gamma_3 = \gamma$ also establishes that for any $\gamma \geq 0$, $\{h_j^{\{\gamma\}}\}$ is an orthonormal family of polynomials. Therefore we can decompose σ in the basis given by $h_j^{\{\gamma\}}$ as follows

$$\sigma(x) = \sum_{j=0}^{\infty} a_j^{\{\gamma\}} h_j^{\{\gamma\}}(x) \quad \text{where} \quad a_j^{\{\gamma\}} = E_{z \sim \mathcal{N}(0,\gamma)}[\sigma(z) h_j^{\{\gamma\}}(z)]$$

Using Lemma 41 the following statement follows by decomposing in the appropriate bases

$$\hat{\sigma}_c(x,y) = \sum_{j=0}^{\infty} a_j^{\{\|x\|^2\}} a_j^{\{\|y\|^2\}} \rho(x,y)^i \quad \text{where} \quad \rho(x,y) := \frac{x \cdot y}{\|x\|\|y\|} \tag{15}$$

A special case of the above $\hat{\sigma}_c$ is when $x = y$. In this case it is easy see that $\hat{\sigma}_c(x,x) = \hat{\sigma}_l(\|x\|^2)$ (defined in the previous section). Accordingly

$$\hat{\sigma}_c(x,x) = \hat{\sigma}_l(\|x\|^2) = \sum_{j=0}^{\infty} (a_j^{\{\|x\|^2\}})^2 \tag{16}$$

We now show the following lemma which is a generalization of Lemma 22.

Lemma 42 *Given any number $\gamma > 0$ define the generalized coefficient of linearity as*

$$\mu^{\{\gamma\}} := 1 - \frac{\left(a_1^{\{\gamma\}}\right)^2}{\sum_{j=0}^{\infty} \left(a_j^{\{\gamma\}}\right)^2}$$

Further let $\delta = 1 - |\rho(x,y)| := 1 - \frac{|x^\top y|}{\|x\|\|y\|} \in [0,1]$, then for any odd function σ, we have that

$$\frac{|\hat{\sigma}_c(x,y)|}{\sqrt{\hat{\sigma}_c(x,x)\hat{\sigma}_c(y,y)}} \leq \begin{cases} \left(\sqrt{1 - (1 + \frac{\mu^{\{\|x\|^2\}}}{2})\delta}\right) \cdot \left(\sqrt{1 - (1 + \frac{\mu^{\{\|y\|^2\}}}{2})\delta}\right) & \text{if } \delta \leq \frac{1}{2} \\ \sqrt{\left(1 - \frac{\mu^{\{\|x\|^2\}}}{2}\right)} \cdot \sqrt{\left(1 - \frac{\mu^{\{\|y\|^2\}}}{2}\right)(1 - \delta)} & \text{if } \delta > \frac{1}{2}. \end{cases}$$

A simple consequence of the above theorem which follows by noting that since $\mu^{\{\gamma\}} \in [0,1]$ for all γ, we have that for odd function σ, the normalized dot-products always decrease in absolute value, i.e.

$$\frac{|\hat{\sigma}_c(x,y)|}{\sqrt{\hat{\sigma}_c(x,x)\hat{\sigma}_c(y,y)}} \leq |\rho(x,y)|.$$

We end this section with the proofs of Lemma 42 and Lemma 41.

Proof [Proof of Lemma 42] Firstly note that since σ is an odd function it is easy to check that $a_0^{\{\gamma\}} = 0$ for all γ. Therefore we have the following consequence,

$$\frac{\hat{\sigma}_c(x,y)}{\sqrt{\hat{\sigma}_c(x,x) \cdot \hat{\sigma}_c(y,y)}} = \frac{\sum_{j=1}^{\infty} a_j^{\{\|x\|^2\}} a_j^{\{\|y\|^2\}} \rho(x,y)^j}{\sqrt{\sum_{j=1}^{\infty} \left(a_j^{\{\|x\|^2\}}\right)^2} \sqrt{\sum_{j=1}^{\infty} \left(a_j^{\{\|y\|^2\}}\right)^2}}$$

A simple application of Cauchy-Schwartz gives

$$\frac{\sum_{j=1}^{\infty} |a_j^{\{\|x\|^2\}}| |a_j^{\{\|y\|^2\}}| |\rho(x,y)|^j}{\sqrt{\sum_{j=1}^{\infty} \left(a_j^{\{\|x\|^2\}}\right)^2} \sqrt{\sum_{j=1}^{\infty} \left(a_j^{\{\|y\|^2\}}\right)^2}} \le \sqrt{\frac{\sum_{j=1}^{\infty} \left(a_j^{\{\|x\|^2\}}\right)^2 |\rho(x,y)|^j}{\sum_{j=1}^{\infty} \left(a_j^{\{\|x\|^2\}}\right)^2}} \sqrt{\frac{\sum_{j=1}^{\infty} \left(a_j^{\{\|y\|^2\}}\right)^2 |\rho(x,y)|^j}{\sum_{j=1}^{\infty} \left(a_j^{\{\|y\|^2\}}\right)^2}}$$

We will now analyse the left term first

$$\frac{\sum_{j=1}^{\infty} \left(a_j^{\{\|x\|^2\}}\right)^2 |\rho(x,y)|^j}{\sum_{j=1}^{\infty} \left(a_j^{\{\|x\|^2\}}\right)^2} \le (1 - \mu^{\{\|x\|^2\}}) |\rho(x,y)| + \mu^{\{\|x\|^2\}} \rho(x,y)^2 \tag{17}$$

$$= (1 - \mu^{\{\|x\|^2\}})(1 - \delta) + \mu^{\{\|x\|^2\}}(1 - \delta)^2 \tag{18}$$

$$= (1 - \delta)(1 - \mu^{\{\|x\|^2\}} \delta) \tag{19}$$

Now if $\delta > \frac{1}{2}$, we have $(1-\delta)(1-\mu^{\{\gamma\}}\delta) \le (1 - \frac{\mu^{\{\gamma\}}}{2})(1-\delta)$. If $\delta \le \frac{1}{2}$, we have $(1-\delta)(1-\mu^{\{\gamma\}}\delta) = 1 - (1 + \mu^{\{\gamma\}})\delta + \mu^{\{\gamma\}}\delta^2 \le 1 - (1 + \frac{\mu^{\{\gamma\}}}{2})\delta$. Therefore we have that

$$\frac{\sum_{j=1}^{\infty} \left(a_j^{\{\|x\|^2\}}\right)^2 |\rho(x,y)|^j}{\sum_{j=1}^{\infty} \left(a_j^{\{\|x\|^2\}}\right)^2} \le \begin{cases} 1 - (1 + \frac{\mu^{\{\gamma\}}}{2})\delta & \text{if } \delta \le \frac{1}{2} \\ (1 - \frac{\mu^{\{\gamma\}}}{2})(1 - \delta) & \text{if } \delta > \frac{1}{2}. \end{cases}$$

Repeating the same analysis for the second term and combining finishes the proof. ∎

Proof [Proof of Lemma 41] Lets consider the moment generating function for a gaussian with covariance Σ.

$$\mathop{\mathbb{E}}_{z \sim \mathcal{N}(0,\Sigma)} [exp(sz_1 + tz_2)]$$

We can reparametrize this distribution as follows - consider two vectors u, v such that $\|u\| = \gamma_1$, $\|v\| = \gamma_2$ and $u \cdot v = \gamma_3$. Now we see that

$$
\begin{aligned}
E_{z \sim \mathcal{N}(0,\Sigma)} [exp(sz_1 + tz_2)] &= E_{g \sim \mathcal{N}(0,I)} [exp(s(u_1 g_1 + u_2 g_2) + t(v_1 g_1 + v_2 g_2))] \\
&= E_{g \sim \mathcal{N}(0,1)} [exp((su_1 + tv_1)g)] \cdot E_{g \sim \mathcal{N}(0,1)} [exp((su_2 + tv_2)g)] \\
&= exp(0.5(su_1 + tv_1)^2) \cdot exp(0.5(su_2 + tv_2)^2) \\
&= exp(0.5(s^2 \|u\|^2 + t^2 \|v\|^2) + st(u \cdot v))
\end{aligned}
$$

Rearranging the above and replacing relevant quantities we get that

$$E_{z \sim \mathcal{N}(0,\Sigma)} [exp(sz_1 - 0.5\gamma_1^2 s^2) \cdot exp(tz_2 - 0.5\gamma_2^2 t^2)] = exp(st\gamma_3) = \sum_{j=0}^{\infty} \frac{1}{j!} s^j t^j \gamma_3^j$$

Working with the expression derived for LHS in (14) we get that

$$\sum_{i,j} \frac{1}{i!j!} E_{z \sim \mathcal{N}(0,\Sigma)} [H_i^{\gamma_1}(z_1) H_j^{\gamma_2}(z_2)] s^i t^j = \sum_{j=0}^{\infty} \frac{1}{j!} s^j t^j \gamma_3^j$$

Matching coefficient we get that

$$E_{z \sim \mathcal{N}(0,\Sigma)}[H_i^{\gamma_1}(z_1) H_j^{\gamma_2}(z_2)] = \begin{cases} j! \cdot \gamma_3^j & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Using the definition

$$h_j^\gamma(z) := \frac{H_j^{(\gamma)}(z)}{\sqrt{\gamma^j \cdot j!}}$$

now finishes the proof. ∎

D.3. Proof of Theorem 13

In this section we prove Theorem 13. We re-state the theorem in a more precise way as follows.

