Proceedings of Machine Learning Research vol 178:1–40, 2022 35th Annual Conference on Learning Theory

Neural Networks can Learn Representations with Gradient Descent

Alex Damian AD27@PRINCETON.EDU
Princeton University

Jason D. Lee JASONLEE@PRINCETON.EDU
Princeton University

Mahdi Soltanolkotabi SOLTANOL@USC.EDU
University of Southern California

Editors: Po-Ling Loh and Maxim Raginsky

Abstract

Significant theoretical work has established that in specific regimes, neural networks trained by gradient descent behave like kernel methods. However, in practice, it is known that neural networks strongly outperform their associated kernels. In this work, we explain this gap by demonstrating that there is a large class of functions which cannot be efficiently learned by kernel methods but can be easily learned with gradient descent on a two layer neural network outside the kernel regime by learning representations that are relevant to the target task. We also demonstrate that these representations allow for efficient transfer learning, which is impossible in the kernel regime.

Specifically, we consider the problem of learning polynomials which depend on only a few relevant directions, i.e. of the form $f^?(x) = g(Ux)$ where $U : \mathbb{R}^d \; ! \; \mathbb{R}^r$ with $d \;\; r$. When the degree of $f^?$ is p, it is known that $n \;\; d^p$ samples are necessary to learn $f^?$ in the kernel regime. Our primary result is that gradient descent learns a representation of the data which depends only on the directions relevant to $f^?$. This results in an improved sample complexity of $n \;\; d^2 r + dr^p$. Furthermore, in a transfer learning setup where the data distributions in the source and target domain share the same representation U but have different polynomial heads we show that a popular heuristic for transfer learning has a target sample complexity independent of d.

Keywords: neural network, gradient descent, representation learning, transfer learning, kernel

1. Introduction

Crucial to the practical success of deep learning is the ability of gradient-based algorithms to learn good feature representations from the training data and learn simple functions on top of these representations. Despite significant progress towards a theoretical foundation for neural networks, a robust understanding of this unique representation learning capability of gradient descent methods has remained elusive. A major challenge is that due to the highly nonconvex loss landscape, establishing convergence to a global optimum that achieves near zero training loss is challenging. Furthermore, due to the overparameterized nature of modern neural nets (containing many more parameters than training data) the training landscape has many global optima. In fact, there are many global optima with poor generalization performance (Zhang et al., 2016; Liu et al., 2020). This paper thus focuses on answering this intriguing question:

How do gradient-based methods learn feature representations and why do these representations allow for efficient generalization and transfer learning?

The most prominent contemporary approach to understanding neural networks is the linearization or neural tangent kernel (NTK) (Soltanolkotabi et al., 2018; Jacot et al., 2018) technique. The premise of the linearization method is that the dynamics of gradient descent are well-approximated by gradient descent on a linear regression instance with fixed feature representation. Using this linearization technique, it is possible to prove convergence to a zero training loss point (Soltanolkotabi et al., 2018; Du et al., 2018b, 2019a). However, this technique often requires unrealistic hyper-parameter choices (e.g. small learning rate, large initialization, or wide networks) that does not allow the features to evolve across the iterations and thus the generalization error with this technique cannot be better than that of a kernel method. Indeed, precise lower bounds show that the NTK solutions do not generalize better than the polynomial kernel (Ghorbani et al., 2019a). As a result this regime of training is also sometimes referred to as the lazy regime Chizat et al. (2019).[1] In practice, neural networks far outperform their corresponding induced kernels (Arora et al., 2019a). Therefore, understanding the representation learning of neural networks beyond the lazy regime is of fundamental importance.

In this paper, we initiate the study of the representation learning of neural networks beyond this NTK/linear/lazy regime. To this aim, we consider the problem of learning polynomials with low-dimensional latent representation of the form $f(x) = g(Ux)$, where U maps from d to r dimensions with $d \quad r$ with g a multivariate polynomial of degree p. This is a natural choice as the failure of the NTK solution is in part due to its inability to learn data-dependent feature representations that adapt to the intrinsic low latent dimensionality of the ground truth function. Existing analysis based on the NTK regime provably require $n \quad d^p$ samples (Ghorbani et al., 2019a) to learn any degree p polynomial, even if they only depend on a few relevant directions. In contrast we show that gradient descent from random initialization only requires $n \quad d^2 + r^p$ samples, breaking the sample complexity barrier dictated by NTK proof techniques. More specifically, our contributions are as follows:

1. Feature Learning: When the target function $f^? = g(Ux)$ only depends on the projection of x onto a hidden subspace span(U), we show that gradient descent learns features that span span(U). Leveraging these features, gradient descent can reach vanishing training loss with a very small network which guarantees good generalization performance. See Section 5.1.

2. Improved Sample Complexity: Using classical generalization theory, we demonstrate that when $f^? : R^d \; ! \quad R$ is a polynomial of degree p which depends on r relevant dimensions (Assumption 1), gradient descent on a two layer neural network learns $f^?$ with only $n \quad d^2 r + dr^p$ samples. This contrasts with the lower bound for random features/NTK methods which require d^p samples to learn any degree p polynomial. See Theorem 1.

3. Transfer learning: We show that when the target task ground truth is $f^?_{target}(x) = g(Ux)$, then by simply retraining the network head, gradient descent learns $f^?_{target}$ with only $N \quad r \quad ^p$ target samples and width $m \quad r \;,^p$ which is independent of the ambient dimension d. In contrast, learning from scratch would require $N \quad d^{Omega(p)}$ target samples.

4. Lower Bound: Finally, we show a lower bound that demonstrates our non-degeneracy assumption (Assumption 2) is strictly necessary. Without the non-degeneracy, there is a family

[1] See Section 4 for a more in depth discussion of this literature and other related work.

of polynomials which depend on single relevant dimensions (i.e. of the form $f^\star(x) = g(u \cdot x)$) which cannot be learned with fewer than $n \gtrsim d^{p/2}$ by any gradient descent based learner.

2. Setup

2.1. Input Distribution and Target Function

In this paper we focus on learning a target function $f^\star(x) : \mathbb{R}^d \to \mathbb{R}$ over the input distribution $\mathcal{D} := \mathcal{N}(0, I_d)$. We assume that $f^\star$ is a degree p polynomial, normalized so that $\mathbb{E}_{x \sim \mathcal{D}}[f^\star(x)^2] = 1$. We will attempt to learn $f^\star$ given n i.i.d. datapoints $\{x_i, y_i\}_{i \in [n]}$ with

$$x_i \sim \mathcal{D}, \qquad y_i = f^\star(x_i) + \epsilon_i \quad \text{and} \quad \epsilon_i \sim \{-\varsigma, \varsigma\}$$

where ς^2 controls the strength of the label noise.

In order to make the problem of learning $f^\star$ tractable, additional assumptions are necessary. The set of degree p polynomials in d dimensions span a linear subspace of $L^2(\mathcal{D})$ of dimension $\Theta(d^p)$. Learning arbitrary degree p polynomials therefore requires $n \gtrsim d^p$ samples. We follow Chen and Meka (2020); Chen et al. (2020b) in assuming that the ground truth $f^\star$ has a special low dimensional latent structure. Specifically, we assume that $f^\star$ only depends on a small number of relevant dimensions and that the expected Hessian is non degenerate. We show in Theorem 2 that this non degeneracy assumption is strictly necessary to avoid sample complexity $d^{\Omega(p)}$.

Assumption 1 There exists a function $g : \mathbb{R}^r \to \mathbb{R}$ and linearly independent vectors $u_1, \ldots, u_r$ such that for all $x \in \mathbb{R}^d$,

$$f^\star(x) = g(\langle x, u_1 \rangle, \ldots, \langle x, u_r \rangle).$$

We will call $S^\star := \mathrm{span}(u_1, \ldots, u_r)$ the principal subspace of $f^\star$. We will also denote by $\Pi^\star := \Pi_{S^\star}$ the orthogonal projection onto $S^\star$.

Note that Assumption 1 guarantees that for any x, $\mathrm{span}(\nabla^2 f^\star(x)) \subseteq S^\star$. In particular, if we denote the average Hessian by $H := \mathbb{E}_{x \sim \mathcal{D}}[\nabla^2 f^\star(x)]$, we have that $\mathrm{span}(H) \subseteq S^\star$ so that H has rank at most r. The following non-degeneracy assumption states that H has rank exactly r.

Assumption 2 $H := \mathbb{E}_{x \sim \mathcal{D}}[\nabla^2 f^\star(x)]$ has rank r, i.e. $\mathrm{span}(H) = S^\star$.

We will also denote the normalized condition number of H by $\kappa := \frac{\|H^\dagger\|}{\sqrt{r}}$.

2.2. The Network and Loss

Let $\sigma(x) = \mathrm{ReLU}(x) = \max(0, x)$, let $a \in \mathbb{R}^m$, $W \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$, and let $\theta = (a, W, b)$. We define the neural network f_θ by

$$f_\theta(x) = a^T \sigma(Wx + b) = \sum_{j=1}^{m} a_j \sigma(w_j \cdot x + b_j),$$

where m denotes the width of the network. We use a symmetric initialization, so that $f_{\theta_0}(x) = 0$ (Chizat and Bach, 2018a). Explicitly, we will assume that m is an even number and that

$$a_j = -a_{m-j}, \quad w_j = w_{m-j} \quad \text{and} \quad b_j = b_{m-j} \qquad \forall j \in [m/2].$$

We will use the following initialization:

$$a_j \quad f \quad 1; 1g; \qquad w_j \quad N \quad 0; \frac{1}{d} I_d \qquad \text{and} \quad b_j = 0:$$

We note that while we focus on such symmetric initialization for clarity of exposition, our results also hold with small random initialization that is not necessarily symmetric. This holds by simple modifications in the proof accounting for the small nonzero output of the network at initialization. We will also denote the empirical and population losses by $L()$ and $L_D()$ respectively:

$$L() = \frac{1}{n} X_{i=1}^{n} (f(x_i) \qquad y_i)^2 \quad \text{and} \quad L_D() = E_{xD} \overset{h}{} (f(x) \quad f\ (x^?)) \overset{i}{}{}^2 :$$

2.3. Notation

We use $.; O();$
$()$ to denote quantities that are related by absolute constants and we treat $p; \& =$
$O(1)$. We use $O;$
to hide additional dependencies on polylog(mnd). We denote the $L^1(D)$ and
$L^2(D)$ losses of a function f by $E_{x;y} jf(x) \quad yj$ and $E_{x;y} (f(x) \quad y)^2$ respectively where $x \quad D, y$
$= f^?(x) + $, and $f\&g$.

3. Main Results

Before we formally state our main result let us specify the exact form of gradient-based training we use in our theory.

```
Input: Learning rates $_t$, weight decay $_t$, number of steps T
preprocess data
  | $y_i \quad y_i \quad \frac{1}{n} P_{i=1}^{n} y_i, \quad \frac{1}{n} P_{i=1}^{n} y_i x_i \quad x_i$ for $i = 1; : : : ; n$
end
$W^{(1)} \quad W^{(0)} \quad {}_1[r_W L() + {}_1 W]$ re-
initialize $b_j \quad N(0; 1)$
for $t = 2$ to $T$ do
  | $a^{(t)} \quad a^{(t \quad 1)} \quad {}_t[r_a L(^{(t \quad 1)}) + {}_t a^{(t \quad 1)}]$ end
return Prediction function $x \; ! \quad + \quad x + a^T(Wx + b)$
```

Algorithm 1: Gradient-based training

With this algorithm in place, we are now ready to state our main result.

Theorem 1 Consider the data model, network and loss per Section 2 and train the network via Algorithm 1 with parameters $_1 = O(\tilde{} \sqrt{d}), {}_1 = {}^1$, and $_t = \; ; _t = $ for $t \; 2$. Further-more, assume n $(d^{22}r)$ $\tilde{}$ $\tilde{}$ and d $(r^{3=2})$. Then, there exists such that if is sufficiently small, $T = ({}^1 \; {}^1)$ and $^{(T)}$ denotes the final iterate of Algorithm 1, we have that the excess population loss in $L^1(D)$ is bounded with probability at least 0:99 by

$$E_{x;y} jf_{(T)}(x) \qquad yj \quad \& \; O\tilde{} \; r \; \frac{dr p \; 2p}{n} + r \; \frac{r p \; 2p}{m} + \frac{1}{n^{1=4}} \; ! \; :$$

It is useful to note that the use of in the algorithm corresponds to the common practice of weight decay and its value is chosen in such a way that $a^{(T)}$ B_a, i.e. to solve a constrained minimization problem (see Section 5.1). In practice, one simply tunes the hyperparameter in order to achieve the desired tradeoff between training and test loss.