Theorem 43 *Let σ be a twice-differentiable monotonically increasing odd function which is concave on $\mathbb{R}^+$. There exists a constant α_σ (defined in Theorem 38) such that for any two inputs x, y such that $\|x\|^2, \|y\|^2 \geq 0.5$ with $\frac{x^\top y}{\|x\| \|y\|} \leq 1 - \delta$ for some $\delta > 0$, after a number of layers*

$$L \geq \hat{L} := \frac{1}{\alpha_\sigma} \log\left(\frac{4 \max(|\|x\|^2 - 1|, |\|y\|^2 - 1|, \mu/4)}{\mu} \right),$$

we have that

$$\frac{\phi_W(x)^\top \phi_W(y)}{\|\phi_W(x)\| \|\phi_W(y)\|} \leq B_{\mu/2}\left(L - \hat{L}, \delta\right)$$

Proof [Proof of Theorem 43] We begin the proof by first noting for any x such that $\|x\|^2 \in [1 - \mu/4, 1 + \mu/4]$ we have that

$$\mu^{\{\|x\|^2\}} \geq \mu/2 \tag{20}$$

where $\mu^{\{\|x\|^2\}}$ is as defined in Lemma 42 and $\mu = \mu^{\{1\}}$ is as defined in Definition 2. To see this first note that since $\mu \leq 1$, $1 - \mu/4 \geq 3/4$ and hence an application of Theorem 38 gives us that

$$\hat{\sigma}_l(\|x\|^2) = \hat{\sigma}_c(x, x) = \sum_{j=0}^{\infty} (a_j^{\{\|x\|^2\}})^2 \in [1 - \mu/4, 1 + \mu/4]$$

As a consequence since $\sum_{j=0}^{\infty} (a_j^{\{1\}})^2 = 1$ we also get that $a_1^{\{\|x\|^2\}} \leq 1 - \mu + \mu/4$. Therefore we have that

$$\mu^{\{\|x\|^2\}} := 1 - \frac{\left(a_1^{\{\|x\|^2\}}\right)^2}{\sum_{j=1}^{\infty} \left(a_j^{\{\|x\|^2\}}\right)^2} \geq 1 - \frac{1 - 3\mu/4}{1 - \mu/4} \geq \mu/2$$

The proof now follows by first noting from Theorem 13 that after $\hat{L}$ layers we have that the representations have norms smaller than $(1 - \mu/4, 1 + \mu/4)$. Formally for $l \geq \hat{l}$

$$\bar{k}^l(x, x), \bar{k}^l(y, y) \in [1 - \mu/4, 1 + \mu/4]$$

Now using the claim derived in (20), using Lemma 42 iteratively and using the definition of $B_\nu(L, \delta)$ immediately implies the theorem. ∎

D.4. Discussion

To get the results for inputs with general norms we introduce extra assumptions on the activation function. As noted earlier, the assumption of differentiability is only for convenience, the result will hold for any function with finite non-smooth points. The most restrictive assumption we require on the activation functions is the assumption of it being odd. This is a specification of our unbiased normalization assumption. We note that many activations like tanh, arctan, soft sign etc satisfy this assumption. Finding a more general condition for global convergence of inner products to 0 is an intriguing open question.

Furthermore our results on the norm convergence holds more generally than odd functions (See Theorem 38). It requires the activation to have a convex-concave structure and be 0 at 0. This assumption is far weaker and in fact most of the activations used in practice satisfy this (for eg. standard/leaky ReLU, SeLU, sigmoid, tanh, arctan etc.). Once again finding general conditions for global convergence of norm to fixed points is an intriguing open question. Poole et al. (2016) informally (but incorrectly) mention that just monotonicity of activation is a sufficient condition, however the standard ReLU with a little shift, i.e. $\max(x, 0) - 1$, itself presents a simple counter-example to the claim.

Appendix E. Optimization Proofs

Suppose we train the network using gradient descent on a loss function $\ell : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$, which defines the empirical loss function

$$\mathcal{L}(\vec{W}) := \frac{1}{n} \sum_{i=1}^{n} \ell(f_{\vec{W}}(x_i), y_i).$$

For the rest of this section we will assume that the loss function ℓ is the square loss, i.e. $\ell(\hat{y}, y) = (\hat{y} - y)^2$. The results presented can appropriately be extended to the setting where the loss function is smooth and strongly convex. Training a finite-width neural network necessitates the study of the conditioning of the finite-width kernel matrices K and $\mathcal{K}$, rather than their infinite-width counterparts. In such settings optimization results typically follow from a simple 2-step modular analysis:

- **Step 1. [Initial Stability]** Standard concentration inequalities imply that if the width is large enough, conditioning of the infinite-width kernel matrices transfers to their finite-width counterparts at initialization.

- **Step 2. [Training Stability]** Standard optimization theory implies that conditioning in finite-width kernel matrices leads to fast training. In the case of training only the top layer this is sufficient. When training all layers, a much more careful analysis is needed to show that the NTK stays "close" to initialization, leading to conditioning throughout the training process.

We now provide a couple of representative optimization results that follow from this type of analysis. Our goal here is to merely provide representative examples of typical optimization scenarios and highlight what benefits conditioning can lead to. Indeed, we believe extensions and improvements can be derived with significantly better bounds.

E.1. Training only the top layer

We consider a mode of training where only the top layer weight vector, v, is updated, while keeping $W_1, W_2, \ldots, W_L$ frozen at their randomly initialized values. To highlight this we introduce the

notation $\vec{W}_{1:L} = \{W_1 \dots W_L\}$. Let $\eta > 0$ be a step size, the update rule at iteration t is given by

$$v_{t+1} = v_t - \eta \cdot \partial_v \mathcal{L}(\{v_t, \vec{W}_{1:L}\}) = v_t - \eta \cdot \frac{1}{n} \sum_{i=1}^{n} 2(v_t \cdot \Phi_{\vec{W}}(x_i) - y_i)\Phi_{\vec{W}}(x_i).$$

Note that in this mode of training, the associated optimization problem is convex in v. To implement Step 1 of the modular analysis, we appeal to the results of Daniely et al. (2016). They show that when the activations are suitably bounded (see Definition 6 in their paper for C-bounded activations) and the width is large enough, then with high probability, each entry in the kernel matrix K is close to the corresponding entry in $\bar{K}$. Specifically, via Theorems 2 and 3 in their paper, we have the following version of Theorem 3 for finite width neural networks:

Lemma 44 (Via Theorem 2 in Daniely et al. (2016)) *For any $\gamma > 0$, suppose that the activation σ is C-bounded and $m = \Omega\left(\frac{(4C)^{L+1}\log(n)}{\gamma^2}\right)$, then with high probability, we have that for all i, j, $|K_{ij} - \bar{K}_{ij}| \le \gamma$.*

Step 2 follows by using standard convex optimization theory (Nesterov, 2014), which tells us that the convergence rate of gradient descent for this problem depends on the condition number of K. Specifically, we have the following result:

Theorem 45 *Suppose $L = \Theta\left(\frac{\log(n/\delta)}{\mu}\right)$. If σ is C-bounded and the width $m = poly(n, \frac{1}{\delta})$, then setting $\eta = \Theta\left(\frac{1}{\lambda_{\max}(\bar{K})}\right)$, we get that with high probability over the initialization,*

$$\mathcal{L}(\{v_t, \vec{W}_{1:L}\}) \le e^{-\Omega(t)} \cdot \mathcal{L}(\{v_0, \vec{W}_{1:L}\})$$

Alternatively, in order to find a point that is ϵ sub-optimal, gradient descent needs $O(\log(\frac{1}{\epsilon}))$ steps.

Similarly, one can also derive a linear convergence theorem for stochastic gradient descent:

Theorem 46 *With the same choice of parameters as in Theorem 45, appropriate choice of η and with high probability over the initialization, stochastic gradient descent finds a point that is ϵ-sub-optimal in expectation in at most $O\left(\log(\frac{1}{\epsilon})\right)$ steps.*

Remark 47 *The rate in the exponent in the theorem above naturally depends upon the condition number of the kernel matrix K. For simplicity, we choose to state the theorem for a depth at which the condition number is $O(1)$. Precise rates depending on L, can be derived from Corollary 4.*

E.2. Training All The Layers Together

In this section we provide a representative result for the training dynamics when all the layers are trained together with a fixed common learning rate. The dynamics are given by

$$\vec{W}(t+1) = \vec{W}(t) - \eta \partial_{\vec{W}} \mathcal{L}(\vec{W}(t))$$

Now since the bottom layers also move the kernel changes at every step. The standard analysis in this setting follows from carefully establishing that the NTK does not change too much during the training procedure allowing for the rest of the analysis to go through. The following theorem from Lee et al. (2019) summarizes one such setting for smooth activation functions.

Theorem 48 (Thereom G.4 in Lee et al. (2019)) *Suppose that the activation σ and its derivative σ' further satisfies the properties that there exists a constant c, such that for all x, x'*

$$|\sigma(x)|, |\sigma'(x)|, \frac{|\sigma'(x) - \sigma'(x')|}{|x - x'|} \le c.$$

Then there exists a constant N (depending on L, n, δ) such that for width $m > N$ and setting the learning rate $\eta = 2(\lambda_{\min}(\bar{K}) + \lambda_{\max}(\bar{K}))^{-1}$, with high probability over the initialization the following is satisfied for gradient descent for all t,

$$\mathcal{L}(\vec{W}(t)) \le e^{-\Omega\left(\frac{t}{\kappa(\bar{K})}\right)} \mathcal{L}(\vec{W}(0))$$

The following corollary is now a simple application of the above theorem and Corollary 6.

Corollary 49 *Suppose the conditions in Theorem 48 are satisfied and the width is taken to be a large enough constant (depending on L, n, δ) and further $L = \Theta(\frac{\log(n/\delta)}{\mu})$, then gradient descent with high probability finds an ϵ suboptimal point in total time $O(\log(1/\epsilon))$.*

Remark 50 *As stated in Theorem 48 the width required could be a very large constant. However, note that we require the depth to be logarithmic in $\frac{1}{\delta}$ for achieving constant condition number. Therefore the exponential in L factors accrued in the analysis of Theorem 48 are actually polynomial in $\frac{1}{\delta}$. Therefore, merging results from Arora et al. (2019c), we can derive a polynomial in $\frac{1}{\delta}$ upper bound on the width of the network. This matches the best known bounds on the overparameterization while improving the optimization rates exponentially (in $\frac{1}{\delta}$). Further we believe similar results can also be derived for ReLU activations following techniques in Allen-Zhu et al. (2018).*

The proofs for the theorems in this section follow easily from our established results and standard arguments from optimization theory. We provide them next for completeness.

E.3. Proofs

We begin by proving simple well-known theorems regarding gradient descent and stochastic gradient descent for linear regression. Consider the following problem

$$\mathcal{L}(w) = \frac{1}{n} \sum_{i=1}^{n} \|a_i^\top w - y\|^2$$

Let A be the matrix whose rows are a_i. We will assume that $\lambda_{\min}(A^\top A) > 0$, which in particular implies that $\min_w \mathcal{L}(w) = 0$. Lets first consider gradient descent, i.e.

$$w_{t+1} = w_t - \eta \nabla \mathcal{L}(w_t)$$

We have the following well known guarantee (Nesterov, 2014).