An intriguing aspect of the above result is that despite the fact that $f^?$ may be of arbitrarily high degree, learning $f^?$ requires only $n \& dr^p + d^2 r$ samples and only requires a very small network with $m \& r^p$. We note that our dependence on the latent dimension r is near optimal as the minimax sample complexity even when the principal subspace $S^?$ is known is (r^p).

We show in Theorem 3 that by resampling the data after the first step, the sample complexity can be further reduced to $d^2 r + r^p$, dropping a factor of d from the second term. The extra factor of d results from the dependence between the data used in the first and second stages and we believe that a more careful analysis could remove this additional factor.

We contrast Theorem 1 with the following lower bound for learning a function class which satisfies Assumption 1 with $r = 1$ but does not satisfy Assumption 2.

Theorem 2 For any p 0, there exists a function class F_p of polynomials of degree p, each of which depends on a single relevant dimension, such that any correlational statistical query learner using q queries requires a tolerance of at most

$$\frac{\log^{p=4}(qd)}{d^{p=4}}$$

in order to output a function f 2 F_p with $L^2(D)$ loss at most 1.

Using the heuristic $\frac{p}{}, \frac{1}{n}$ which represents the expected scale of the concentration error, we get the immediate corollary that violating Assumption 2 allows us to construct a function class which any neural network with polynomially many parameters trained for polynomially many steps of gradient descent cannot learn without at least $n \& d^{p=2}$ samples. We emphasize that this is only a heuristic argument as concentration errors are random rather than adversarial.

On the other hand, Theorem 1 shows that incorporating Assumption 2 allows gradient descent to efficiently learn polynomials of arbitrarily high degree with only d^2 samples.

The difference in sample complexity between Theorem 1 and Theorem 2 is that in Theorem 1, our non-degeneracy assumption (Assumption 2) allows the network f to extract useful features that aid robust learning and allowed learning high degree polynomials with $n \& d^2$ samples. Theorem 2 shows that violating this assumption allows us to construct a function class which cannot be learned without d

$^{(p)}$ samples, demonstrating the necessity of Assumption 2.

The fact that the network f extracts useful features not only allows it to learn $f^?$ efficiently, but also allows for efficient transfer learning. In particular, Theorem 3 shows that we can efficiently learn any target polynomial $g^?(x)$ that depends on the same relevant dimensions as $f^?$ with sample complexity independent of d by simply truncating and retraining the head of the network:

Theorem 3 Let $g^?(x)$ be a degree p polynomial with $E_D[g^?(x)^2] = 1$ and $g(x) = g(^?x)$ for all x 2 R^d. Let $D_N = f(x_i; y_i)g_{i2[N]}$ be a second dataset with $y_i = g(x_i) + {}_i$. We retrain the last layer of the network f in Theorem 1 with gradient descent with learning rate and weight decay , i.e. we will use the function class:

$$g_a(x) = a^T(W^{(1)}x + b)$$

where $W^{(1)}$ is the second iterate of Algorithm 1 on the pre-training dataset. Assume that $d =$ $(\tilde{r}^{3=2})$. Then there exists such that if the network is $\tilde{}$pretrained on n $(d^{22}r)$ datapoints from $f^{?}$ and is sufficiently small, the excess population loss in $L^{1}(\tilde{D})$ after $T = (\ ^{1} \ ^{1})$ steps
is bounded with probability at least 0:99 by

$$E_{x;y}\, jg_{a(T)}(x) \quad yj \quad \& \; \tilde{O} \quad \underline{s} \quad \frac{r^{p \quad 2p}}{\min(m;N)} + \frac{1}{N^{1=4}} \quad ! \quad :$$

Learning $g^{?}(x)$ therefore only requires $N; m \ \& \ r^{p}$, which is independent of the ambient dimension d. We note that this is minimax optimal for learning arbitrary degree p polynomials even when the hidden subspace $S^{?}$ is known. Theorem 3 also shows that $n \quad d^{2}r$ pre-training samples are necessary for gradient descent to learn the subspace $S^{?}$ from the pre-training data.

4. Related work

A growing body of recent work show the connection between gradient descent on the full network and the Neural Tangent Kernel (NTK) Jacot et al. (2018); Oymak and Soltanolkotabi (2019, 2020); Du et al. (2019b); Arora et al. (2019b); Du et al. (2018a); Lee et al. (2019). Using this technique one can prove concrete results about neural network training (Li and Liang, 2018; Du et al., 2018a, 2019b; Allen-Zhu et al., 2018; Zou et al., 2018) and generalization (Arora et al., 2019b; Oymak et al., 2019; Allen-Zhu et al., 2019; Cao and Gu, 2019; Oymak et al., 2021) in the kernel regime. The key idea is that for a large enough initialization, it suffices to consider a linearization of the neural network around the origin. This allows connecting the analysis of neural networks with the well-studied theory of kernel methods. This is also sometimes referred to as lazy training, as with such an initialization the parameters of the neural networks stay close to the parameters at initialization and these results can only show that neural networks are as powerful as shallow learners such as kernels. There is however growing evidence that this NTK-style analysis might not be sufficient to completely explain the success of neural networks in practice. The papers Chizat et al. (2019); Woodworth et al. (2019) provides empirical evidence that by choosing a smaller initialization the test error of the neural network decreases. A similar performance gap between the performance of the NTK and neural networks has been observed in Ghorbani et al. (2020). This NTK-style analysis however does not yield satisfactory results in the setting studied in this paper. In particular for learning the polynomials of the form we study in this paper, Ghorbani et al. (2019b) demonstrates that one needs at least d^{p} samples in the kernel regime. In contrast, our results only require on the order of d^{2} samples.

Leveraging the fact that linearized models are not feature learners, Ghorbani et al. (2019b) and Wei et al. (2019) showed precise upper and lower bounds on the sample complexity of NTK methods. They showed that because NTK is unable to learn new features, learning any polynomial in dimension d of degree p requires $n = \ (d^{p})$ samples, which gives no improvement over polynomial kernels. On the empirical front, the NTK linearization analysis is also lacking. Arora et al. (2019a) demonstrated that the kernel predictor loses more than 20% in test accuracy relative to a deep network trained with SGD and state-of-art regularization on CIFAR-10. Our work is motivated by the contrast between these negative theoretical results for linearized NTK models and the spectacular empirical performance of deep learning.

The gap between such shallow learners and the full neural network has been established in theory (Wei et al., 2019; Allen-Zhu and Li, 2020, 2019; Yehudai and Shamir, 2019b; Ghorbani et al., 2019a; Woodworth et al., 2020; Dyer and Gur-Ari, 2019; Du and Lee, 2018) and observed in practice (Arora et al., 2019a; Lee et al., 2019; Chizat and Bach, 2018a). There is an emerging literature on learning beyond the lazy/NTK regime in the small initialization setting. The papers Li et al. (2018); Stöger and Soltanolkotabi (2021) shows that for the problem of low-rank reconstruction in a non-lazy regime with small random initialization gradient descent finds globally optimal solutions with good generalization capability. This is carried out by utilizing a spectral bias phenomena exhibited by the early stages of gradient descent from small random initialization that puts the iterates on the trajectory towards generalizable models. For the problem of tensor decomposition it has also been shown that gradient descent with small initialization is able to leverage low-rank structure (Wang et al., 2020). In Li et al. (2020), it has been shown that neural networks with orthogonal weights can be learned via SGD and outperform any kernel method. One crucial element in their analysis is that the early stage of the training is connected with learning the first and second mo-ment of the data. Higher-order approximations of the training dynamics (Bai and Lee, 2020; Bai et al., 2020) and the Neural Tangent Hierarchy (Huang and Yau, 2019) have also been recently pro-posed towards closing this gap. None of the above papers, however, focus on learning polynomial representations efficiently via neural networks as carried out in this paper.

Another line of work focuses on learning single activations such as the ReLU function. In this context (Yehudai and Shamir, 2019a) shows that it is hard to learn a single ReLU activation via stochastic gradient descent with random features where as learning such activations is possible in a non-NTK regime (Soltanolkotabi, 2017; Goel et al., 2017, 2019) again highlighting this impor-tant gap. In related work where the label also only depends on a single relevant direction (Daniely and Malach, 2020), the authors show that in the context of learning the parity function, gradient descent is able to efficiently learn the planted set. However, this is a result of the unbalanced data distribution which skews the gradient towards the planted set. In contrast, we consider isotropic Gaussian data so that no information can be extracted from the data distribution itself and features must be extracted from higher order correlations between the data and the labels. Chen and Meka (2020) also studied the problem of learning polynomials of few relevant dimensions. They provide an algorithm that learns polynomials of degree p in d dimensions that depends on r hidden dimen-sions with n & $C(r; p)d$ samples where $C(r; p)$ is an unspecified function of $r; p$ which is likely exponential in r. However, their algorithm is not a variant of gradient descent, and requires a clever spectral initialization. On the other hand, this work focuses on the ability of gradient descent to automatically extract hidden features and learn representations from the data.

There is also a line of work Mei et al. (2018); Chizat and Bach (2018b); Mei et al. (2019); Javanmard et al. (2020); Sirignano and Spiliopoulos (2020); Wei et al. (2019), which is concerned with the mean-field analysis of neural networks. The insight is that for sufficiently large width the training dynamics of the neural network can be coupled with the evolution of a probability distribution described by a PDE. These papers use a smaller initialization than in the NTK-regime and, hence, the parameters can move away from the initialization. However, these results do not provide explicit convergence rates and require an unrealistically large width of the neural network. To the extent of our knowledge such an analysis technique has not been used to show efficient learning of polynomial representations using neural networks as carried out in this paper.

A concurrent line of work studied the feature learning ability of gradient descent in the mean field regime with data sampled from the boolean cube (Abbe et al., 2022). The authors identified a necessary and sufficient condition for learning with sample complexity linear in d, dubbed the merged staircase property, in the special case when the hidden weights of the two layer neural network are initialized at 0. However, the zero initialization hinders the feature learning ability of the network. For example, the boolean function XOR violates the merged staircase property, however noisy XOR is known to be learnable by two layer neural networks with sample complexity linear in d (Bai and Lee, 2020; Chen et al., 2020a). In this work we study the impact that the nonzero initialization of the hidden weights has on the feature learning ability of neural networks.

5. Proof Sketches

5.1. Proof of Theorems 1 and 3

The proofs of Theorems 1 and 3 are essentially identical so we will focus on Theorem 1. We begin by noting that the symmetric initialization implies that $f(x) = 0$ for all x 2 R^d. This implies that the population gradient of each feature w_j can be written as

$$r_{w_j} L_D() = E_{x D}\, 2(f(x) \qquad f^{?}(x)) r_{w_j} f(x) = \quad 2\, E_{x D}[f^{?}(x) r_{w_j} f(x)]:$$

Using the chain rule, we can further expand this as

$$2\, E_{x D}[f^{?}(x) r_{w_j} f(x)] = \quad 2a_j\, E_x[f^{?}(x) x 1_{w_j x 0}]:$$

The main computation that drives Theorems 1 and 3 is that for any unit vector w 2 R^d, the expression $E_x[f^{?}(x)x1_{wx0}]$ has a natural series expansion in powers of w, which can be computed explicitly in terms of the Hermite expansions of $f^{?}$ and ^{0}. Explicitly, if $C_k = E_x[r^k f^{?}(x)]$ is a symmetric k tensor denoting the expected kth derivative of $f^{?}$ and c_k are the Hermite coefficients of $^{0}(x) = 1_{x0}$,

$$E_x[f^{?}(x)x1_{wx+b0}] = \overset{p\ 1}{X} \frac{1}{k!} \overset{h}{} c_{k+1}C_{k+1}(w^{k}) + c_{k+2}wC_k(w^{k})_{\ k=0} \overset{i}{}$$

$$= \underbrace{\frac{Hw}{p\overline{2}}}_{O(d\ ^{1=2})} + \underbrace{\frac{1}{2}[c_3C_3(w;w) + c_4wC_2(w;w)]}_{O(d\ ^{1})} + \underbrace{\frac{1}{6}[]}_{O(d\ ^{3=2})} + ::: \qquad (1)$$

where we note that $C_2 = E_x[r^2 f^{?}(x)] = H$. We emphasize that because w is a unit vector, its inner product with any fixed unit vector is of order $d\ ^{1=2}$ so temporarily ignoring factors of r, $C_{k+1}(w^{k}); C_k(w^{k}) = O(d\ ^{\frac{k}{2}})$. Therefore Equation (1) is an asymptotic series in $d\ ^{1=2}$. As k increases, each term in Equation (1) reveals more information about $f^{?}$. However, this information is also better hidden. A standard concentration argument shows that extracting information from the C_k term in this series requires $n\ \ d^k$ samples. This paper focuses on the first term in this expansion, $\frac{Hw}{p\overline{2}}$, which requires $n\ \ d^2$ samples to isolate. We directly truncate this series expansion:

Lemma 4 With high probability over the random initialization, r_{w_j}

$$L_D() = \quad 2a_j \frac{H}{2} + O\frac{w_j}{p\overline{d}}^{r} : \sim \frac{p}{}$$

Note that the remainder term, of order $d^{\ 1}$, contains all higher order terms in the series expansion.

Recall that $H = E_{x D}[r^2 f^?(x)]$ is the average Hessian of $f^?$ with respect to D. Because $f^?$ depends only on the subspace $S^?$, this implies that up to higher order terms, the population gradient at initialization already points each feature vector w_j towards the principal subspace $S^?$. In addition, Assumption 2 guarantees that the gradients at initialization span the principal subspace $S^?$.

However, it is also important to note that the population gradient is bounded by $r_{w_j} L_D() = O(d^{\ 1=2})$ and we only have access to the empirical gradient $r_{w_j} L()$. As mentioned above, ex-tracting the necessary subspace information from $r_{w_j} L_D()$ to learn $f^?$ therefore requires $n \,\&\, d^2$ samples, which is the dominant term in our final sample complexity result.

Once we show that the gradient at initialization contains all the relevant features, we note that after the first step of gradient descent,

$$W^{(1)} = W^{(0)} \quad {}_1[r_W L(^{(0)}) + {}_1{}^1 W] = \quad {}_1 r_W L(^{(0)}):$$

After the first step, the model therefore resembles a random feature model with random features $fHwg_{w2S^{d\ 1}} \quad S^?$. Previous results have shown that in these linearized regimes, e.g. random feature models/NTK, learning degree p polynomials requires $n \,\&\, d^{\,p}$ samples and width $m \,\&\, d^{\,p}$. As our "random features" are now constrained to the hidden subspace $S^?$, which has dimension r, we should expect that our sample complexity improves to $n \,\&\, r^p$.

The remainder of Algorithm 1 runs ridge regression on the network head a with fixed features $x \;! \quad (W^{(1)}x + b)$. We can directly analyze the generalization of this algorithm using standard techniques from Rademacher complexity. In particular, a high level sketch of the remainder of the proof goes as follows:

1. (Section A.2): We use the features from Lemma 4 to construct a vector $a^? \ 2 \ R^m$ such that
$$L(a^?; W^{(1)}; b) \quad 1 \qquad \text{and} \qquad ka^?k = Q\,\frac{r^{p2p}}{\cancel{p}\ m}:$$
2. (Section A.3): We show the equivalence between ridge regression and norm constrained linear regression implies the existence of $\ > \ 0$ such that the Tth iterate $a^{(T)}$ satisfies
$$L(a^{(T)}; W^{(1)}; b) \quad 1 \qquad \text{and} \qquad ka^{(T)}k \quad ka^?k:$$
3. (Section A.3): A standard Rademacher generalization bound for two layer neural networks bounds the population risk $E_{x;y}\, jf_{(T)}(x) \quad yj$ by the empirical risk $\frac{1}{n} P_{i=1}^{n} jf_{(T)}(x_i) \quad y_i j$ and $ka^{(T)}k$ which are small from step 2.

5.2. Proof of Theorem 2

Statistical query learners are a family of learners that can query values $q(x;y)$ and receive outputs $\hat{q}$ with $j\hat{q} \quad E_{x;y}[q(x;y)]j \quad$ where denotes the query tolerance (Goel et al., 2020; Diakonikolas et al., 2020). An important class of statistical query learners is that of correlational/inner product statistical queries (CSQ) of the form $q(x;y) = yh(x)$. This includes a wide class of algorithms including gradient descent with square loss. For example from Section 5.1, for a two layer neural network we have

$$r_{w_j} L_D() = E_{x;y}[yh(x)] \quad \text{where} \quad h(x) = \quad 2a_j x 1_{w_j x + b_j 0}:$$

In order to prove Theorem 2, we must construct a function class F_p such that inner product queries of the form $E_{x;y}[yh(x)]$ provide little to no information about the target function. The standard approach is to construct a function class with small pairwise correlations, i.e. for $f = g 2 F_p$, $jE_x[f(x)g(x)]j$ (Goel et al., 2020; Diakonikolas et al., 2020). The number of functions in the function class F_p and the size of the pairwise correlations directly imply a correlational statistical query lower bound:

Lemma 5 (Modified from Theorem 2 in Szőrényi (2009)) Let F be a class of functions and D be a data distribution such that

$$E_{xD}[f(x)^2] = 1 \quad \text{and} \quad jE_{xD}[f(x)g(x)]j \qquad 8f = g 2 F:$$

Then any correlational statistical query learner requires at least $\frac{jFj(^2\ )}{2}$ queries of tolerance to output a function in F with $L^2(D)$ loss at most 2 2.

To construct F_p, we begin by showing that there are a large number of approximately orthogonal unit vectors in $S^{d\ 1}$:

Lemma 6 There exists an absolute constant c such that for any > 0, there exists a set S of $\frac{1}{2}e^{c^2d}$ unit vectors such that for any $v; w 2 S$ such that $v = w$, we have $jv\ wj$.

The proof bounds the probability that randomly sampled unit vectors have a large inner product and existence then follows from the probabalistic method. Therefore for any m, we can find m unit vectors in R^d such that their pairwise inner products are all bounded by $d^{1=2}\sqrt{\log m}$. We combine this with the fact that if $f_u(x) = \frac{He_k(u\ x)}{\sqrt{k!}}$ where He_k denotes the kth Hermite polynomial,

$$E_{xD}[f_u(x)f_v(x)] = (u\ v)^k:$$

Therefore $ju\ vj \quad d^{1=2}\sqrt{\log m}$ implies $jE_{xD}[f_u(x)f_v(x)]j \quad d^{k=2}(\log m)^{k=2}$. Theorem 2 then directly follows from Lemma 5 (see Appendix D for a more detailed proof).

6. Experiments

6.1. Sample Complexity

In this section we present a toy example that clearly demonstrates the gap between kernel methods and gradient descent on two layer networks. For $u 2 S^{d\ 1}$, consider the target function

$$f_u^?(x) = g(u\ x) \quad \text{where} \quad g(x) = \frac{He_2(x)}{2} + \frac{He_p(x)}{\sqrt{2p!}}; \tag{2}$$

which satisfies $E_{xD}[f_u^?(x)^2] = 1$. Note that $f^?$ only depends on the projection of x onto a single relevant direction, u. We show in Section 5.1 that gradient descent is capable of isolating the subspace spanned by u and then fitting a one dimensional random feature model to g, and that this entire process requires $n \ d^2$ samples to generalize.

On the other hand, existing works Ghorbani et al. (2019b, 2020) have shown that $n \ d^p$ samples are strictly necessary in order to learn $f^?$ in the NTK or random features regime. The theory predicts that with $n < d^2$ samples, kernel regression will return the 0 predictor and with $d^2 < n < d^p$ samples, kernel regression will return $\frac{1}{2}He_2(u\ x)$, incurring a $L^2(D)$ loss of $\frac{1}{2}$. 1

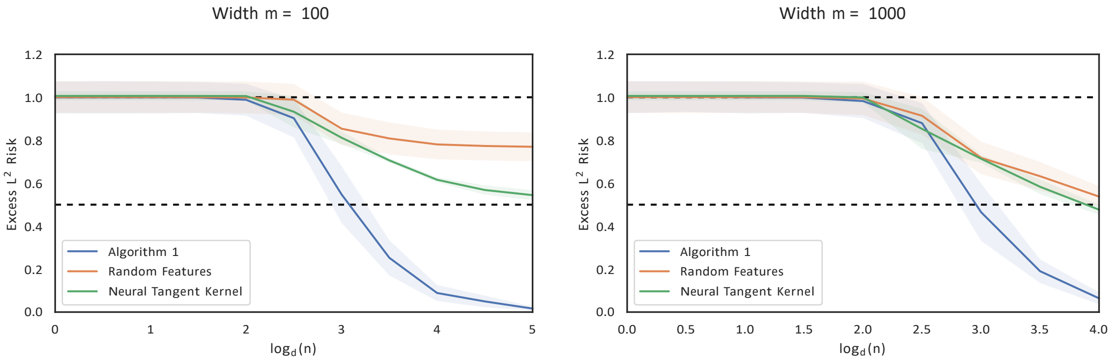

Figure 1: Sample Complexity: The x axis plots $\log_d(n)$ and the y axis plots the excess risk in $L^2(D)$ for each of three methods: Algorithm 1, random features, and the neural tangent kernel. The top dashed horizontal line is at $L^2 = 1$, which corresponds to outputting the zero predictor. The middle horizontal line is at $L^2 = \frac{1}{2}$ and corresponds to learning the optimal quadratic predictor $\frac{1}{2}He_2(x_1)$. Due to its improved sample efficiency, Algorithm 1 easily achieves near zero excess risk despite the relatively high degree of $f^?$, while random features and the neural tangent kernel are only able to learn the optimal quadratic predictor. See Section 6 for additional experimental details.

We empirically verify these predictions. We take $d = 10$ and $p = 4$ and consider the function $f^?_{e_1}(x) = \frac{e_2(x_1)}{H\ \ 2} + \frac{He_4(x_1)}{4\ \ 3}$. We use label noise $^2 = 1$ and attempt to learn $f^?$ using Algorithm 1, a random feature model, and a linearized NTK model. All experiments are conducted on a two layer neural network with widths $m = 100$ and $m = 1000$. For each value of n, the weight decay parameter is tuned on a holdout set of size 10^5 and test accuracies are reported over a separate test set of size 10^5. Errors bars reflect the mean and standard deviation over 10 random seeds.

We note that while Algorithm 1 easily converged to vanishing excess risk, even at width $m = 100$, both the random features model and the neural tangent kernel model only managed to fit the quadratic term $\frac{1}{2}He_2(u\ \ x)$, as predicted by the theory in Ghorbani et al. (2019b, 2020).

The key to learning a function of the form $f^?_u$ is to use the fact that the $\frac{1}{2}He_2(u\ \ x)$ component of $f^?_u$ gives enough information to identify u. Afterwards, any random feature or kernel method can efficiently fit any sufficiently smooth univariate function $g : \mathbb{R}\ !\ \ \mathbb{R}$ ontop of $u\ \ x$. Our analysis in Section 5.1 shows that this is exactly the way that Algorithm 1 learns $f^?_u$ and this is reflected by the steep and sudden drop from trivial risk ($L^2 = 1$) to vanishing excess risk without plateauing at $L^2 = 0{:}5$ in Figure 1.

6.2. Transfer Learning

The proof of Theorem 1 involves showing that Algorithm 1 learns features corresponding to $S^?$ (see Section 5.1) and the proof of Theorem 3 shows that this implies efficient transfer learning. We again verify this empirically. We consider the function:

$$f^?_{\text{target}}(x) = g_{\text{target}}(u\ \ x) \quad \text{where} \quad g_{\text{target}} = \frac{He_p(x)}{p\overline{p!}}:$$

Note that this was exactly the hard example in Theorem 2 that was unlearnable without $n \& d^{\frac{p}{2}}$ samples by a correlational statistical query learner (and in particular, gradient-based learners).

We pretrain with n samples on the $f^?(x)$ from Section 6.1, then train the output layer using N samples from $f^?_{target}$. As in Section 6.1, we use a label noise strength of $^2 = 1$. We pick $p = 3$ so that random feature methods or the neural tangent kernel will require at least $n \& d^3$ samples to learn $f^?$.

We note that in Figure 2, when $n = d^0; d^1$, fine tuning on N target samples gives trivial risk until $N \& d^3$, which is to be expected of a kernel method with no prior information. However, for n d^2 pretraining samples, we can fine tune on just $N = O(1)$ target samples to reach nontrivial loss and the loss decays rapidly as a function of N. This experiment therefore fully supports the conclusion of Theorem 3.