Theorem 51 *For gradient descent we have that*

$$\mathcal{L}(w_t) \le e^{-\frac{t}{4\kappa(A^\top A)}} \mathcal{L}(w_0)$$

Next we consider the stochastic gradient descent algorithm,

$$w_{t+1} = w_t - \eta \tilde{\nabla} \mathcal{L}(w_t),$$

where $\tilde{\nabla} \mathcal{L}(w_t)$ is a gradient over a single a_i, which is uniformly randomly sampled. Since we have assumed that the loss is 0, even SGD is known to have linear convergence in this setting. Since we did not find a concise proof of this fact anywhere we include it here for completeness.

Theorem 52 *Let $\|a_i\|^2 \leq \beta$ for all i, then stochastic gradient descent produces an $\epsilon-$sub-optimal point in expectation in total number steps bounded by*

$$\frac{8n\beta}{\lambda_{\min}(A^\top A)} \cdot \log(1/\epsilon))$$

Proof

$$
\begin{aligned}
E_t[\|w_{t+1} - w^*\|^2] &\leq E_t[\|w_t - w^*\|^2] - \eta(w_t - w^*)^\top E_t[\tilde{\nabla} \mathcal{L}(w_t)] + \eta^2 E_t[\|\tilde{\nabla} \mathcal{L}(w_t)\|^2] \\
&\leq \|w_t - w^*\|^2 - \eta(w_t - w^*)^\top \nabla \mathcal{L}(w_t) + \eta^2 \|\tilde{\nabla} \mathcal{L}(w_t)\|^2 \\
&\leq \|w_t - w^*\|^2 - \eta \mathcal{L}(w_t) + \eta^2 E_i[(a_i^\top w - y_i)^2 \|a_i\|^2] \\
&\leq \|w_t - w^*\|^2 - \eta \mathcal{L}(w_t) + \eta^2 \beta \mathcal{L}(w_t)
\end{aligned}
$$

where E_t refers to expectation conditioned on all the randomness till step t. Rearranging the above we get,

$$\mathcal{L}(w_t)(\eta - \eta^2 \beta) \leq \|w_t - w^*\|^2 - E_t[\|w_{t+1} - w^*\|^2]$$

Summing the above over time T gives us that

$$E\left[\mathcal{L}\left(\frac{1}{T} \cdot \sum_{t=1}^T w_t\right)\right] \leq E\left[\frac{1}{T} \cdot \sum_{t=1}^T \mathcal{L}(w_t)\right] \leq \frac{E[\|w_1 - w^*\|^2]}{T(\eta - \eta^2 \beta)} \leq \frac{n \cdot E[\mathcal{L}(w_1)]}{\lambda_{\min}(A^\top A)T(\eta - \eta^2 \beta)}$$

Setting η to $1/(2\beta)$ and $T = 8n\beta/\lambda_{\min}(A^\top A)$ we get that

$$E\left[\mathcal{L}\left(\frac{1}{T} \cdot \sum_{t=1}^T w_t\right)\right] \leq \frac{E[\mathcal{L}(w_1)]}{2}$$

Repeating this process a total of $\log(1/\epsilon)$ times gives us that after $\frac{8n\beta}{\lambda_{\min}(A^\top A)} \log(1/\epsilon)$ steps, SGD produces a point which is ϵ-sub-optimal point in expectation. ∎

Proof [Proof of Theorem 45] The statement follows by noticing that at that setting of depth, Corollary 4 implies that the infinite-width kernel has constant condition number. Now invoking Lemma 44 implies that the finite-width kernel also has a constant condition number. The statement then follows from Theorem 51. ∎

Proof [Proof of Theorem 46] We wish to invoke Theorem 52. To this end note that, using Lemma 44 and the fact that the diagonal entries are 1 in $\bar{K}$, we get that $\beta \leq 2$ w.h.p. Similarly using Corollary 4, we can derive that $\lambda_{\min} = \Omega(n)$ w.h.p. Therefore using Theorem 52 we get the required result. ∎

Proof [Proof of Corollary 49] The theorem follows by noticing that Corollary 6 implies that at that depth, the condition number of the infinite-width NTK is constant. The statement now follows from Theorem 48. ∎

Appendix F. Proofs on SQ Learnability of Random Deep Neural Nets

In this section we prove Theorem 16 regarding SQ learnability of randomly initialized deep neural networks. Specifically, we prove our result under the assumption that the (normalized) activation σ is subgaussian with constant subgaussian norm. In particular we assume that

$$\mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,1)}\left[e^{\lambda \sigma(X)}\right] \le e^{\lambda^2 \alpha^2 / 2}, \tag{21}$$

for a constant $\alpha > 0$. Many activations such as the sign, ReLU and tanh satisfy this assumption.

A key component in establishing SQ hardness of learning is to show that given two non-collinear unit length vectors, a randomly initialized network of depth h and sufficiently large width width makes, in expectation, the pair nearly orthogonal. In other words, the magnitude of the expected dot product between any pair decreases exponentially with depth. While Das et al. (2019) proved the result for sign activations, we prove the statement for more general activations and then use it to establish SQ hardness of learning. As mentioned in Section 5, we will work with networks that will normalize the output of each layer to unit length via the operation $\Pi : \mathbb{R}^m \to \mathbb{R}^m$. Then we have the following theorem:

Theorem 53 *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a non linear activation with μ being the coefficient of non-linearity as in Definition 2 and satisfying (21). Let $x_i, x_j \in \mathbb{R}^d$ be unit length vectors such that $|x_i \cdot x_j| \le 1 - \delta$. Define $\Phi_{\tilde{W}}(x) \coloneqq \frac{1}{\sqrt{m}} \Pi(\sigma(W_L \frac{1}{\sqrt{m}} \Pi(\sigma(W_{L-1} \cdots \frac{1}{\sqrt{m}} \Pi(\sigma(W_1 x) \cdots)))$, where each column of W_1 is sampled from $\mathcal{N}(0, I_{d \times d})$ and each column of W_i is sampled from $\mathcal{N}(0, I_{m \times m})$ for $i > 1$. Furthermore, the operation Π normalizes the output of each layer to unit length. Let $m > c_1 \frac{L}{\mu^2 \delta^2}$ for a universal constant $c_1 > 0$ and for $h \in [1, L]$ define ρ_h be the dot product obtained by taking the representation of x_i, x_j at depth h of the network defined above. Then for any $h > 1$, it holds that*

$$\left| \mathbb{E}\left[\rho_{L_0(\delta) + h}\right] \right| \le e^{-\Omega(h)} + L e^{-\Omega(L)}.$$

where $L_0(\delta) = c_1 \frac{\log(\frac{1}{\delta})}{\mu}$ and $c_1 > 0$ is a universal constant.

While the above theorem is not a black blox application of our main result (Theorem 3) since careful concentration arguments are required due to finite width, the calculations are of a similar flavor.

We now show how the above theorem can be used to generalize the SQ lower bound of Das et al. (2019). Before describing our results, we recall that in the SQ model (Kearns, 1998) the learning algorithm does not have access to a labeled training set. Instead, for a given target function f and a distribution D over $\mathbb{R}^d$, the algorithm has access to a query oracle $SQ_{f,D}(\psi, \tau)$. The oracle takes as input a query function ψ, and outputs a value v such that $|\mathbb{E}_D[\psi(x, f(x))] - v| \le \tau$. The goal of the algorithm is to use the query algorithm to output a function g that ϵ approximates f, i.e., $Pr_D[g(x)f(x)] \ge \epsilon$, for a given $\epsilon > 0$.

(Das et al., 2019) established an SQ learnability lower bound for a subclass $\mathcal{F}$ of neural networks with the property that a randomly initialized neural network falls in $\mathcal{F}$ with high probability. This however only establishes that the class $\mathcal{F}$ is hard to SQ learn as opposed to showing that a randomly initialized neural network is hard to learn. Furthermore, the lower bound only applies to networks with sign activations. We now show how to generalize their result in two ways: (a) we allow arbitrary activations satisfying (21), and (b) our lower bound shows that a randomly initialized network is hard

to learn in the SQ model with constant probability. We achieve the stronger lower bound by carefully adapting the lower bound technique of Bshouty and Feldman (2002).

In our context we will fix a non linear activation $\sigma : \mathbb{R} \to \mathbb{R}$ and let the target be of the form $\text{sgn}(f_{\vec{W}}(x))$ where,

$$f_{\vec{W}}(x) = \left(v \cdot \frac{1}{\sqrt{m}}\Pi(\sigma(W_L \frac{1}{\sqrt{m}}\Pi(\sigma(W_{L-1}\cdots\frac{1}{\sqrt{m}}\Pi(\sigma(W_1 x)\cdots)))))\right)$$

where each column of W_1 is sampled from $\mathcal{N}(0, I_{d \times d})$ and v and each column of W_i is sampled from $\mathcal{N}(0, I_{m \times m})$ for $i > 1$. Furthermore we will use the depth L and the dimensionality d to parameterize the bit complexity of the network description. We say that an algorithm $(p(d, L), r(d, L), q(d, L))$-SQ learns $\text{sgn}(f_{\vec{W}}(x))$ if with probability at least $1/2$ over the randomness in $\vec{W}$, the algorithm makes at most $p(d, L)$ queries to the SQ oracle for $\text{sgn}(f_{\vec{W}}(x))$, receives responses from the oracle up to tolerance $\tau = 1/r(d, L)$ and outputs a g that $\epsilon = 1/q(d, L)$-approximates f. Furthermore it is the case that each query function ψ used by the algorithm can be evaluated in time $q(d, L)$.

Then we have the following theorem, that is a more formal restatement of Theorem 16 and extends the result of Das et al. (2019).