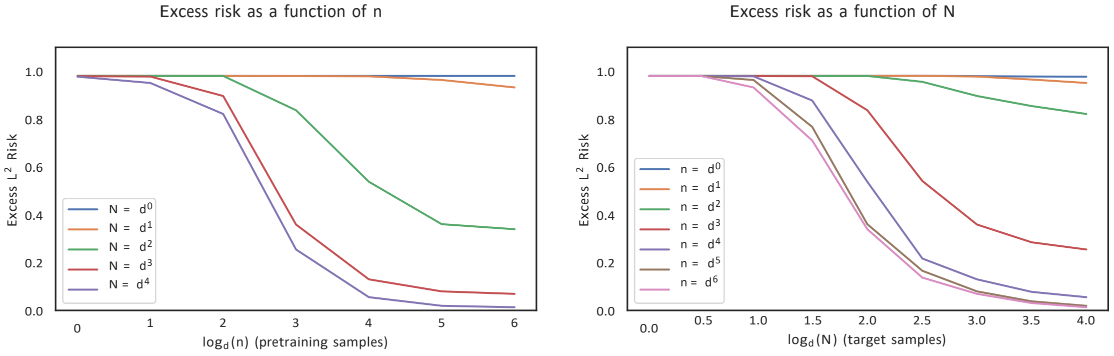

Figure 2: Transfer Learning: The x axes plot $\log_d(n)$ and $\log_d(N)$ respectively. We note that with little pretraining ($\log_d(n) = 0; 1$), Algorithm 1 is unable to extract a robust representation that enables transfer. For n d^2, we observe that finetuning the representation from Algorithm 1 gives nontrivial loss even for $N = O(1)$, as predicted by Theorem 3. See Section 6 for additional experimental details.

7. Discussion and Future Work

In this work we provide a clear separation between gradient-based training and kernel methods. We show that there is a large family of degree p polynomials which are efficiently learnable by gradient descent with n d^2 samples, in contrast to the lower bound of d^p for random feature/NTK analysis. The main idea driving both our sample complexity result (Theorem 1) and our transfer learning result (Theorem 3) is that gradient descent learns useful representations of the data.

One promising direction for future work is tightening the dimension dependence of our upper bound. In particular, our n d^2 sample complexity is driven by the difficult in learning from a degree 2 Hermite polynomial. However, our lower bound for such functions (Theorem 2) only rules out learning with n d samples. In this situation the lower bound is tight as Chen et al. (2020b) show that sparse degree 2 polynomials can be efficiently learned with n d samples.

Another promising direction from future work is generalizing our result to the situation in which the hidden layer and the output layer are trained together. This introduces dependencies between the hidden and output layers which are difficult to control. However, such analysis may lead to a better understanding of learning order and inductive bias in deep learning.

Acknowledgments

AD acknowledges support from a NSF Graduate Research Fellowship. JDL and AD acknowledge support of the ARO under MURI Award W911NF-11-1-0304, the Sloan Research Fellowship, NSF CCF 2002272, NSF IIS 2107304, ONR Young Investigator Award, and NSF-CAREER under award #2144994. MS is supported by the Packard Fellowship in Science and Engineering, a Sloan Fellowship in Mathematics, an NSF-CAREER under award #1846369, DARPA Learning with Less Labels (LwLL) and FastNICS programs, and NSF-CIF awards #1813877 and #2008443.

References

Emmanuel Abbe, Enric Boix-Adsera, and Theodor Misiakiewicz. The merged-staircase property: a necessary and nearly sufficient condition for sgd learning of sparse functions on two-layer neural networks. arXiv preprint arXiv:2202.08658, 2022.

Zeyuan Allen-Zhu and Yuanzhi Li. What can resnet learn efficiently, going beyond kernels? arXiv preprint arXiv:1905.10337, 2019.

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 Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In Advances in neural information processing systems, pages 6155–6166, 2019.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. arXiv preprint arXiv:1904.11955, 2019a.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. arXiv preprint arXiv:1901.08584, 2019b.

Yu Bai and Jason D Lee. Beyond linearization: On quadratic and higher-order approximation of wide neural networks. International Conference on Learning Representations (ICLR), 2020.

Yu Bai, Ben Krause, Huan Wang, Caiming Xiong, and Richard Socher. Taylorized training: Towards better approximation of neural network training at finite width. arXiv preprint arXiv:2002.04010, 2020.

Yuan Cao and Quanquan Gu. Generalization error bounds of gradient descent for learning overparameterized deep relu networks. arXiv preprint arXiv:1902.01384, 2019.

Minshuo Chen, Yu Bai, J. Lee, Tuo Zhao, Huan Wang, Caiming Xiong, and Richard Socher. Towards understanding hierarchical learning: Benefits of neural representations. ArXiv, abs/2006.13436, 2020a.

Minshuo Chen, Yu Bai, Jason D Lee, Tuo Zhao, Huan Wang, Caiming Xiong, and Richard Socher. Towards understanding hierarchical learning: Benefits of neural representations. Neural Information Processing Systems (NeurIPS), 2020b.

Sitan Chen and Raghu Meka. Learning polynomials of few relevant dimensions. arXiv preprint arXiv:2004.13748, 2020.

Lenaic Chizat and Francis Bach. A note on lazy training in supervised differentiable programming. arXiv preprint arXiv:1812.07956, 8, 2018a.

Le̍naïc Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 31, pages 3036–3046. Curran Associates, Inc., 2018b. URL https://proceedings.neurips.cc/paper/2018/file/a1afc58c6ca9540d057299ec3016d726-Paper.pdf.

Le̍naïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. Advances in Neural Information Processing Systems, 32:2937–2947, 2019.

Amit Daniely and Eran Malach. Learning parities with neural networks. Advances in Neural Information Processing Systems, 33:20356–20365, 2020.

Ilias Diakonikolas, Daniel M Kane, Vasilis Kontonis, and Nikos Zarifis. Algorithms and sq lower bounds for pac learning one-hidden-layer relu networks. In Conference on Learning Theory, pages 1514–1539, 2020.

Simon Du and Jason Lee. On the power of over-parametrization in neural networks with quadratic activation. In International conference on machine learning, pages 1329–1338. PMLR, 2018.

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, 2018a.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. arXiv preprint arXiv:1810.02054, 2018b.

Simon S Du, Jason D Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. International Conference on Machine Learning (ICML), 2019a.

Simon S Du, Jason D Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In International Conference on Machine Learning, pages 1675–1685, 2019b.

Ethan Dyer and Guy Gur-Ari. Asymptotics of wide networks from feynman diagrams. arXiv preprint arXiv:1909.11304, 2019.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Limitations of lazy training of two-layers neural network. In Advances in Neural Information Processing Systems, pages 9108–9118, 2019a.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Linearized two-layers neural networks in high dimension. arXiv preprint arXiv:1904.12191, 2019b.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. When do neural networks outperform kernel methods? arXiv preprint arXiv:2006.13409, 2020.

S. Goel, S. Karmalkar, and A. Klivans. Time/accuracy tradeoffs for learning a relu with respect to gaussian marginals. In NeurIPS, 2019.

Surbhi Goel, Varun Kanade, Adam Klivans, and Justin Thaler. Reliably learning the relu in polynomial time. In Conference on Learning Theory, pages 1004–1042, 2017.

Surbhi Goel, Aravind Gollakota, Zhihan Jin, Sushrut Karmalkar, and Adam Klivans. Superpolynomial lower bounds for learning one-layer neural networks using gradient descent. arXiv preprint arXiv:2006.12011, 2020.

Friedrich Gotze, Holger Sambale, and Arthur Sinulis. Concentration inequalities for polynomials in alpha-sub-exponential random variables. arXiv: Probability, 2019.

Jiaoyang Huang and Horng-Tzer Yau. Dynamics of deep neural networks and neural tangent hierarchy. arXiv preprint arXiv:1909.08156, 2019.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In Advances in neural information processing systems, pages 8571–8580, 2018.

Adel Javanmard, Marco Mondelli, Andrea Montanari, et al. Analysis of a two-layer neural network via displacement convexity. Ann. Statist., 48(6):3619–3642, 2020.

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 Advances in neural information processing systems, pages 8570–8581, 2019.

Y. Li, T. Ma, and H. Zhang. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In Conference On Learning Theory, pages 2–47, 2018.

Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. In Advances in Neural Information Processing Systems, pages 8157–8166, 2018.

Yuanzhi Li, Tengyu Ma, and Hongyang R Zhang. Learning over-parametrized two-layer relu neural networks beyond ntk. arXiv preprint arXiv:2007.04596, 2020.

Shengchao Liu, Dimitris Papailiopoulos, and Dimitris Achlioptas. Bad global minima exist and sgd can reach them. Advances in Neural Information Processing Systems, 33:8543–8552, 2020.

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. ISSN 0027-8424. doi: 10.1073/pnas.1806579115. URL https://www.pnas.org/content/115/33/E7665.

Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In Conference on Learning Theory, pages 2388–2464. PMLR, 2019.

Samet Oymak and Mahdi Soltanolkotabi. Overparameterized nonlinear learning: Gradient descent takes the shortest path? In International Conference on Machine Learning, pages 4951–4960. PMLR, 2019.

Samet Oymak and Mahdi Soltanolkotabi. Towards moderate overparameterization: global convergence guarantees for training shallow neural networks. IEEE Journal on Selected Areas in Information Theory, 2020.

Samet Oymak, Zalan Fabian, Mingchen Li, and Mahdi Soltanolkotabi. Generalization guarantees for neural networks via harnessing the low-rank structure of the jacobian. arXiv preprint arXiv:1906.05392, 2019.

Samet Oymak, Mingchen Li, and Mahdi Soltanolkotabi. Generalization guarantees for neural architecture search with train-validation split. In International Conference on Machine Learning, pages 8291–8301. PMLR, 2021.

Giuseppe Da Prato and Luciano Tubaro. Wick powers in stochastic pdes: an introduction. 2007.

Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks: a central limit theorem. Stochastic Processes Appl., 130(3):1820–1852, 2020. ISSN 0304-4149.

Mahdi Soltanolkotabi. Learning relus via gradient descent. In Advances in neural information processing systems, pages 2007–2017, 2017.

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.

Dominik Stöger and Mahdi Soltanolkotabi. Small random initialization is akin to spectral learning: Optimization and generalization guarantees for overparameterized low-rank matrix reconstruction. Advances in Neural Information Processing Systems, 34, 2021.

Balázs Szörényi. Characterizing statistical query learning: Simplified notions and proofs. In ALT, 2009.

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

Xiang Wang, Chenwei Wu, Jason D Lee, Tengyu Ma, and Rong Ge. Beyond lazy training for overparameterized tensor decomposition. Advances in Neural Information Processing Systems, 33, 2020.

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.

Blake Woodworth, Suriya Genesekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Kernel and deep regimes in overparametrized models. In Conference on Learning Theory (COLT), 2019.

Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. arXiv preprint arXiv:2002.09277, 2020.

Gilad Yehudai and Ohad Shamir. On the power and limitations of random features for understanding neural networks. CoRR, abs/1904.00687, 2019a. URL http://arxiv.org/abs/1904.00687.

Gilad Yehudai and Ohad Shamir. On the power and limitations of random features for understanding neural networks. arXiv preprint arXiv:1904.00687, 2019b.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530, 2016.

Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Stochastic gradient descent optimizes over-parameterized deep relu networks. arXiv preprint arXiv:1811.08888, 2018.

Appendix A. Proofs

We define = C log(nmd) for a sufficiently large constant C. Throughout the appendix we will use e to track failure probabilities of various lemmas and theorems.

Definition 7 (High probability events) We say that an event A happens with high probability if it happens with probability at least 1 poly(n; m; d)e where poly(n; m; d) does not depend on C.

Note that high probability events are closed under taking union bounds over sets of size poly(n; m; d). We will assume throughout that cd for a sufficiently small absolute constant c.

The following lemma bounds kx_ik and is a direct corollary of Lemma 27:

Lemma 8 With high probability, kx_ik^2 2 $\frac{d}{2}$; 2d for i = 1; : : : ; n.

All remaining proofs will be conditioned on this high probability event.

A.1. Hermite Expansions

A.1.1. HERMITE EXPANSION OF

Let (x) := ReLU(x) = max(0; x). Then the Hermite expansion of (x) is

$$(x) = \frac{1}{p\ 2} + \frac{x}{2} + \frac{1}{p\ 2} \frac{1}{X} \frac{(\ \ 1)^{k\ \ 1}}{k!2^k(2k\ \ \ 1)} He_{2k}(x):$$

$$_{k1}$$

Let c_k denote the Hermite coefficients of , i.e. (x) = $\mathrm{P}_{k0} \frac{c_k}{k!} He_k(x)$. Note that

$$^0(x) = \mathrm{X}_{k0} \frac{c_{k+1}}{k!} He_k(x) = \frac{1}{2} + \frac{1}{p\ 2} \mathrm{X}_{k0} \frac{(\ \ \ 1)^k}{k!2^k(2k+1)} He_{2k+1}(x):$$

A.1.2. HERMITE EXPANSION OF $f^?$

Let the Hermite expansion of $f^?$ be

$$f^?(x) = \mathrm{X}^p_{k=0} \frac{hC_k; He_k(x)i}{k!}$$

where C_k 2 (R^d) ^{k} is the symmetric k-tensor defined by $C_k := E_x[r^k f^?(x)]$:

Note that

$$r f^?(x) = \mathrm{X}^{p\ \ 1}_{k=0} \frac{C_{k+1}(He_k(x))}{k!}\ 2\ R^d:$$

Lemma 9 (Parseval's Identity)

$$1 = E_x[f^?(x)^2] = \mathrm{X}^p_{k=0} \frac{kC_kk_F^2}{k!}:$$

Note that as an immediate consequence of Lemma 9, $kC_kk_F^2$ k!. In addition, Assumption 1 guarantees that C_k x ^{k} = C_k $(^?x)$ ^{k}.

A.1.3. CONCENTRATING ;

Lemma 10 Let $= \frac{1}{n}\sum_{i=1}^{n} y_i$ and $= \frac{1}{n}\sum_{i=1}^{n} y_i x_i$. Then, with high probability,

$$j \quad C_0 j \lesssim \sqrt{\frac{p}{n}} \quad \text{and} \quad kC_1 \quad k \lesssim \sqrt{\frac{d^{p+1}}{n}}:$$

Proof Let $F(x_1; :::; x_n) = \frac{1}{n}\sum_{i=1}^{n} f^?(x_i) \quad C_0$. Note that

$$E_{x_1;:::;x_n}[F(x)^2] = \frac{1}{n}\mathrm{Var}(f^?(x)) \quad \frac{1}{n}:$$

The bound on j C$_0$j therefore immediately follows from Lemma 30 applied to F. The bound on k C$_1$k is a special case of Lemma 32 with (x) = x. ■

A.1.4. HERMITE EXPANDING THE FEATURES

Note that by the scale invariance of (x) = ReLU(x), Algorithm 1 does not depend on kw_jk for j = 1; :::; m. Therefore we can assume WLOG that kw_jk = 1 for j = 1; :::; m and w_j Unif(S^{d} ^{1}). For the remainder of the appendix we will assume that kw_jk = 1.

Definition 11 We define $\hat{f}^?(x) := f^?(x) \quad x$.

The functions g(w) and $g_n(w)$ capture the features that can be learned after one step of gradient descent:

Definition 12 For kwk = 1, we define

$$g(w) := E_x[\hat{f}^?(x)x^0(w \ x)] \quad \text{and} \quad g_n(w) := \frac{1}{n}\sum_{i=1}^{n} f^?(x_i) \quad _i x_i{}^0(w \ x_i):$$

We note that $w_j^{(1)} = 2_1 a_j g_n(w_j)$ and $g(w) = E_x[g_n(w)]$. In fact, Corollary 34 shows that with probability at least 1 4ne ,

$$\sup_w kg(w) \quad g_n(w)k \lesssim \sqrt{\frac{d^{p+1}}{n}}:$$

Lemma 13 With high probability,

$$g(w) = \frac{Hw}{\sqrt{2}} + O\left(\sqrt{\frac{d^{p+1}}{n}} + \sqrt{\frac{r^2}{d^2}}\right):$$

Proof By Stein's lemma and the orthogonality of Hermite polynomials,

$$\begin{aligned} g(x) &= E_x[\hat{f}^?(x)x^0(w \ x)] \\ &= E_x[r\hat{f}^?(x)^0(w \ x) + w\hat{f}^?(x)^{00}(w \ x)] \\ &= \sum_{k)}^{p\ 1} \frac{c_{k+1} E_x[r_x^{k+1}\hat{f}^?(x)](w}{k!} + w \sum_{k=0}^{p} \frac{c_{k+2} E_x[r^k f^?(x)](w \ x)}{k!} \\ &= \frac{C_1}{2} + \frac{w(C_0 \quad)}{2} + \frac{C_2 w}{2} + \sum_{k=0} \frac{c_{2k}C_{2k}(w^{2k})}{2^{k}(2k)!} + w \sum_{k1} \frac{c_{2k+2}C_{2k}(w^{2k})}{2^{k2}(2k)!}: \end{aligned}$$

$(2k \quad 1)!$ $(2k \quad 1)$

Note that these sums are finite as $C_k = 0$ for $k > p$. Next, by Lemma 46 we have the high probability bounds,

$$\begin{aligned}\|C_{k+1}(w^{\otimes k})\|_F &\lesssim \sqrt{\frac{r^{\lfloor \frac{k}{2} \rfloor}}{d^k}} \lesssim \sqrt{\frac{r^3}{d^3}} &&\text{for } k \ge 3\\ \|wC_k(w^{\otimes k})\|_F &\lesssim \sqrt{\frac{r^{\lfloor \frac{k}{2} \rfloor}}{d^k}} \lesssim \sqrt{\frac{r^2}{d^2}} &&\text{for } k \ge 2.\end{aligned}$$

Applying these bounds term by term and using Lemma 10 to bound $|C_0|$ and $\|C_1\|$ gives the desired result. ∎

Corollary 14 With high probability,

$$g_n(w) = \frac{Hw}{\sqrt{2}} + O\left(\sqrt{\frac{d^{p+1}}{n}} + \sqrt{\frac{r^2}{d^2}}\right).$$

Corollary 15 With high probability,

$$\|g_n(w)\| \lesssim \sqrt{\frac{r^2}{d}} + \sqrt{\frac{d^{p+1}}{n}}.$$

Furthermore, it will become necessary to bound terms of the form $g_n(w) \cdot x_i$. Note that $g_n(w)$ and x_i are dependent random variables. The following lemma handles this dependence.

Lemma 16 Let $w \in S^{d-1}$ and assume $n \ge d^{2p}$. Then with high probability,

$$\max_{j \in [n]} \|g_n(w) \cdot x_j\| \lesssim \sqrt{\frac{r^3}{d}}.$$

Proof We can decompose

$$|g_n(w) \cdot x_j| \le |g(w) \cdot x_j| + |[g(w) - g_n(w)] \cdot x_j|.$$

For the first term, note that $g(w)$ and x_j are independent so $g(w) \cdot x_j \sim N(0, \|g(w)\|^2)$ so with high probability,

$$|g(w) \cdot x_i| \le \|g(w)\|\sqrt{2} \lesssim \sqrt{\frac{r^3}{d}} + \sqrt{\frac{d^{p+2}}{n}}.$$

Next,

$$\begin{aligned}&[g(w) - g_n(w)] \cdot x_i\\ &= x_j \cdot \left[\frac{1}{n}\sum_{i \ne j}\left[f^\star(x_i)x_i\sigma'(w \cdot x_i) - g(w)\right]\right] + \frac{1}{n}\left[f^\star(x_j)\|x_j\|^2\sigma'(w \cdot x_j) - g(w)\right].\end{aligned}$$

Note that in the first term, the x_j and the sum are independent. Therefore by Corollary 34 the first term is bounded with high probability by $O\left(\sqrt{\frac{d^{p+2}}{n}}\right)$. In addition, by Lemma 30, the second term is bounded by $O\left(\frac{d^{p/2}d}{n}\right)$ which completes the proof. ∎

A.2. Random Feature Approximation

A.2.1. UNIVARIATE RANDOM FEATURE APPROXIMATION

This section shows that after we reinitialize the biases we can use random features to transform the activation $(x) = \mathrm{ReLU}(x)$ into $(x) = x^p$ which is more natural for learning polynomials.

Lemma 17 Let $a \quad \mathrm{Unif}(\{\ 1;1\})$, and $b \quad \mathrm{Unif}([\ 1;1])$. Then for any $k \quad 0$ there exists $v_k(a;b)$ such that for $|x| \quad 1$,

$$E[v_k(a;b)(ax+b)] = x^k \qquad \text{and} \qquad \sup_{a;b} |v_k(a;b)| \lesssim 1:$$

Proof First, for $k = 0$ we can take $v_0(a;b) := 6b$. Then,

$$E[v_0(a;b)(ax+b)] = \frac{3}{2}\int^{1} b[(x+b)+(\ x+b)]db$$
$$= \frac{3}{2}\int_{x}^{1} b(x+b)db + \int_{x}^{1} b(\ x+b)db$$
$$= 1$$

and $\sup_{a;b} |v_0(a;b)| = 6$. Next, for $k = 1$ we can take $v_1(a;b) := 2a$. Then,

$$E[v_1(a;b)(ax+b)] = \frac{1}{2}\int^{1} [(x+b) \qquad (\ x+b)]db$$
$$= \frac{1}{2}\int_{x}^{1} (x+b)db \qquad \int_{x}^{1} (\ x+b)db$$
$$= x$$

and we have $\sup_{a;b} |v_1(a;b)| = 2$. Next, note that by integration by parts we have for any function f,

$$E[2(1 \quad a)f''(b)(ax+b)] = \int_{x}^{1} f''(b)(\ x+b)db$$
$$= f'(1)(\ x+1) \qquad \int_{x}^{1} f'(b)$$
$$= f(x) + f'(1)(\ x+1) \quad f(1)$$
$$= f(x) + [f'(1) \quad f(1)] \quad f'(1)x:$$

Therefore for $k \quad 2$ if $f(x) = x^k$ and

$$v_k(a;b) := 2(1 \quad a)f''(b) \quad [f'(1) \quad f(1)]v_0(a;b) + f'(1)v_1(a;b)$$

we have

$$E[v_k(a;b)(ax+b) = x^k \qquad \text{and} \qquad \sup_{a;b} |v_k(a;b)| \lesssim 1:$$

■

Corollary 18 Let a Unif(f 1; 1g), and b N(0; 1). Then for any k 0 there exists $v_k(a;b)$ such that for jxj 1,

$$E[v_k(a;b)(ax+b)] = x^k \quad \text{and} \quad \sup_{a;b} jv_k(a;b)j \;.\; 1:$$

Proof Let $\bar{v}_k$ be the function constructed in Lemma 17 and let

$$v_k(a;b) = 1_{jbj1} \frac{\bar{v}_k(a;b)}{2(b)}$$

where $(b) := \frac{e^{\frac{x^2}{2}}}{p\ 2}$ denotes the density of b. Then,

$$E_{a;b}[v_k(a;b)(ax+b)] = E_a \int v_k(a;b)(ax+b)(b)db\ b$$

$$= E_{a;b\,\mathrm{Unif}([\ 1;1])}[\bar{v}_k(a;b)(ax+b)] = x^k$$

and

$$\sup_{a;b} jv_k(a;b)j = \sup_{a;b2[\ 1;1]} \frac{v_k(a;b)}{2(b)} \;.\; 1:$$

■

A.2.2. MULTIVARIABLE RANDOM FEATURE APPROXIMATION

Definition 19 For kwk = 1, we define

$$r(w) := g_n(w) \quad \frac{Hw}{p\ 2}:$$

Corollary 14 shows that with high probability, kr(w)k . O $\tilde{}$ q $\frac{d}{n} + \frac{p\ r}{d^2}$. Recall that

Lemma 20 With high probability over the data $fx_ig_{i2[n]}$, we have for j 4p,

$$E_w\left[k\,r^?(w)k^j\right]^{1=j} \;.\; \sqrt{\frac{d^{p+1}}{n}} + \sqrt{\frac{r^2}{d^3}} : h$$

Proof We can decompose $r(w) = [g_n(w) \quad g(w)] + \left[g(w) \quad \frac{Hw}{p\ 2}\right]$ and note that

$$E_w\left[k^?r(w)k^j\right]^{1=j} \quad E_w\left[k^?[g_n(w) \quad g(w)]k^j\right]^{1=j} + E_w\left[{}^?g(w) \quad \frac{Hw}{p\ 2}{}^j\right]^{1=j}$$

$$E_w\left[{}^?\ g(w) \quad \frac{Hw}{p\ 2}{}^j\right]^{1=j} + O\left(\sqrt{\frac{d^{p+1}}{n}}\right):$$

Recall that

$$g(w) \quad \sigma\left(\frac{Hw}{\sqrt{2}}\right)$$

$$= \frac{C_1 \quad w(C_0 \quad w^{\otimes 2k-1})}{\sqrt{2}} + \sum_{k \geq 2} \frac{c_{2k+2} C_{2k}(w^{\otimes 2k})}{2} + \sum_{k \geq 2} \frac{c_{2k} C_{2k}(w^{\otimes 2k-1})}{(2k-1)!} + w \sum_{k \geq 1} \frac{c_{2k+2} C_{2k}(w^{\otimes 2k})}{(2k)!}:$$

Therefore,

$$\nabla^{?} g(w) \quad \frac{Hw}{\sqrt{2}} = \sum \frac{c_{2k} C_{2k}(w^{\otimes 2k-1})}{(2k-1)!} + \sum_{k \geq 1} \frac{c_{2k+2} C_{2k}(w^{\otimes 2k})}{(2k)!} w + O\left(\frac{?}{d^{p+1}}\right):$$

$$\sum_{k \geq 2} \qquad \qquad n$$

We can bound the jth moment term by term. We have by Corollary 39 and Lemma 43 that for $k \geq 2$,

$$\mathbb{E}_w\left[C_{2k}(w^{\otimes 2k-1})\right]^{1=j}_{j} \cdot \sqrt{\frac{r^{\lfloor \frac{2k-1}{2} \rfloor}}{d^{2k-1}}} \cdot \sqrt{\frac{r}{d^3}}$$

and for $k \geq 1$,

$$\mathbb{E}_w\left[\|\nabla^{?} w\| C_{2k}(w^{\otimes 2k})\right]^{1=j}_{j} \cdot \mathbb{E}_w\left[\|\nabla^{?} w\|^{2j}\right]^{\frac{1}{2j}} \mathbb{E}_w\left[C_{2k}(w^{\otimes 2k})^{2j}\right]^{\frac{1}{2j}}$$

$$\cdot \sqrt{\frac{r}{d}} \sqrt{\frac{r^k}{d^{2k}}}$$

$$\cdot \frac{r^2}{d^3}: \qquad \blacksquare$$

We can now show that the random features $g_n(w)$ are sufficiently expressive to allow us to efficiently represent any polynomial of degree p restricted to the principal subspace $S^?$.