Theorem 54 *Fix any non linear activation σ with the coefficient of non-linearity being μ that satisfies (21). Any algorithm that $(p(d, L), poly(d, L), poly(d, L))$-SQ learns the random depth L networks as defined above with width $m = \Omega(L\frac{\mu^2}{\delta^2})$ must satisfy $p(d, L) \geq e^{\Omega(L)}$.*

Proof [Proof of Theorem 53] We use the following notation in the proof. Given input x_i, we denote $x_i^{(h)}$ to be the representation obtained at depth h of the network and $\hat{x}_i^{(h)}$ to be the corresponding normalized input. Recall that we are normalizing the output of each layer to be unit length. Similarly, given x_i, x_j, we denote by $\hat{\rho}_h = \hat{x}_i^{(h)} \cdot \hat{x}_j^{(h)}$ and $\rho_h = x_i^{(h)} \cdot x_j^{(h)}$. Next we have that conditioned on $x_i^{(h-1)}$ and $x_j^{(h-1)}$,

$$\|x_i^{(h)}\|^2 = \frac{1}{m}\sum_{j=1}^{m} \sigma^2(w_j \cdot \hat{x}^{(h-1)})$$

where $w_j \sim N(0, I)$ and $\mathbb{E}[\|x_i^{(h)}\|^2] = 1$. Furthermore, since $\sigma(w_j \cdot \hat{x}^{(h-1)})$ is a subgaussian random variable with constant subgaussian norm, $\|x_i^{(h)}\|^2$ is a sum of subexpoential random variables. By Bernstein's inequality for subexponential random variables (Vershynin, 2018) we have that for a universal constant $c > 0$,

$$Pr(\left|\|x_i^{(h)}\|^2 - 1\right| > t) \leq 2e^{-c\min\left(mt^2, mt\right)}. \tag{22}$$

Similarly we have that

$$\rho_h = \frac{1}{m}\sum_{j=1}^{m} \sigma(w_j \cdot \hat{x_i}^{(h-1)})\sigma(w_j \cdot \hat{x_j}^{(h-1)})$$

with $\mathbb{E}[\rho_h] = \hat{\sigma}(\hat{\rho}_{h-1})$. Noting that product of subgaussian random variables is subexponential and again applying Bernstein's inequality for subexponential random variables we get that

$$Pr(\left|\rho_h - \hat{\sigma}(\rho_{h-1})\right| > t) \leq 2e^{-c\min\left(mt^2, mt\right)}. \tag{23}$$

Next, we will use (22) and (23) to argue that with high probability $\hat{\rho}_h$ remains close to $\hat{\sigma}(\hat{\rho}_{h-1})$. For suitable constant $\epsilon < 1$ to be chosen later, we have that

$$
\begin{aligned}
Pr(|\hat{\rho}_h - \hat{\sigma}(\hat{\rho}_{h-1})| > t) &= Pr\left(\left|\frac{\rho_h}{\|x_i^{(h)}\|\|x_j^{(h)}\|} - \hat{\sigma}(\hat{\rho}_{h-1})\right| > t\right) \\
&\leq Pr\left(\left|\frac{\rho_h - \hat{\sigma}(\hat{\rho}_{h-1})}{\|x_i^{(h)}\|\|x_j^{(h)}\|}\right| > t\right) + Pr\left(\left|\hat{\sigma}(\hat{\rho}_{h-1})\left(\frac{1}{\|x_i^{(h)}\|\|x_j^{(h)}\|} - 1\right)\right| > t\right) \\
&\leq Pr(\|x_i^{(h)}\|\|x_j^{(h)}\| > (1+\epsilon)^2) + Pr\left(\left|\rho_h - \hat{\sigma}(\hat{\rho}_{h-1})\right| > t(1+\epsilon)^2\right) \\
&\quad + Pr\left(\left|\|x_i^{(h)}\|\|x_j^{(h)}\| - 1\right| > \frac{2t}{\hat{\sigma}(\hat{\rho}_{h-1})}\right).
\end{aligned}
$$

Noticing that $\hat{\sigma}(\hat{\rho}_{h-1}) \leq 1 - (1 + \frac{\mu}{2})\delta$, and using (22) and (23), we get that

$$
Pr(|\hat{\rho}_h - \hat{\sigma}(\hat{\rho}_{h-1})| > t) \leq 2\left(e^{-cm\epsilon^2} + e^{-cm(1+\epsilon)^4 t^2} + e^{-\frac{mt}{2(1-(1+\mu/2)\delta)}}\right).
$$

Setting $t = \delta\mu/4$ and ϵ to be a small enough constant we get that

$$
Pr\left(|\hat{\rho}_h - \hat{\sigma}(\hat{\rho}_{h-1})| > \frac{\delta\mu}{4}\right) \leq 2\left(e^{-\Omega(m)} + e^{-\Omega(m\mu^2\delta^2)} + e^{-\Omega(m\delta\mu)}\right).
$$

Setting $m \geq c_1 \frac{L}{\mu^2\delta^2}$ and using a union bound over all layers we get that with probability at least $1 - Le^{-\Omega(L)}$, the updates of $\hat{\rho}_h$ will approximately satisfy the ideal updates from Theorem 3 and as a result, for a constant $c_1 > 0$, after $L_0(\delta) = c_1\frac{\log(1/\delta)}{\mu}$ depth, with high probability, $\hat{\rho}_h$ (and ρ_h) will fall below $1/4$ and will continue to be below $1/2$ for all L. Define G to be the intersection of above good event and that $\rho_h \in [(1-\epsilon)\hat{\rho}_h, (1+\epsilon)\hat{\rho}_h]$ for all $h \in [L]$. Then we know that $P(G) \geq 1 - 2Le^{-\Omega(L)}$. Conditioned on this good event and using Lemma 22 we have that for $h > L_0(\delta)$,

$$
\begin{aligned}
\left|\mathbb{E}[\rho_h|G, \rho_{h-1}]\right| &= \left|\hat{\sigma}(\hat{\rho}_{h-1})\right| \\
&\leq (1 - \frac{\mu}{2})|\hat{\rho}_{h-1}| \\
&\leq (1+\epsilon)(1 - \frac{\mu}{2})|\rho_{h-1}| \\
&\leq (1 - \frac{\mu}{4})|\rho_{h-1}|
\end{aligned}
$$

for a small enough constant ϵ. Hence we get that for $h > L_0(\delta)$, $|\mathbb{E}[\rho_h|G]| \leq e^{-\Omega(h)}$. Finally notice that

$$
\mathbb{E}[\rho_h] = P(G)\,\mathbb{E}[\rho_h|G] + P(\bar{G})\,\mathbb{E}[\rho_h|\bar{G}]
$$

Combined with the probability of the good event and noticing that $\hat{\sigma}(\hat{\rho}_h)$ is always bounded, we get that

$$
\left|\mathbb{E}[\rho_{L_0(\delta)+h}]\right| \leq e^{-\Omega(h)} + Le^{-\Omega(L)}.
$$

∎

Proof [Proof of Theorem 16 (same as Theorem 54)] We will consider a randomly initialized deep neural network defined as

$$
f_{\bar{W}}(x) = v \cdot \Pi(\sigma(W_L\frac{1}{\sqrt{m}}\Pi(\sigma(W_{L-1}\cdots\frac{1}{\sqrt{m}}\Pi(\sigma(W_1 x)\cdots)))))
$$

45

We consider a distribution D that is the uniform distribution over a set $\mathcal{S}$ that consists of half of the inputs in $\{+1, -1\}^d$ thereby ensuring that no two inputs are collinear. In particular, one can take the set of all 2^{d-1} inputs that fall on one side of a fixed halfspace. The first step in the analysis is to show that $f = \text{sgn}(f_{\vec{W}})$ is uncorrelated with any fixed function $g : \mathbb{R}^d \to [-1, 1]$. In particular we have that

$$
\begin{aligned}
\mathbb{E}_{\vec{W}}\big[\mathbb{E}_x[g(x)f(x)]^2\big] &= \frac{1}{|S|^2}\Big(\sum_x \mathbb{E}_{\vec{W}}[g^2(x)f_{\vec{W}}^2(x))] + 2\sum_{x \neq y}\mathbb{E}_{\vec{W}}[g(x)g(y)\text{sgn}(f_{\vec{W}}(x))\text{sgn}(f_{\vec{W}}(y))]\Big) \\
&= \frac{1}{|S|^2}\Big(\sum_x \mathbb{E}_{\vec{W}}[g^2(x)] + 2\sum_{x \neq y}g(x)g(y)\,\mathbb{E}_{\vec{W}}[\text{sgn}(f_{\vec{W}}(x))\text{sgn}(f_{\vec{W}}(y))]\Big) \\
&\leq \frac{1}{|S|^2}\Big(\sum_x \mathbb{E}_{\vec{W}}[g^2(x)] + 2\sum_{x \neq y}|g(x)||g(y)|\,\mathbb{E}_{\vec{W}}[|\Phi_{\vec{W}}(x)\cdot\Phi_{\vec{W}}(y)|]\Big)
\end{aligned}
$$

Next, Theorem 53 implies that

$$
\mathbb{E}_{\vec{W}}|\Phi_{\vec{W}}(x)\cdot\Phi_{\vec{W}}(y)| \leq Le^{-\Omega(L)},
$$

Substituting above and noticing that $g(x) \in [-1, 1]$ we have

$$
\begin{aligned}
\mathbb{E}_{\vec{W}}\big[\mathbb{E}_x[g(x)f(x)]^2\big] &\leq \frac{1}{|S|^2}\Big(\sum_x 1 + 2\sum_{x \neq y}Le^{-\Omega(L)}\Big). \\
&\leq Le^{-\Omega(L)}.
\end{aligned}
\tag{24}
$$

Next assume that there exists an algorithm $\mathcal{A}$ that makes $p(d, L)$ queries of tolerance $r(d, L)$ to an SQ oracle for a random function $\text{sgn}(f_{\vec{W}}(x))$ as defined above, and with probability at least half (over the randomness of the algorithm and the random draw of the function), outputs a function g such that

$$
\mathbb{E}_D[\text{sgn}(f_{\vec{W}}(x))h(x)] \geq \frac{1}{q(d, L)}.
$$

Here we assume that both $r(d, L)$ and $q(d, L)$ are polynomial in d and L. To get a contradiction we will use the technique from the work of Bshouty and Feldman (2002) (see Theorem 31). As a first step, since we are in the case of learning with respect to a fixed distribution, from the work of Bshouty and Feldman (2002) it follows that we can, without loss of generality, assume that the statistical queries $\psi(x, \text{sgn}(f_{\vec{W}}(x)))$ used by the algorithm are correlation queries, i.e. $\psi(x, f_{\vec{W}}(x)) = \mathbb{E}_D[\text{sgn}(f_{\vec{W}}(x))g(x)]$. Next we simulate the algorithm $\mathcal{A}$ and each time the algorithm makes a statistical query $\mathbb{E}_D[\text{sgn}(f_{\vec{W}}(x))g_i(x)]$, we add g_i to a set $\mathcal{H}$. Finally, if the algorithm outputs a hypothesis h at the end, we add h to $\mathcal{H}$ as well. Notice that if $\mathcal{A}$ makes $p(d, L)$ queries then $|\mathcal{H}| \leq p(d, L) + 1$. Next from (24) and a union bound over $\mathcal{H}$ we can say that

$$
\Pr_{\vec{W}}\left[\exists h \in \mathcal{H} : \mathbb{E}_D[(\text{sgn}(f_{\vec{W}}(x))h(x))]^2 > \frac{1}{\max(q^2(d, L), r^2(d, L))}\right] \leq
$$
$$
O\Big(\max(q^2(d, L), r^2(d, L))(p(d, L) + 1)e^{-\Omega(L)}\Big).
$$