Lemma 21 For any $k \leq p$, there exists an absolute constant C such that if $n \geq C d^2 r^{2p+1}$ and $d \geq C r^{3=2}$,

$$\text{Mat}\left[\mathbb{E}\left((\nabla^{?} g_n(w))^{\otimes 2k}\right)\right] \succeq \%(rd)^{?} \quad {}^{k}\,\Pi_{\text{Sym}^k(S^?)}$$

where $\Pi_{\text{Sym}^k(S^?)}$ denotes the orthogonal projection onto symmetric k tensors restricted to $S^?$.

Proof Note that because every vector in span $\text{Mat}\left[\mathbb{E}\left((\nabla^{?} g_n(w))^{\otimes 2k}\right)\right]$ is a vectorized symmetric k tensor, it suffices to show that

$$\mathbb{E}_w[(\nabla^{?} g_n(w))^{\otimes 2k}](T;T) \gtrsim (rd^2)^{k}$$

for all symmetric k tensor T with $\|T\|_F^2 = 1$. Recall that $g_n(w) = \frac{Hw}{\sqrt{2}} + r(w)$. Therefore by the binomial theorem,

$$\left\langle T; (\nabla^{?} g_n(w))^{\otimes 2k}\right\rangle \quad \frac{Hw}{\sqrt{2}}$$

$${}_{k}E = \overset{*}{}\,T\,;\,\not{p}_{2}$$

$${}_{k}{}^{+}$$

$$+\ (w)$$

where j(w)j . P $_{i=1}^{k}$ T (H w)

k i D F k?r(w)k^{i}: Therefore by 2Young's inequality, E 2 3 4 1 H w 5 *

k + 2

E T; (?g$_{n}$(w))

k E 2 T; p 2 E[(w)^{2}]:

Next by Cauchy-Schwarz,

E[(w) P. X^{k} r h 4 i h 4i E$_{w}$ kT ((H w)

k i)k$_{F}$ E$_{w}$ k?r(w)k

i = 1

X^{k} r 2 h 4i . E$_{w}$ T (H w)

k i F E$_{w}$ k?r(w)k :

i = 1

Let T^ be the symmetric k tensor defined by T^(v$_{1}$; : : : ; v$_{k}$) = T (H v$_{1}$; : : : ; H v$_{k}$). Then by Lemma 47,

E T ((H w) 1 ^

k i)^{2} min (H)^{2i} E TF(w 2 * + 3

k i)^{2}

d i H w

. k 2 E 4 T; p 2 2 5

min (H)^{2}

3 *

+ 2

= (r d^{2})^{i} E 4 T; p H w 2 k 5 :

Therefore,

2 *

k + 2 3 k 2 4 H w p+1 5 X 2 2 i dE$_{w}$[(w)] r. E

T; p (rd^{2}) + 3 : n d

i = 1

Because we assumed n Cd^{2}r^{2p+1} and d Cr$^{3=2}$ for a sufficiently large constant C, we have

2 * + 2 3

E$_{w}$[(w)^{2}] . 1 4 E 4 T; p H w 2 k 5 :

Combining everything gives

2 *

D k + 2 3 E 2 4 1 H w 5

E T; (?g(w))

k E 2 T;

* + 2

4 p 2 H w E[(w)^{2}] 2 5

3

1

$^{k}\ 4^{E}\quad T;\ \ ~~p~~_{2}$

$\&\ d\ \ {}_{k}T\ {}_{F}\hat{}\ \ {}^{2}$

$d\ \ {}^{k}{}_{min}(H)^{2k}$

$=\ (rd^{2})\ \ {}^{k}:$

■

Corollary 22 Assume n Cd^2r^{2p+1} and d $Cr^{3=2}$ for a sufficiently large constant C. Then for any k p and any symmetric k tensor T supported on $S^?$, there exists $z_T(w)$ such that

$$E_w[z_T(w)(g_n(w)\ \ x)^p] = hT; x$$

^{k}i and we have the bounds

$$E_w[z_T(w)^2] .\ \ (rd^2)^k kT k_F{}^2 \quad \text{and} \quad jz_T(w)j .\ \ (rd^2)^k kT k_F kg_n(w)k^k:$$

Proof Let

$$z_T(w) := \mathrm{Vec}(T)^T\ \mathrm{Mat}(E[g_n(w)$$

^{2k}])^y Vec($g_n(w)$

^{k}): Note that Vec(T) 2 span(Mat(E[$g(w)$

^{2k}])) by Lemma 21. Therefore,

$$E_w[z_T(w)(g_n(w)\ \ x)^k$$

$$= E_w\ \ h\ \ \mathrm{Vec}(T)^T\ \mathrm{Mat}(E[g_n(w)$$

$$^{2k}])^y\ \mathrm{Vec}(g(w)$$

$$^k)\ \mathrm{Vec}(g_n(w)$$

$$^k)^T\ \mathrm{Vec}(x$$

$$^k) = hT; x$$

$$^k i:$$

i

For the bounds on z we have

$$E_w[z_T(w)^2]$$

$$= E_w[\mathrm{Vec}(T)^T\ \mathrm{Mat}(E[g_n(w)$$

$$^{2k}])^y\ \mathrm{Vec}(g(w)$$

$$^k)\ \mathrm{Vec}(g_n(w)\ \ ^2$$

$$^k)^T\ \mathrm{Mat}(E[g(w)$$

$$^{2k}])^y\ \mathrm{Vec}(T)] = \mathrm{Vec}(T)^T\ \mathrm{Mat}(E[g_n(w)$$

$$^{2k}])^y\ \mathrm{Vec}(T)$$

$$.\ \ (rd^2)^k kT k_F$$

■

and

$$jz_T(w)j = \mathrm{Vec}(T)^T\ \mathrm{Mat}(E[g_n(w)$$

$$^{2k}])^y\ \mathrm{Vec}(g_n(w)$$

$$^k)$$

$$.\ \ (rd^2)^k kT k_F kg_n(w)k^k:$$

Lemma 23 Assume n Cd^2r^{2p+1} and d $Cr^{3=2}$ for a sufficiently large constant C. Let $_1 = q\ \frac{d}{C^{2=3}}$, let k p and let T be a k tensor. Then with high probability, there exists $h_T(a; w; b)$ such that if

$$f_{h_T}(x) := E_{a;w;b}[h_T(a;w;b)(w^{(1)} \; x + \; b)]$$

we have

$$\frac{1}{n} X^n (f_h(x_i) \quad hT; x_i \qquad \overline{} \, n$$

$$^p i)^2 \; . \quad 1_{\; i=1}$$

and the moment bounds

$$E_{w;a;b}[h_T(a;w;b)^2] \; . \quad r^{k2k3k} kT k_F \;\; ^2$$

$$\sup j h_T(a;w;b)j] \; . \quad r^{k2k6k} kT k \;\; :_F^w$$

Proof We define

$$h_T(a; w; b) := \frac{v_k(a; b)z_T(w)}{(2) \;\; k} 1_{1kg_n(w)k1} \prod^{n} 1_{jg_n(w)x_ij1} : \; i=1$$

where $v_k(a; b)$ and $z_T(w)$ are constructed in Corollary 18 and Corollary 22 respectively. Recall that $w^{(1)} = 2_1 ag_n(w)$. Then for $x \; 2 \; fx_1; : : : ; x_n g$,

$$f_{h_T}(x)$$

$$= \frac{1}{(2_1)^k} E_{a;w;b}[v_k(a; b)z_T(w)(2_1 ag_n(w) \;\; x + b)] \qquad = E_w \;\; z_T(w)$$

$$'' \qquad '' \qquad '' \qquad \#!\#\#$$

$$(g_n(w) \;\; x)^k + O \;\; jg_n(w) \;\; xj^k \;\; 1 \qquad 1_{1kg_n(w)k1} \prod_{i=1}^{n} 1_{jg_n(w)x_ij1} \qquad \#$$

$$''$$

$$= hT; x$$

$$^k i + \text{poly}(d) \;\; P_w[_1 kg_n(w)k \;\; 1] + \sum_{i=1}^{n} P_w[j2_1 g_n(w) \;\; x_i j \;\; 1]$$

$$= hT; x$$

$$^k i + \text{poly}(n; d)e \;\; \underline{1}$$

$$= hT; x$$

$$^k i + O \quad n$$

where the second to last line followed from Lemma 16. The first part of the lemma now follows from a union bound over $x_1; : : : ; x_n$. For the bounds on h, we have

$$E_{a;w;b}[h(a; w; b)^2]$$

$$= \frac{1}{(2_1)^{2k}} E_{a;w;b} \, v_k(a; b)^2 z_T(w)^2 \; .$$

$$^{2k}(rd^2)^k kT k^2$$

$$= r^{k2k3k}_1 kT k_F :^2 \qquad F$$

and

$$\sup_w jh(a; w; b)j = \sup_w \frac{v_k(a; b)z_T(w)}{(2) \;\; k} 1_{1kg_n(w)k1}$$

$$. \;\; _1{}^k (rd^2)^k kg_n(w)k^k kT k_F =$$

$$^{2k}(r_1 d^2)^k kT k_F$$

$$. \;\; r^{k2k6k} kT k_F :$$

■

Corollary 24 Assume $n \;\; Cd^2 r^{2p+1}$ and $d \;\; Cr^{3=2}$ for a sufficiently large constant C and let $_1 = q \frac{d}{3}$. Then with high probability, there exists $h(a; w; b)$ such that if

$$f_h(x) := E_{a;w;b}[h(a; w; b)(w^{(1)} \;\; x + b)]$$

we have

$$\frac{1}{n}\sum_{i=1}^{n}(f_h(x_i) - f^\star(x_i))^2 \lesssim \frac{1}{n}$$

and the moment bounds

$$\begin{aligned} E_{w;a;b}[h_T(a;w;b)^2] &\lesssim r^{p2p3p} \\ \sup_w |h_T(a;w;b)| &\lesssim r^{p2p6p}: \end{aligned}$$

Proof We know from Lemma 35 that

$$f^\star(x) = \sum_{k \le p} \langle T_k; x^{\otimes k}\rangle$$

with $\|T_k\|_F \lesssim r^{\frac{p-k}{4}}$. Let

$$h(a;w;b) := \sum_{k \le p} h_{T_k}(a;w;b):$$

Then $\frac{1}{n}\sum_{i=1}^{n}(f_h(x_i) - f^\star(x_i))^2 \lesssim \frac{1}{n}$ is immediate from Lemma 23 and

$$E_{a;w;b}[h(a;w;b)^2] \lesssim \sum_{k \le p} E_{a;w;b}[h_{T_k}(a;w;b)^2] \lesssim \sum_{k \le p} r^{k2k3k} r^{\frac{p-k}{2}} \lesssim r^{p2p3p}:$$

and

$$\sup_{a;w;b} |h(a;w;b)| \le \sum_{k \le p} \sup_{a;w;b} |h_{T_k}(a;w;b)| \lesssim \sum_{k \le p} r^{k2k6k} r^{\frac{p-k}{2}} \lesssim r^{p2p6p}:$$

complete the proof. ■

Lemma 25 Assume $n \ge C d^2 r^{2p+1}$, $d \ge C r^{3=2}$, and $m \ge r^{p2p6p+1}$ for a sufficiently large constant C and let $\eta_1 = \sqrt{\frac{d}{3}}$. Then with high probability, there exists $a^\star \in \mathbb{R}^m$ such that if $\theta^\star = (a^\star; W^{(1)}; b^{(1)})$,

$$\frac{1}{n}\sum_{i=1}^{n}(f_{\theta^\star}(x_i) - f^\star(x_i))^2 \lesssim \frac{1}{n} + \frac{r^{p2p6p+1}}{m} \quad \text{and} \quad \|a^\star\|^2 \lesssim \frac{r^{p2p6p}}{m}:$$

Proof Let $a_j^\star := \frac{1}{m} h(a_j; w_j; b_j)$ where h is the function constructed in Lemma 24. Then,

$$\begin{aligned} E_{i\in[n]}[(f_{\theta^\star}(x_i) - f^\star(x_i))^2] &\lesssim E_{i\in[n]}[(f_{\theta^\star}(x_i) - f_h(x_i))^2] + E_{i\in[n]}[(f_h(x_i) - f^\star(x_i))^2] \\ &= E_{i\in[n]}[(f_{\theta^\star}(x_i) - f_h(x_i))^2] + \frac{1}{n}: \end{aligned}$$

For $j \quad m=2$, let

$$Z_j(x) := a_j^?(w_j^{(1)}\ x + b_j) + a_m\ {}^?_j(w^{(1)}\ {}_{jm}x + b_m\ \ {}_j):$$

Note that

$$f_?(x) \quad f_h(x) = \sum_{j\ 2\ \underline{m}} (Z_j(x) \quad E[Z_j(x)])$$

and the $Z_j(x)$ are all i.i.d.. Let

$$\overline{Z_j}(x) := Z_j(x)1_{w_j^{(1)}x1}1_{w^{(1)}\ {}_{j\,m}x1}:$$

Then with probability $1 \quad \mathrm{poly}(n;m;d)e$ we have that $Z_j(x_i) = \overline{Z_j}(x_i)$ for $i = 1;:::;n$. Therefore,

$$f_?(x) \quad f_h(x) = \sum_{2\ \ \underline{m}} Z_j(x) \quad E\,\overline{Z}_j(x) + \frac{m}{2}E\,Z_j\overline{(x)} \quad E\,Z_j(x):\ j$$

For the first term, by Bernstein's inequality we have with probability at least $1 \quad 2e \ ,$ $\sum$

$$\sum_{j\ 2} Z_{j\underline{m}}(x) \quad \overline{}\ E[Z_j(x)] \ . \ \sqrt{\frac{E_{a;w;b}[h(a;w;b)^2]}{m}} + \frac{r^{p2p3p}}{m}. \ \sqrt{\frac{r^{p2p6p+1}}{m}}$$

The second term is bounded as in the proof of Lemma 23 by $\mathrm{poly}(n;d)e \quad \frac{1}{m}$ because $P[w^{(1)}\ x > 1]\ e$ from the choice of $_1$. Therefore for any fixed x, with high probability we have

$$f_?(x) = f^?(x) + O\left(\frac{1}{m} + \sqrt{\frac{r^{p2p6p+1}}{}}\, n\right)$$

and the first part of the lemma follows from a union bound.

We will now turn to the bound on $ka^?k^2$. Let $z_i = (a_i^?)^2 + (a^?_m\ {}_i)^2$. Note that $fz_ig_{im=2}$ are positive, i.i.d., and bounded by $O(m\ \ ^2r^{2p4p12p})$. In addition, they have expectation $O(m\ \ ^2r^{p2p3p})$. Therefore by Popoviciu's inequality they have variance bounded by

$$O\ m\ \ ^1r^{p2p3p}m\ \ ^2r^{2p4p12p} = O\ m\ \ ^3r^{3p6p15p}:$$

Therefore by Bernstein's inequality we have that with high probability,

$$ka^?k^2 = E[ka^?k^2] + O\left(\frac{1}{}\sqrt{\frac{r^{3p6p15p}}{m}} + \frac{r^{2p3p6p}}{m^2}\right) . \ \frac{r^{p2p6p}\,m}{m}:$$

■

A.3. Proof of Theorem 1

We will define

$$\tilde{L}()() := \frac{1}{n} \sum_{i=1}^{n} (f(x_i) \qquad f^?(x_i))^2 :$$

to be the empirical L^2 losses with respect to the true labels (recall $y_i = f^?(x_i) + {}_i, {}_i$ f ; g).

Lemma 26 Assume n $C d^2 r^{2p+1}$ and d $C r^{3=2}$ for a sufficiently large constant C and let ${}_1 = \sqrt{\frac{d}{3}}$. Let $a^?$ be the vector constructed in the proof of Lemma 25 and let $= (a^?; W^{(1)}; b^{(1)})$. Then with high probability,

$$L() \qquad \&^2 \, . \, \frac{r^{p \quad 2p \;\; 6p+1}}{m} + \sqrt{\frac{r}{n}} :$$

Proof Let ${}_i = f(x_i) \qquad f^?(x_i)$. Then,

$$\frac{1}{n} k + k_2^2 = \frac{1}{n} kk_2^2 + 2h; i + kk^2 :$$

First, by Hoeffding's inequality, we have with high probability,

$$\frac{kk^2}{n} \quad \&^2 + \frac{C\&^2}{\sqrt{p}\, n} = \&^2 + O \quad \sqrt{\frac{p}{n}} \quad :$$

Similarly, by Hoeffding's inequality we have with high probability, $\frac{1}{n} h; i \quad \& \sqrt{\frac{2 \tilde{L}()}{n}} = O \frac{p}{n}$.

■

We are now ready to directly prove Theorem 1.

Proof [Proof of Theorem 1] Note that we can assume that there is an absolute constant C such that n $C d^2 r^{2p+1}$, and m $r^{p 2p 6p+1}$. Otherwise, we can simply take ! 1 and return the zero predictor.

From Lemma 26 we know that with high probability, there exists $a^?$ such that if $= (a^?; W^{(1)}; b^{(1)})$,

$$L() \qquad {}^2. \quad \frac{r^{p \quad 2p \;\; 6p+1}}{m} + \sqrt{\frac{r}{n}} :$$

and $ka^?k_2^2$. $\frac{r^{p\,2p\,6p+1}}{m}$: Therefore by equality of norm constrained linear regression and ridge regression, there exists > 0 such that if

$$a^{(1)} = \min_a L(a; W^{(1)}; b^{(1)}) + \frac{kak^2}{2} ;$$

$$L(a^{(1)}; W^{(1)}; b^{(1)}) \;\; L(a^?; W^{(1)}; b^{(1)}) \qquad \text{and} \quad ka_1k \;\; ka^?k:$$

Note that we can approximate $a^{(1)}$ by $a^{(T)}$ to within arbitrary accuracy within $T = (\tilde{}^{1} \quad ^{1})$ steps. Let

$$F = ff \ : \ kak_2 \quad ka^?k; kw_jk \quad 1g:$$

Then with high probability, $f_{(a^{(T)};W^{(1)};b^{(1)})}$ 2 F. In addition, from Lemma 49,

$$\sup_{f 2 F} \frac{1}{n} \sum_{i=1}^{n} jf(x_i) \quad y_ij \quad E_{x;y} jf(x) \quad yj \ . \ \sqrt{\frac{ka \ ?k^2 md}{n}} + \sqrt{\frac{}{n}}$$
$$. \ \sqrt{\frac{dr^{p2p6p}}{n}} :$$

Therefore,

$$E_{x;y} jf_{(T)}(x) \quad yj \quad E_{x;y} jf^?(x) \quad yj$$
$$. \ \sqrt{\frac{dr^{p2p6p}}{n}} + \sqrt{\frac{r^{p2p6p+1}}{m}} + \frac{}{n}^{1=4} \quad :$$

which completes the proof.

■

Appendix B. Transfer Learning

Proof [Proof of Theorem 3] The proof of Theorem 3 is virtually identical to that of Theorem 1. We can use Lemma 25 to construct $a^?$ such that if $^? = (a^?; W^{(1)}; b^{(1)})$ then with high probability,

$$L(^?) \quad \&2 \ . \ \sqrt{\frac{r^{p \ 2p \ 6p+1}}{m}} + \sqrt{\frac{}{n}} \quad \text{and} \quad ka^? k^2 \ . \ \sqrt{\frac{r^{p \ 2p \ 6p}}{m}}:$$

In addition, there exists such that if T $(\ ^{1} \ ^{1})$,

$$L(^{(T)}) \quad L(^?) \qquad \text{and} \qquad ka^{(T)}k \quad ka^?k:$$

Now let F = $f_{(a;W;b)}$: kak_2 $ka^?k$. Then by Lemma 48,

$$E_{x;y} jg_{a^{(T)}}(x) \quad yj \ \& \ O^{\sim}\left(\sqrt{\frac{d^2 r}{n}} + \sqrt{\frac{r^{p \ 2p}}{\min(n; m)}}\right) :$$

■

Appendix C. Concentration Lemmas

Lemma 27 (Corollary of Lemma 1 in (?)) Let X $^2(d)$. Then, for any t 0, P[X

$$d + 2\sqrt{dt} + 2t] \ \exp(\ t)$$
$$P[X \ d \quad 2\sqrt{dt}] \ \exp(\ t):$$

Corollary 28 Let w N(0; I$_d$). Then for some constant C,

$$P " \frac{k^? w k^2}{k w k^2} 2 \quad r d \frac{\dot{}}{C} \frac{C r}{d} \# \& 1:$$

Lemma 29 (Corollary 5.35 in Vershynin (2018)) Let X 2 Rnd with X$_{ij}$ N(0; 1). Then with probability at least 1 2e ,

$$k X k_2 \quad {}^p n + {}^p d + {}^p 2:$$

C.1. Polynomial Concentration

Lemma 30 Let g be a polynomial of degree p. Then there exists an absolute constant C$_p$ depending only on p such that for any ,

$$P[jg(x) \quad E[g(x)]j \quad {}^p E[g(x)^2]] \quad 2 \exp \quad C_p \min(^2; {}^{2=p}):$$

Proof Note that by Lemma 9,

$$E[r^k g(x)]_{HS} \qquad {}^p k!:$$

Therefore by Theorem 1.2 of (Gotze et al., 2019), there exists an absolute constant C$_p$ such that

$$P[jg(x) \quad E[g(x)]j \quad {}^p E[g(x)^2]] \quad 2 \exp \quad C_p \min_{1 \, p} {}^{2=s}_{s} = 2 \exp \quad C_p \min(^2; {}^{2=p}):$$

■

Lemma 31 Let (x) 2 fx; ReLU(x)g. There exists an absolute constant C such that for any x$_1$; : : : ; x$_n$ 2 R^d, there exists N^x; with jN^{x}j e$^{Cd \log(n=)}$ such that for every w 2 S$^{d\ 1}$, (w) 2 N^x ,^{0}(w x$_i$) = ^{0}((w) x$_i$) for i = 1; : : : ; n and kw (x)k .