Since the correlation of each function in $\mathcal{H}$ with $\text{sgn}(f_{\vec{W}}(x))$ is at most $1/r(d, L)$, a zero answer to every query asked by the algorithm is a valid output of the SQ oracle, and hence with probability at

46

least $1/2$, the algorithm must output a function in $\mathcal{H}$ that is correlated with $\operatorname{sgn}(f_{\vec{W}}(x))$. In other words, we have that

$$\Pr_{\vec{W}}\left[\exists h \in \mathcal{H} : \mathbb{E}_{D}[(\operatorname{sgn}(f_{\vec{W}}(x))h(x))]^2 > \frac{1}{\max(q^2(d,L), r^2(d,L))}\right] \geq \frac{1}{2}.$$

From the above we get that

$$\max(q^2(d,L), r^2(d,L))(p(d,L)+1) \geq e^{\Omega(L)}.$$

∎

Appendix G. Interpolation analysis

We need a bit more notation for the analysis. We denote by the infinite matrix X the linear map from $\mathcal{H} \to \mathbb{R}^n$ corresponding to the inputs $x_1, x_2, \ldots, x_n$, so that for any $v \in \mathcal{H}$, $Xv \in \mathbb{R}^n$ has ith component $v^\top \Phi(x_i)$. Note that $XX^\top = \bar{K}$, the kernel matrix for the training data defined by $\bar{k}$. We denote by $\mathbf{y}$ the vector $\langle y_1, y_2, \ldots, y_n \rangle^\top \in \mathbb{R}^n$, and by $\Sigma = \mathbb{E}_{(x,y)}[\Phi(x)\Phi(x)^\top]$ the data covariance matrix. If $\bar{K}$ is non-singular, then the linear predictor $v_S = X^\top \bar{K}^{-1}\mathbf{y}$ interpolates on S, i.e., $v_S^\top \Phi(x_i) = y_i$ for all $i \in [n]$, and indeed, is the *minimum norm* interpolating linear predictor.

The first step in the analysis is the following bound on $\|\Sigma\|$:

Lemma 55 *For any positive integer N and $\gamma \in (0, 1/2)$, if the number of hidden layers $L \geq \lceil \frac{\log(N)}{-\log(1-\frac{\mu}{2})}\rceil + L_0(\Delta(N,\gamma))$, then $\|\Sigma\| \leq \frac{8\ln(N)}{N}$.*

Proof Let $T = \{x_1', x_2', \ldots, x_N'\}$ be a sample set of size N drawn i.i.d. from the marginal distribution of $\mathcal{D}$ over the x-coordinate. Let X' denote by the infinite matrix corresponding to the linear map from $\mathcal{H} \to \mathbb{R}^N$ such that for any $v \in \mathcal{H}$, $X'v \in \mathbb{R}^m$ has ith component $v^\top \Phi(x_i')$. Let $X'X'^\top = \bar{K}'$, the top layer kernel matrix for the training data defined by $\bar{k}$.

Then by the definition of Δ, with probability at least $1 - \gamma$, Assumption A holds for T with $\delta = \Delta(N,\gamma)$. Conditioned on this Assumption A holding, Theorem 3 (part 1) implies that $\|\bar{K}'\| \leq 2$ since $L \geq \lceil \frac{\log(N)}{-\log(1-\frac{\mu}{2})}\rceil + L_0(\Delta(N,\gamma))$. Thus, $\|X'^\top X'\| = \|X'X'^\top\| \leq 2$, which implies that $\Pr_T[\|X'^\top X'\| \leq 2] \geq 1 - \gamma$.

Note that for any $x \in \mathbb{S}^{d-1}$, we have $\|\Phi(x)\|^2 = \bar{k}(x,x) = \hat{\sigma}^{(L)}(1) = 1$. Thus $\|\Phi(x)\Phi(x)^\top\| = 1$, $\operatorname{Tr}(\Sigma) \leq 1$, $\operatorname{intdim}(\Sigma) := \frac{\operatorname{Tr}(\Sigma)}{\|\Sigma\|} \leq \frac{1}{\|\Sigma\|}$, and $(\Phi(x)\Phi(x)^\top)^2 = \Phi(x)\Phi(x)^\top$. Thus, Theorem 7.7.1 in (Tropp, 2015) and some simple calculations imply that for $\ell := \ln(8/\|\Sigma\|)$, we have

$$\Pr_T\left[\|\tfrac{1}{N}X'^\top X' - \Sigma\| > \tfrac{2\ell}{3N} + \sqrt{\tfrac{2\ell\|\Sigma\|}{N}}\right] \leq \tfrac{1}{2}.$$

By a union bound, we have

$$\Pr_T\left[\|X'^\top X'\| \leq 2 \text{ and } \|\tfrac{1}{N}X'^\top X' - \Sigma\| \leq \tfrac{2\ell}{3N} + \sqrt{\tfrac{2\ell\|\Sigma\|}{N}}\right] \geq \tfrac{1}{2} - \gamma > 0.$$

This implies that $\|\Sigma\| \leq \frac{2\ell}{N} + \sqrt{\frac{2\ell\|\Sigma\|}{N}} \leq \frac{2\ell}{N} + \frac{2\ell/N+\|\Sigma\|}{2} \implies \|\Sigma\| \leq \frac{6\ell}{N} \implies \|\Sigma\| \leq \frac{8\ln(N)}{N}$ using the fact that $\ell = \ln(8/\|\Sigma\|)$. ∎

The following lemma is a standard calculation that is a slight generalization of a similar statement[5] in (Bartlett et al., 2019b):

Lemma 56 *The excess risk of the minimum norm estimator satisfies*

$$\mathop{\mathbb{E}}_{(x,y)}\big[(y - v_S^\top \Phi(x))^2\big] - \mathop{\mathbb{E}}_{(x,y)}\big[(y - v^{*\top}\Phi(x))^2\big] \le 2v^{*\top}Bv^* + 2\epsilon^\top C\epsilon,$$

where $\epsilon = \mathbf{y} - Xv^*$, $B = (I - X^\top \bar{K}^{-1}X)\Sigma(I - X^\top \bar{K}^{-1}X)$, and $C = \bar{K}^{-1}X\Sigma X^\top \bar{K}^{-1}$.

Proof Since v^* is a minimizer of $\mathbb{E}_{(x,y)}[(y - v^\top\Phi(x))^2]$, we have $\nabla_v \mathbb{E}_{(x,y)}[(y - v^{*\top}\Phi(x))^2] = 0$, which implies that $\mathbb{E}_{(x,y)}[(y - v^\top\Phi(x))\Phi(x)] = 0$. Using this fact, we have

$$\mathop{\mathbb{E}}_{(x,y)}\big[(y - v_S^\top\Phi(x))^2\big] = \mathop{\mathbb{E}}_{(x,y)}\big[(y - v^{*\top}\Phi(x) + (v_S - v^*)^\top\Phi(x))^2\big]$$

$$= \mathop{\mathbb{E}}_{(x,y)}\big[(y - v^{*\top}\Phi(x))^2\big] + 2(v_S - v^*)^\top \mathop{\mathbb{E}}_{(x,y)}\big[(y - v^{*\top}\Phi(x))\Phi(x)\big]$$

$$+ \mathop{\mathbb{E}}_{(x,y)}\big[((v_S - v^*)^\top\Phi(x))^2\big]$$

$$= \mathop{\mathbb{E}}_{(x,y)}\big[(y - v^{*\top}\Phi(x))^2\big] + \mathop{\mathbb{E}}_{(x,y)}\big[((v_S - v^*)^\top\Phi(x))^2\big].$$

Using this fact, and that $v_S = X^\top \bar{K}^{-1}\mathbf{y} = X^\top \bar{K}^{-1}(Xv^* + \epsilon)$, we get that the excess risk equals

$$\mathop{\mathbb{E}}_{(x,y)}\big[((v_S - v^*)^\top\Phi(x))^2\big] = \mathop{\mathbb{E}}_{(x,y)}\big[((X^\top \bar{K}^{-1}(Xv^* + \epsilon) - v^*)^\top\Phi(x))^2\big]$$

$$= \mathop{\mathbb{E}}_{(x,y)}\big[((X^\top \bar{K}^{-1}X - I)v^* + X^\top \bar{K}^{-1}\epsilon)^\top\Phi(x))^2\big]$$

$$\le 2\mathop{\mathbb{E}}_{(x,y)}\big[((X^\top \bar{K}^{-1}X - I)v^*)^\top\Phi(x))^2\big] + 2\mathop{\mathbb{E}}_{(x,y)}\big[((X^\top \bar{K}^{-1}\epsilon)^\top\Phi(x))^2\big]$$

$$= 2v^{*\top}Bv^* + 2\epsilon^\top C\epsilon.$$

The last equality uses the fact that for any $v \in \mathcal{H}$, we have

$$\mathop{\mathbb{E}}_{(x,y)}\big[(v^\top\Phi(x))^2\big] = \mathop{\mathbb{E}}_{(x,y)}\big[v^\top\Phi(x)\Phi(x)^\top v\big] = v^\top \Sigma v.$$

∎

We can now prove Theorem 17:

Proof [(Theorem 17)] First, as in the proof of Lemma 55, by the definition of Δ, with probability at least $1 - \gamma$, Assumption A holds for S with $\delta = \Delta(n^2, \gamma)$. Conditioned on this Assumption A holding, Theorem 3 (part 1) implies that $\lambda_{\min}(\bar{K}) \ge 1/2$ since $L = \lceil\frac{\log(n^2)}{-\log(1-\frac{\mu}{2})}\rceil + L_0(\Delta(n^2,\gamma))$. Thus, with probability at least $1 - \gamma$ over the choice of S, $\bar{K}$ is non-singular, and hence there exists an interpolating linear predictor.

We now bound the excess risk via Lemma 56. We first analyze the $v^{*\top}Bv^*$ part of the bound. Note that $(I - X^\top \bar{K}^{-1}X)$ is the matrix corresponding to the projection on to the orthogonal complement of the row space of X, and so $\|I - X^\top \bar{K}^{-1}X\| \le 1$. Thus, $\|B\| \le \|(I - X^\top \bar{K}^{-1}X)\|\|\Sigma\|\|(I - X^\top \bar{K}^{-1}X)\| \le \|\Sigma\|$, and so

$$v^{*\top}Bv^* \le \|\Sigma\|\|v^*\|^2. \tag{25}$$

---

5. Here, we don't need the $\mathbb{E}[y|\Phi(x)]$ to be a linear function of $\Phi(x)$.

Next, we turn to bounding the $\epsilon^\top C \epsilon$ part. We have

$$\epsilon^\top C \epsilon \leq \|C\| \|\epsilon\|^2 = \|C\| \|\mathbf{y} - Xv^*\|^2 \leq n(1 + \|v^*\|^2)\|C\|,$$

since for all i, $y_i \in [-1, 1]$ and $\|\Phi(x_i)\| = 1$. So now we need to bound $\|C\|$. We have $C \preceq \bar{K}^{-1} X(\|\Sigma\| I) X^\top \bar{K}^{-1} = \|\Sigma\| \bar{K}^{-1}$, so $\|C\| \leq \frac{\|\Sigma\|}{\lambda_{\min}(K)}$. As described in the beginning of this proof, we have $\Pr_S[\lambda_{\min}(\bar{K}) \geq 1/2] \geq 1 - \gamma$. This implies that with probability at least $1 - \gamma$ over the choice of S, we have

$$\epsilon^\top C \epsilon \leq 2n(1 + \|v^*\|^2)\|\Sigma\|. \tag{26}$$

Finally, note that the setting $L = \lceil \frac{\log(n^2)}{-\log(1 - \frac{\mu}{2})} \rceil + L_0(\Delta(n^2, \gamma))$ implies that Lemma 55 holds for $N = n^2$. So, $\|\Sigma\| \leq \frac{16 \ln(n)}{n^2}$. Plugging this bound into (25) and (26), and using Lemma 56, we get the bound stated in the theorem. ∎

Appendix H. Conditioning for One Layer ReLU Networks

In this section we establish that given a set of non-collinear points in $\mathbb{R}^d$, a sufficiently wide one layer neural network with (un-normalized) relu activations leads to a non-singular gram matrix at the output layer.

Theorem 57 *Let $S = \{x_1, x_2, \ldots, x_n\}$ be a set of n vectors in $\mathbb{R}^d$ such that each x_i is a unit length vector and for each $i \neq j$, it holds that $|x_i \cdot x_j| \leq 1 - \delta$. Let $w_1, w_2, \ldots, w_m$ be vectors drawn i.i.d. from $\mathcal{N}(0, I_{d \times d})$ and consider the feature mapping $\Phi : \mathbb{R}^n \to \mathbb{R}^m$ defined as*

$$\Phi(x) = \frac{1}{\sqrt{m}}(\sigma(w_1 \cdot x), \sigma(w_2 \cdot x), \ldots, \sigma(w_m \cdot x)),$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is the (un-normalized) relu activation defined as $\sigma(x) = \max(x, 0)$. Let $\Phi(X)$ be the corresponding $m \times n$ data matrix obtained by applying Φ to points in S, i.e., column i of $\Phi(X)$ equals $\Phi(x_i)$. There exists a universal constant $c > 0$, such that if $m \geq c \frac{n^8 \log(n/\delta)}{\delta^3}$, then w.p. at least $1 - 1/poly(n)$, we have that $\sigma_{\min}(\Phi(X)^T \Phi(X)) \geq \Omega(\frac{\delta^{3/2}}{n^3})$.

Note: The recent work of Panigrahi et al. (2020) implies a stronger bound and hence supersedes the result above. In particular, an implication of Theorem 4 in Panigrahi et al. (2020) is that under the same setting as in the Theorem above, if $m \geq \frac{n^4 \log n}{\delta^3}$, then with probability at least $1 - e^{-\tilde{\Omega}\left(m\delta^3 n^{-2} \log^{-3} n\right)}$, we have that $\sigma_{\min}(\Phi(X)^T \Phi(X)) \geq \Omega(\frac{\delta^{3/2}}{polylog(n)})$.

Proof [Proof of Theorem 57] The proof is a modification of the gradient lower bound argument as detailed in the proof of Lemma 9.3 of Allen-Zhu et al. (2018). We will show that $\sigma_{\min}(\Phi(X)) \geq \frac{\delta^{\frac{3}{4}}}{2000n^{3/2}}$. This will imply the claim of the Theorem. We will first show that for a fixed $\alpha \in \mathbb{R}^n$, such that $\|\alpha\| = 1$, $\|\Phi(X)\alpha\|$ is large. Then we will complete the argument using a union bound over an appropriate net for unit length vectors in $\mathbb{R}^n$. We have that

$$\Phi(X)\alpha = \sum_{i=1}^{n} \alpha_i \Phi(x_i)$$

and hence

$$\|\Phi(X)\alpha\|^2 = \frac{1}{m}\sum_{j=1}^{m}(\sum_{i=1}^{n}\alpha_i h_j(x_i))^2,$$

where $h_j(x_i) = \sigma(w_j \cdot x_i)$. Next, fix a particular $j \in [m]$ and let

$$T_j = \sum_{i=1}^{n}\alpha_i h_j(x_i).$$

We will first show that with non-trivial probability T_j is large. Let $i^* \in [n]$ be such that $|\alpha_{i^*}| \geq \frac{1}{\sqrt{n}}$, with ties broken arbitrarily. Next, we will write

$$w_j = \left(\sqrt{1-\theta^2}z_1 x_{i^*} + g\right) + \left(\theta z_2 x_{i^*}\right)$$

$$= w_{j,1} + w_{j,2}$$

Here we pick $\theta = \frac{\sqrt{\delta}}{5n}$ and z_1, z_2 are independent $\mathcal{N}(0,1)$ Gaussians and g is a standard d dimensional Gaussians orthogonal to x_{i^*}. Next, define G_j to be the following good event

$$G_j = \mathbb{1}\left(|w_{j,1} \cdot x_{i^*}| \leq \frac{\sqrt{\delta}}{10n} \wedge \forall i \neq i^*, |w_{j,1} \cdot x_i| > \frac{\sqrt{\delta}}{4n}\right).$$

Next, we have that $P_{w_{j,1}}(G_j) \geq \frac{\sqrt{\delta}}{50n}$. This is established in Lemma 58 at the end of the section. Conditioning on G_j, i.e., fixing the randomness in $w_{j,1}$, we notice that

$$|w_{j,2} \cdot x_{i^*}| = \theta|z_2|$$

and for any $i \neq i^*$,

$$|w_{j,2} \cdot x_i| \leq \theta|z_2|.$$

Since θz_2 is a standard Gaussian with variance θ^2, we have that the event $E : \{\theta|z_2| \in [\frac{\sqrt{\delta}}{9n}, \frac{\sqrt{\delta}}{5n}]\}$ holds with constant probability, i.e.,

$$P(E) \geq 0.2.$$

Now conditioned on $G_j \cap E$, we have that fixing the randomness in $w_{j,1}$ fixes the sign of $\sum_{i \neq i^*}\alpha_i\sigma(w_j \cdot x_i)$. Furthermore, after fixing the randomness in $w_{j,1}$, there is still a probability of 0.5 over the randomness in $w_{j,2}$ that $\alpha_{i^*}\sigma(w_j \cdot x_{i^*})$ matches that of $\alpha_{i^*}\sigma(w_{j,2} \cdot x_{i^*})$. Combining everything, we get that with probability at least $\frac{\sqrt{\delta}}{500n}$ (over w_j), it holds that $T_j^2 \geq \frac{\delta}{81n^2}$.

Next define B to be the event that for all j, $|T_j| \leq 100\sqrt{n}\sqrt{\log n \log m}$. It is easy to see that B holds with probability at least $1 - 1/poly(n)$, and hence when conditioned on B, we also have that $|T_j^2| \geq \frac{\delta}{81n^2}$ with probability at least $\frac{\sqrt{\delta}}{500n}$. Next we will argue that when conditioned on B, $\|\Phi(X)\alpha\|^2$ is large except with exponentially small probability. Combined with the fact that B happens with high probability, this will imply that over the randomness in $w_1, \ldots, w_m$, $\|\Phi(X)\alpha\|$ is large with high probability.

When conditioned on B, $\|\Phi(X)\alpha\|^2$ is an average of m independent random variables, each bounded in $[0, 100^2 n \log n \log m]$ and that

$$\mathbb{E}[\|\Phi(X)\alpha\|^2] \geq \frac{\delta^{3/2}}{81000n^3}.$$

Hence from Chernoff bound and the fact that B holds with high probability we get that with probability at least $1 - e^{\Omega(-\frac{\delta^3 m}{n^7 \log n \log m})}$, $\|X\alpha\|^2 \geq \Omega(\frac{\delta^{3/2}}{n^3})$.

Having argued the bound for a fixed α, we now consider an appropriate net over unit length vectors in $\mathbb{R}^n$ to argue that over all α, $\|\Phi(X)\alpha\|$ is large. In particular, consider an ϵ-net of the unit sphere with $\epsilon = \frac{\delta^{3/4}}{2000 n^{5/2}\sqrt{m}\sqrt{\log n \log m}}$. The size of such a net is at most $(3/\epsilon)^n$. Hence, we get that with probability at least $1 - e^{-\Omega(\frac{\delta^3 m}{n^7 \log m \log n})} e^{n \log(3/\epsilon)}$, for any vector α in the net, we have $\|\Phi(X)\alpha\| \geq \frac{\delta^{3/4}}{1000 n^{3/2}}$. This in turn implies that with the same probability, for any α on the unit sphere with $\hat{\alpha}$ being its closest vector in the net, we have

$$\|\Phi(X)\alpha\| \geq \|\Phi(X)\hat{\alpha}\| - \epsilon\|\Phi(X)\|$$
$$\geq \frac{\delta^{3/4}}{1000 n^{3/2}} - \epsilon\|\Phi(X)\|.$$

The bound then follows from noticing that via standard Gaussian concentration we have that with probability at least $1 - 1/poly(n)$, $\|\Phi(X)\| \leq O(\sqrt{mn \log m \log n})$. ∎

Lemma 58 *Let $x_1, x_2, \ldots, x_n$ be unit length vectors in $\mathbb{R}^m$, where $|x_i \cdot x_j| \leq 1 - \delta$ for $i \neq j$. Let*

$$w = \sqrt{1 - \theta^2} z x_1 + g$$

where z is $\mathcal{N}(0, 1)$, g is a standard d dimensional Gaussian orthogonal to x_1, and $\theta = \delta^{1/4}/(5n)$. Define G to be the event

$$G = \mathbb{1}\left(|w \cdot x_1| \leq \frac{\sqrt{\delta}}{10n} \wedge \forall i \neq 1, |w \cdot x_i| > \frac{\sqrt{\delta}}{4n}\right).$$

Then it holds that

$$P_w(G_j) \geq \frac{\sqrt{\delta}}{50n}.$$

Proof We have that $w \cdot x_1$ is $\mathcal{N}(0, (1 - \theta^2))$ and also that $\theta \leq 1/5$. Hence, we have that

$$P\left(|w \cdot x_1| < \frac{\sqrt{\delta}}{10n}\right) \geq \frac{\sqrt{\delta}}{25n}. \tag{27}$$

For a fixed $i \neq 1$, we have that

$$w \cdot x_i = \sqrt{1 - \theta^2} z (x_1 \cdot x_i) + g.x_i.$$

Conditioning on the fact that $|w \cdot x_1|$ is at most $\frac{\sqrt{\delta}}{10n}$, $w \cdot x_i$ is a Gaussian with mean at most $\frac{\sqrt{\delta}}{10n}$ and variance at least $\frac{\delta^2}{4}$ (since projection of x_i on g is at least $\delta/\sqrt{2}$). Hence, with probability at least $1 - 1/8n$, we have that $|w \cdot x_i| > \frac{\sqrt{\delta}}{4n}$. Using a union bound we get that, conditioned on z being small,

with probability at least 0.5, all $i \neq 1$ satisfies $|w \cdot x_i| > \sqrt{\delta}/(4n)$. Combining with (27) we get the claim. ∎

Theorem 1(b) in (Gneiting, 2013) provides a generalization of Theorem 57 to a large class of activations σ, although it doesn't prove a quantitative lower bound on the smallest eigenvalue. For completeness, we reformulate that theorem in our language here:

Theorem 59 *Suppose the Hermite expansion of σ has infinitely many even and infinitely many odd coefficients. If the inputs satisfy Assumption A, then the kernel matrix $\bar{K}$ for a 1-hidden layer neural network is non-singular.*

Appendix I. NormReLU Analyses

In this section, we derive closed form expressions for the functions $\lambda(c)$ and $b(c)$ in the definition of NormReLU:

$$\text{NormReLU}_c(x) = \lambda(c) \cdot [\max\{x - c, 0\} + b(c)].$$

Recall that the quantities $\lambda(c)$ and $b(c)$ are chosen so that the function is normalized according to (2). First, since

$$\mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,1)} [\text{NormReLU}_c(x)] = \lambda(c) \cdot \mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,1)} [\max\{X - c, 0\} + b(c)].$$

So $b(c)$ can be obtained from the equation $\mathbb{E}_{X \sim \mathcal{N}(0,1)}[\max\{X - c, 0\} + b(c)] = 0$. Let $\varphi(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2})$ be the Gaussian density function $\Phi(x) = \Pr_{X \sim \mathcal{N}(0,1)}[X \leq x]$ be the Gaussian cumulative distribution function. Using formulas for the mean of truncated normal distributions, we have

$$\mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,1)} [\max\{X - c, 0\} + b(c)] = \int_{x=c}^{\infty} (x - c)\varphi(x)dx + b(c) = \varphi(c) - (1 - \Phi(c))c + b(c).$$

So $b(c) = (1 - \Phi(c))c - \varphi(c)$. Now, to compute $\lambda(c)$, we note that

$$\mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,1)} [\text{NormReLU}_c(X)^2] = \lambda(c)^2 \cdot \mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,1)} [(\max\{X - c, 0\} + b(c))^2].$$

So $\lambda(c) = \frac{1}{\sqrt{\mathbb{E}_{X \sim \mathcal{N}(0,1)}[(\max\{X-c,0\}+b(c))^2]}}$. Using formulas for the variance of truncated normal distributions, we have

$$\mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,1)} [(\max\{X - c, 0\} + b(c))^2] = \int_{x=c}^{\infty} ((x - c) + b(c))^2 \varphi(x)dx + \int_{x=-\infty}^{c} b(c)^2$$

$$= (1 - \Phi(c)) + c\varphi(c) + 2(b(c) - c)\varphi(c) + (b(c) - c)^2(1 - \Phi(c)) + b(c)^2\Phi(c)$$

$$= (1 - \Phi(c))\Phi(c)c^2 + (1 - 2\Phi(c))\varphi(c)c + (1 - \Phi(c) - \varphi(c)^2).$$

Hence,

$$\lambda(c) = [(1 - \Phi(c))\Phi(c)c^2 + (1 - 2\Phi(c))\varphi(c)c + (1 - \Phi(c) - \varphi(c)^2)]^{-1/2}.$$

I.1. Conditioning Analysis

The following is the main theorem that provides bounds on correlations for the NormReLU activation. We specifically focus on the case $c = -1.5975$ since we use that value of c in all our experiments.

Theorem 60 *For the NormReLU activation, there exist constants α', δ' such that for any two inputs x, y such that $\|x\|^2, \|y\|^2 \in [0.5, 2.0]$ with $\frac{x^\top y}{\|x\|\|y\|} \leq 1 - \delta$ for $\delta > \delta'$ and for any $\epsilon > 0$, after a number of layers*

$$L \geq \hat{L} := \frac{2}{\alpha'} \log\left(\frac{3}{\min(\epsilon, \mu/4)}\right),$$

we have that

$$\frac{\phi_W(x)^\top \phi_W(y)}{\|\phi_W(x)\|\|\phi_W(y)\|} \leq B_{\mu/2}\left(L - \hat{L}, \delta - \delta'\right) + \delta'\epsilon$$

The constants are given by $\alpha' = 0.0798$ and $\delta' = 0.0185$.

Remark 61 *Note that the above theorem implies that for any two inputs with corelation sufficiently bounded away from 1 (quantified by δ'), for any $\epsilon \to 0$, after $O(\log(1/\epsilon))$ layers the corelation decays to ϵ, recovering our exponential conditioning results for NormReLU. Furthermore, the range $[0.5, 2.0]$, is chosen arbitrarily, the related constants will depend on the choice of the range.*

Before stating the proof we will require some preliminaries. We request the reader to recall the notation established in Appendix D. The proof follows the same schematic of the proof for Theorem 13, with the main caveat that unlike in the case of odd function the constant bias term for NormReLU, $a_0^{\{\gamma\}}$ is not necessarily 0 at all γ (recall that our conditions impose that $a_0^{\{\gamma\}} = 1$) and hence it needs to be accounted for in our analysis. We now present closed form expressions for the quantities of interest for NormReLU. In particular it can be easily derived via calculations similar to those done for the formulas for $b(c)$ and $\lambda(c)$ that

$$\frac{a_0^{\{\gamma\}}}{\lambda(c)} = \frac{\mathbb{E}_{X \sim \mathcal{N}(0,1)}[\text{NormReLU}_c(\sqrt{\gamma}X)]}{\lambda(c)} = \sqrt{\gamma}\varphi(c/\sqrt{\gamma}) - (1 - \Phi(c/\sqrt{\gamma}))c + b(c)$$

Furthermore we can also derive such a closed form expression for the norm transfer function $\hat{\sigma}_l(\gamma)$, and its derivative viz.

$$\frac{\hat{\sigma}_l(\gamma)}{\lambda^2(c)} = \sum_{j=0}^{\infty} \frac{\left(a_j^{\{\gamma\}}\right)^2}{\lambda^2(c)} = \frac{\mathbb{E}_{X \sim \mathcal{N}(0,1)}[\text{NormReLU}_c^2(\sqrt{\gamma}X)]}{\lambda^2(c)}$$

$$= (c^2 + \gamma - 2cb(c))(1 - \Phi(c/\sqrt{\gamma})) + (2b(c) - c)\sqrt{\gamma}\varphi(c/\sqrt{\gamma}) + b(c)^2$$

$$\frac{\hat{\sigma}_l'(\gamma)}{\lambda^2(c)} = 1 - \Phi(c/\sqrt{\gamma}) + \frac{b(c)}{\sqrt{\gamma}}\varphi(c/\sqrt{\gamma})$$

Furthermore we define the following function which represents a form of bias which increases correlations over layers:

$$\text{bias}(\gamma) := \frac{\left(a_0^{\{\gamma\}}\right)^2}{\sum_{j=0}^{\infty}\left(a_j^{\{\gamma\}}\right)^2}$$

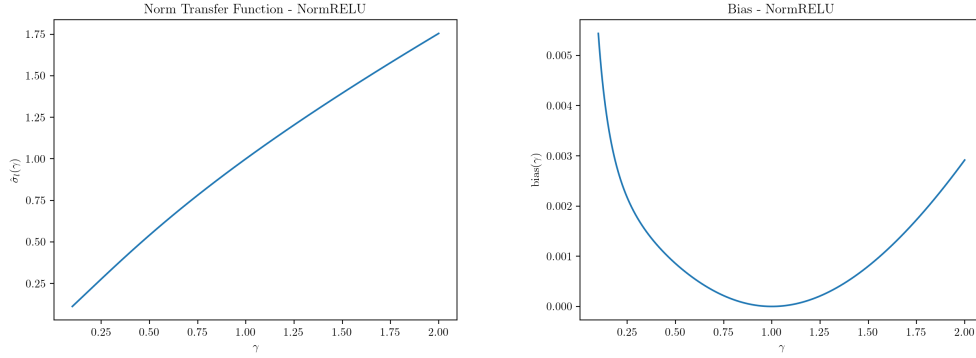The following observations are evident from the graphs in Figure 6.

Figure 6: Plots of the functions $\hat{\sigma}_l(\gamma)$ and bias in the range $[0.5, 2]$. They can be seen to be concave and convex respectively.

Fact 1 *The following properties hold for the NormReLU activation*

1. *For $\gamma \in [0.5, 2]$, $\hat{\sigma}_l(\gamma)$ is a non-decreasing concave function with $\hat{\sigma}_l(0.5) \sim 0.5399$ and $\hat{\sigma}'_l(1) \sim 0.8428$.*

2. *For $\gamma \in [0.5, 2]$, $\mathrm{bias}(\gamma)$ is a convex non-negative function with minimum value at $\gamma = 1$, $\mathrm{bias}(1) = 0$. Furthermore we have that $\mathrm{bias}(0.5) \sim 0.00086$ and $\mathrm{bias}(2) \sim 0.0029$.*

The above facts lead to the following simple calculations. Firstly note that since $\mathrm{bias}(\gamma)$ is a convex function we have the following

$$\mathrm{bias}(\gamma) \leq \begin{cases} 2\mathrm{bias}(0.5)(1 - \gamma) & \text{if } \gamma \in [0.5, 1] \\ \mathrm{bias}(2)(\gamma - 1) & \text{if } \gamma \in [1, 2] \end{cases} \tag{28}$$

The constant (defined in [Theorem 38](#)) $\alpha_\sigma = \min(\alpha_\sigma^-, \alpha_\sigma^+)$ for NormReLU activation are as follows.

$$\alpha_\sigma^- := 2\hat{\sigma}_l(0.5) - 1 \sim 0.0798$$

$$\alpha_\sigma^+ := 1 - \hat{\sigma}'_l(1) \sim 0.1572$$

With these in place we are now ready to prove the theorem.

Proof Firstly consider two sequences defined by the following:

$$(1 - \gamma_l^-) = (1 - \alpha_\sigma^-)(1 - \gamma_{l-1}^-) \qquad \gamma_0^- = 0.5$$

$$(\gamma_l^+ - 1) = (1 - \alpha_\sigma^+)(\gamma_{l-1}^- - 1) \qquad \gamma_0^+ = 2.0$$

The following is immediate from the above derivations above and [(28)](#).

$$\sum_{l=0}^{\infty} \mathrm{bias}(\gamma_l^-) = \frac{\mathrm{bias}(0.5)}{\alpha_\sigma^-}, \quad \sum_{l=L'}^{\infty} \mathrm{bias}(\gamma_l^-) = \frac{\mathrm{bias}(0.5)\epsilon}{\alpha_\sigma^-}$$

$$\sum_{l=0}^{\infty} \mathrm{bias}(\gamma_l^+) = \frac{\mathrm{bias}(2)}{\alpha_\sigma^+}, \quad \sum_{l=L'}^{\infty} \mathrm{bias}(\gamma_l^+) = \frac{\mathrm{bias}(2)\epsilon}{\alpha_\sigma^+}$$

Now note from Theorem 38 we have that if $\|x\|^2 \leq 1$, the norm of the representation at layer l, $\bar{k}^l(x,x) \in [\gamma_l^-, 1]$ and correspondingly if $\|x\|^2 \geq 1$, $\bar{k}^l(x,x) \in [1, \gamma_l^+]$. The same holds for y as well. Furthermore from the properties of the bias function we have that,

$$\mathrm{bias}(\bar{k}^l(x,x)) \leq \begin{cases} \mathrm{bias}(\gamma_l^-) & \text{if } \|x\| \leq 1 \\ \mathrm{bias}(\gamma_l^+) & \text{if } \|x\| \geq 1 \end{cases}.$$

The same holds for y. Define the following shorthand for the corelation between the representations at layer l:

$$\rho_l := \frac{\bar{k}^l(x,y)}{\sqrt{\bar{k}^l(x,x)\bar{k}^l(y,y)}}, \qquad \delta_l := 1 - |\rho_l|.$$

Using the analysis derived in the proof of Lemma 42 we have the following,

$$|\rho_l| \leq \sqrt{\mathrm{bias}(\bar{k}^{l-1}(x,x))\mathrm{bias}(\bar{k}^{l-1}(y,y))} + \sqrt{\frac{\sum_{j=1}^{\infty}\left(a_j^{\{\bar{k}^{l-1}(x,x)\}}\right)^2 |\rho_{l-1}|^j}{\sum_{j=1}^{\infty}\left(a_j^{\{\bar{k}^{l-1}(x,x)\}}\right)^2}}\sqrt{\frac{\sum_{j=1}^{\infty}\left(a_j^{\{\bar{k}^{l-1}(y,y)\}}\right)^2 |\rho_{l-1}|^j}{\sum_{j=1}^{\infty}\left(a_j^{\{\bar{k}^{l-1}(y,y)\}}\right)^2}}$$

$$\leq \sqrt{\mathrm{bias}(\bar{k}^{l-1}(x,x))\mathrm{bias}(\bar{k}^{l-1}(y,y))} + (1-\delta_{l-1})\sqrt{(1-\mu^{\{\bar{k}^{l-1}(x,x)\}}\delta_{l-1})}\sqrt{(1-\mu^{\{\bar{k}^{l-1}(y,y)\}}\delta_{l-1})}.$$
$$(29)$$

Iterating the above equation over L' steps we immediately get that

$$|\rho_{L'}| \leq 1 - \delta_0 + \sum_{l=1}^{L'}\sqrt{\mathrm{bias}(\bar{k}^{l-1}(x,x))\mathrm{bias}(\bar{k}^{l-1}(y,y))}$$

$$\leq 1 - \delta + \max\left(\sum_{l=0}^{\infty}\mathrm{bias}(\gamma_l^-), \sum_{l=0}^{\infty}\mathrm{bias}(\gamma_l^+)\right)$$

$$= 1 - \delta + \max\left(\frac{\mathrm{bias}(0.5)}{\alpha_\sigma^-}, \frac{\mathrm{bias}(2)}{\alpha_\sigma^+}\right).$$

We define

$$\delta' = \max\left(\frac{\mathrm{bias}(0.5)}{\alpha_\sigma^-}, \frac{\mathrm{bias}(2)}{\alpha_\sigma^+}\right) \sim 0.0185.$$

Furthermore as argued in the proof of Theorem 13 we have that for any $l \geq L'$ we have that

$$\mu^{\{\bar{k}^l(x,x)\}}, \mu^{\{\bar{k}^l(y,y)\}} \geq \mu/2.$$

Now using the above and iterating (29) from L' onwards we get that

$$|\rho_L| \leq B_{\mu/2}(L - L', \delta - \delta') + \max\left(\sum_{l=L'}^{\infty}\mathrm{bias}(\gamma_l^-), \sum_{l=L'}^{\infty}\mathrm{bias}(\gamma_l^+)\right)$$

$$\leq B_{\mu/2}(L - L', \delta - \delta') + \max\left(\frac{\mathrm{bias}(0.5)\epsilon}{\alpha_\sigma^-}, \frac{\mathrm{bias}(2)\epsilon}{\alpha_\sigma^+}\right)$$

$$\leq B_{\mu/2}(L - L', \delta - \delta') + \delta'\epsilon.$$
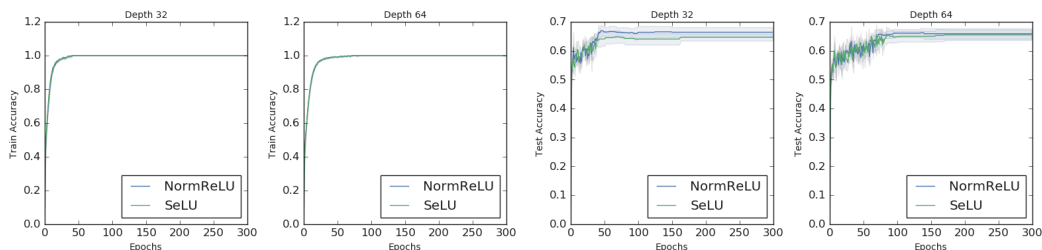
This finishes the proof. ∎

Figure 7: Train and test accuracy vs. the number of epochs. The plots are obtained by training a depth 32 and a depth 64 fully connected network with 3000 hidden units in each layer.
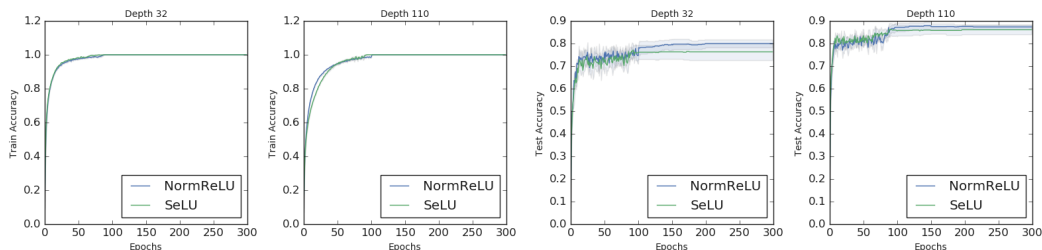

Figure 8: Train and test accuracy vs. the number of epochs. The plots are obtained by training a depth 32 and a depth 110 ResNet architecture with skip connections but without the use of batch normalization.

I.2. Comparison of NormReLU and SeLU

In this section we compare the training and generalization behavior of the NormReLU activation that we proposed in this work and the closely related SeLU activation Klambauer et al. (2017) (see Figure 2). Similar to Section 7 we first train, on the CIFAR-10 dataset, fully connected feedforward networks with depths 32 and 64 and containing 3000 hidden units in each layer. Figure 7 shows the training and the test accuracy achieved when using either NormReLU or the SeLU activation. As can be seen, the training and generalization behavior of the two activations is very similar.

We next train ResNet architectures of depth 32 and 110 and with skip connections on the CIFAR-10 dataset. As in Section 7, we switch off batch normalization. Figure 8 shows that in this case as well, the two activations behave similarly in terms of speed of training and generalization performance, with NormReLU achieving slightly higher test accuracies.

I.3. Discussion of Training Settings

All the experiments in this work were performed on the CIFAR-10 dataset (Krizhevsky et al., 2009). In each case, the learning rate was chosen via cross validation. All the models were trained for 300 epochs and the learning rate was reduced by a factor of 0.1 after every 100 iterations. In each case, the model was trained for 10 independent runs. For the experiment in Figure 1 the fully connected networks were trained with learning rates of 0.005 for batch normalization, 0.003 for layer normalization and 0.0009 for NormReLU. The convolutional networks were trained with learning rates of 0.1 for batch normalization, 0.05 for layer normalization and 0.001 for NormReLU. For the experiment in Figure 3, the fully connected networks of depth 32 were trained with learning rates of 0.01 for batch normalization, 0.001 for layer normalization and 0.0009 for NormReLU. The

depth 64 networks were trained with learning rates of 0.007 for batch normalization, 0.001 for layer normalization and 0.0009 for NormReLU.

For the experiment in Figure 4 the depth 32 networks were trained with learning rates of 0.1 for batch normalization and 0.009 for NormReLU. The depth 110 networks were trained with learning rates of 0.05 for batch normalization and 0.005 for NormReLU. To compare with fixup initialization as in Figure 5 we use the fixup-110 architecture as proposed in the work of Zhang et al. (2019). We switch on data augmentation as used in Zhang et al. (2019) and train with fixup initialization using a learning rate of 0.1 and the learning rate schedule as proposed by the authors in Zhang et al. (2019). We train with NormReLU using a learning rate of 0.005. For the experiment in Figure 7 we use a learning rate of 0.001 for depth 32 and of 0.0005 for depth 64. The same learning rate is used for both NormReLU and SeLU. Finally, for the experiment in Figure 8 we use a learning rate of 0.01 at depth 32 and of 0.009 at depth 110 and keep it the same for both the activations.