Proof Note that the planes w x$_1$ = 0; : : : ; w x$_n$ = 0 divides the sphere S$^{d\ 1}$ into at most P $_{i=0}^{d}$ ^n_i . n^d convex regions. For each region there exists an net of size $\frac{3}{}$ ^{d}. Therefore we can take the union of these nets over each region which has size at most $\frac{3n}{}$ ^{d} = e$^{Cd \log(n=)}$. ■

Lemma 32 Let f(x) be a polynomial of degree p and let (x) 2 fx; ReLU(x)g. Then there exists an absolute constant C$_p$ depending only on p such that for any > 0, with probability at least 1 2ne , we have

$$\sup_{w 2 S^{d\ 1}} \frac{1}{n} \overset{n}{X}_{i=1} f(x_i) x_i \, (^0 w \; x_i) \qquad E \;\; f(x) x \, (^0 w \; x) \quad C_p \qquad {}^p E[g(x)^2] \; r \frac{d \; {}^{p+1}}{n}:$$

Proof Note that we may assume log(2n) otherwise there is nothing to prove. Let C be a sufficiently large absolute constant. We fix a truncation radius R := (C)$^{p=2}$ and WLOG assume that E[f(x)^2] = 1. Let

$$Y(w) := \frac{1}{n}\overset{n}{\underset{i=1}{X}} f(x_i)x_i{}^0(w\ \ x_i) \qquad \text{and} \qquad \tilde{Y}(w) := \frac{1}{n}\overset{n}{X} f(x_i)x_i{}^0(w\ \ x_i)1_{fjf(x_i)jRg}: {}_{i=1}$$

First, note that by Lemma 30, with probability at least 1 2e ^{2}, we have jf(x)j R. Therefore by a union bound we have with probability at last 1 2ne ^{2} we have jf(x$_i$)j R for i = 1;:::;n. Conditioned on this event, Y(w) = $\tilde{Y}$(w) uniformly over all w 2 S^d ^{1}. Next, we will bound sup$_w$ E$_x$[Y(w)] E$_x$[$\tilde{Y}$(w)]:

$$\begin{aligned}\sup_w E_x[Y(w)] \quad E_x[\tilde{Y}(w)] = \sup_w\ E_x\ \ g(x)x^0(w\ \ x)1_{fjg(x_i)j>Rg}\ \ E_x\\ jf(x)jkxk1_{fjg(x_i)j>Rg}\\ E\,g(x)^2{}^{1=2}\ E\ \ kxk^4\ {}^{1=4}\ P[jg(x_i)j > R]^{1=4}\\ 2\ \overset{p}{}\ 2d\ \exp(=2)\\ .\ \ \sqrt{\frac{d}{n}}:\end{aligned}$$

Finally, we concentrate sup$_w$ $\tilde{Y}$(w) E$_x$[$\tilde{Y}$(w)]. Let = $\sqrt{\frac{d}{n}}$, let N$_{1=4}$ be a minimal 1=4-net of S^d ^{1} with jN$_{1=4}$j e^{Cd} and let N ^{x} be the net defined in Lemma 31 with jN ^{x}j e$^{Cd\log(n=)}$ and let (w) be the projection function defined in Lemma 31. Then because Y($\tilde{w}$) = Y((w)),

$$\begin{aligned}&\sup_w \tilde{Y}(w) \quad E_x[\tilde{Y}(w)]\\ &\sup_{w2N^x} Y(\tilde{w}) \quad E_x[\tilde{Y}(w)] + \sup_w E_x[Y(\tilde{w})] \quad E_x[\tilde{Y}((w))]\\ &\sup_{w2N^x} Y(\tilde{w}) \quad E_x[\tilde{Y}(w)] + \sup_w kE_x[Y(w)] \quad E_x[Y((w))]k + O\left(\sqrt{\frac{d}{n}}\right):\end{aligned}$$

Next, because w ! E $_x$[Y(w)] is O(1) Lipschitz (see Section A.1.4), we can bound this by

$$\sup_w \tilde{Y}(w) \quad E_x[\tilde{Y}(w)] \quad \sup_{w2N^x} Y(\tilde{w}) \quad E_x[\tilde{Y}(w)] + O\left(\ +\sqrt{\frac{d}{n}}\right):$$

Therefore it remains to bound sup$_{w2N^x}$Y($\tilde{w}$) E$_x$[$\tilde{Y}$(w)]. First, for fixed w we have

$$\tilde{Y}(w) \quad E_x[\tilde{Y}(w)] = \sup_{u2S^{d\ \ 1}} u\ \ \left[Y(\tilde{w}) \quad E_x[\tilde{Y}(w)]\right]\ \ 2\ \sup_{u2N_{1=4}} u\ \ \left[Y(\tilde{w}) \quad E_x[\tilde{Y}(w)]\right]:$$

Let Z$_i$(w) := g(x$_i$)(u x$_i$)^0(w x$_i$)1$_{fjg(x)j<Rg}$ so that

$$u\ \ \left[\tilde{Y}(w) \quad E_x[\tilde{Y}(w)]\right] = \frac{1}{n}\overset{n}{\underset{i=1}{X}} Z_i(w) \quad E_x[Z_i(w)]:$$

Then note that for fixed w, $Z_i(w)$ is R-sub Gaussian so for each u 2 $N_{1=4}$, with probability 1 2e ^{z} we have

$$\overset{h}{u}\quad \tilde{Y}(w) \quad E_x[\tilde{Y}(w)] \overset{i}{\ } \quad R \quad \overset{r}{\ }\frac{2z}{n}:$$

so by a union bound we have with probability 1 $2e^{Cd\log(n=)}e^{\ z}$,

$$2 \sup_{u2N_{1=4};w2N^{x}} \overset{h}{u}\quad \tilde{Y}(w) \quad E_x[\tilde{Y}(w)] \overset{i}{\ } \quad 2R \quad \overset{r}{\ }\frac{2z}{n}:$$

so setting z = Cd log(n=) + we have with probability 1 2e,

$$2 \sup_{u2N_{1=4};w2N^{x}} \overset{h}{u}\quad \tilde{Y}(w) \quad E_x[\tilde{Y}(w)] \overset{i}{\ } \ . \ R \quad \overset{r}{\ }\frac{d\log(n=)\ -}{n\ +}:$$

Using = $\overset{q}{\ }\frac{d}{n}$ and putting everything together gives with probability 1 2ne ,

$$\sup kY(w) \quad E[Y(w)]k \ . \ \overset{r}{\ }\frac{(d\log n +)^p}{n} \quad \overset{r}{\ }\frac{d^{\,p+1}}{n}:w$$

■

Lemma 33 Let $_i$ f &;&g. Then with high probability,

$$\sup_w \frac{1}{n}\overset{n}{\underset{i=1}{X}} {}_i x_i{}^0(w\ x_i)\ .\ \& \quad \overset{r}{\ }\frac{d}{n}:$$

Proof Note that

$$\sup_w \frac{1}{n}\overset{n}{X} {}_i x_i{}^0(w\ x_i) = \sup_{u;w} \frac{1}{n}\ "\ \overset{n}{\underset{i=1}{X}} {}_i(u\ x_i)^0(w\ x_i)\ :{}_{i=1}\ \#$$

Next, note that for fixed u; w, $_i(ux_i)^0(wx_i)$ is &^{2}sub-Gaussian so for any > 0, with probability 1 2e ,

$$\frac{1}{n}\overset{n}{\underset{i=1}{X}} {}_i(u\ x_i)^0(w\ x_i)\ \& \quad \overset{r}{\ }\frac{\ }{n}:$$

Therefore,

$$\sup_{u;w} \ "\ \frac{1}{n}\overset{n}{\underset{i=1}{X}} {}_i(u\ x_i)^0(w\ x_i)\ \#\ . \quad \sup_{u2N_{1=4};w2N^{x}_{1=4}} \overset{n}{\underset{i=1}{X}} {}_i(u\ x_i)^0(w\ x_i)]:$$

By a union bound, with probability at least 1 2e,

$$\sup_{u2N_{1=4};w2N^{x}_{1=4}} \overset{n}{\underset{i=1}{X}} {}_i(u\ x_i)^0(w\ x_i)]\ .\ \&\ \overset{r}{\ }\frac{d\log n + }{n}\ .\ \&\ \overset{r}{\ }\frac{d}{n}$$

which completes the proof. ■

Corollary 34 With high probability,

$$\sup_{w} kg(w) \quad g_n(w)k \;. \quad r \frac{d^{p+1}}{n}:$$

Appendix D. CSQ Lower Bound

Proof [Proof of Lemma 5] The proof is a modified version of the proof in Szörényi (2009). Let $h; i_D$ denote the L^2 inner product with respect to D. We will show that there are at least two functions $f; g$ 2 F such that for each query h_k, $jhf; h_k i_D j$ and $jhg; h_k i_D j$. Therefore, we can simply respond to each query adversarially with 0 and it is impossible for the learner to distinguish between $f; g$. Note that failing to do so will result in a loss of $kf \quad gk_D^2$ 2 2. Let the kth query be h_k and let

$$A_k^+ = ff \; 2 \; F \;:\; hf; h_k i_D \;\; g \qquad \text{and} \qquad A_k = ff \; 2 \; F \;:\; hf; h_k i_D \;\; g$$

Then by Cauchy-Schwarz we have

$$A_k^{+\,2\,2} \quad \Big* h_k; \sum_{f\,2\,A_k^+} f \Big+_D^2 \quad \Big\| \sum_{f\,2\,A_k^+} f \Big\|_D^2 = \sum_{f;g\,2\,A_k^+} hf; gi_D \;\; A_k^+ + \;\; A_k^{+\,2} \qquad A_k^+$$

which implies

$$A_k^+ \;2\; \frac{1}{\quad}\;2\;\frac{1}{\quad}:$$

Similarly, we have that A_k $-\frac{1}{2}$ so the number of functions that are eliminated from the kth query is at most $-\frac{2}{2}$. We can continue this process for at most $\frac{jFj(\quad)}{2}$ iterations. ■

Proof [Proof of Lemma 6] Let $v_1; : : : ; v_k \quad S^{d \;\; 1}$. Then for every pair $i = j$, $v_i \;\; v_j$ is $O(d^{\;\;1})$ subgaussian so for an absolute constant c, with probability 1 2e $^{2c^2 d}$, $jv_i \;\; v_j j$. Therefore with probability 1 $k^2 e^{\;\;2c^2 d} >$ 0 this holds for all $i = j$ so there must exist at least one collection of such points. ■

Proof [Proof of Theorem 2] Let S be the set constructed in Lemma 6. Let

$$F = x \;! \;\frac{He_p(v \;\; x)}{p\overline{k!}} : v \; 2 \; S$$

and note that for all f 2 F, $kfk_D = 1$. Then for $v; w$ 2 S and $v = w$,

$$\frac{He_k(v \;\; x)}{p\overline{k!}}; \frac{He_k(w \;\; x)}{p\overline{k!}}_D = (v \;\; w)^k \;\; ^k:$$

Therefore, by Lemma 5 we have for any ,

$$4q \quad e^{c^2 d}(^2 \qquad ^k)$$

In particular if we take $= \frac{q \overline{\log(4q(cd)^{k=2})}}{cd}$ we get

$$2 \quad \frac{1 + \log^{k=2} \ 4q(cd)^{k=2}}{(cd)_k^{=2}} \ . \ \frac{\log^{k=2}(qd)}{d^{k=2}} :$$

■

Appendix E. Additional Technical Lemmas

For a k tensor T, let Sym(T) denote the symmetrization of T along all k! permutations of indices.

Lemma 35 There exist $T_0; : : : ; T_p$ such that

$$f^?(x) = \sum_{k} \ h T_k ; x^{k}_{i \ kp}$$

and $kT_k k_F \ . \ r^{\frac{p \ k}{4}}$ for k p.

Proof Note that from the Taylor series of $f^?(x)$ we have

$$T_k = \frac{r^k f^?(0)}{{}^j) \ k!} = \sum_{j p \ k} \frac{C_{j+k}(He_j(0))}{k!j!} = \sum_{2jp \ k} \frac{(\ 1)^j (2j \ \ 1)!!C_{2j+k}(I}{k!(2j)!} :$$

Therefore,

$$kT_k k_F \ . \ \sum \ C_{2j+k}(I \ {}^j) \ . \ r^{p \ k} \ \overline{\ {}_4 \ } : 2jp \ k$$

■

E.1. Gaussian Lemmas

Lemma 36

$$E_{wN(0;I_d)}[w^{2k}] = (2k \qquad 1)!! \, Sym(I_d^{k})$$

Proof We will show equality for each coordinate. Let $i_1; : : : ; i_{2k}$ be an index set and let $c_1; : : : ; c_d$ be defined by $c_j = jfk : i_k = jgj$. First we will consider the case there is an odd c_j. Then, $E_{wN(0;I_d)}[w^{2k}]_{i_1;:::;i_{2k}} = 0$ and $h(2k \qquad 1)!! \, Sym(I_d^{k}) i_{i_1;:::;i_{2k}} = 0$ because in order for this to be nonzero there must exist a pairing of $i_1; : : : ; i_{2k}$ such the numbers in each pair are identical.

Next, assume that each c_j is even. Then, $E_{wN(0;I_d)}[w^{2k}]_{i_1;:::;i_{2k}} = \prod_{j=1}^{d}(c_j \quad 1)!!$ by the standard formula for Gaussian moments. Finally, consider

$$h \ (2k \quad 1)!! \, Sym(I_d \qquad \overline{\quad 2k! \quad}$$

$$ {}^{k})^{i}_{i_1;:::;i_{2k}} = \;{}^{(2}_{\;k} \qquad 1)!!\; X \;\; 1_{i_1 = i_2} \;\; 1_{i_2{}^{k}\;{}^{1}= i_{2k}} : $$

Note that by a simple counting argument, the number of permutations such that this product of indicators is nonzero is exactly $k! \prod_{i=1}^{d} \frac{c_j!)}{(c_j=2!}$ as you can first order the indices corresponding to each c_j, then split them into groups of two, then shuffle these groups of two. Therefore,

$$h(2k \quad 1)!!I_d^{k}i_{i_1;:::;i_{2k}} = \frac{k!(2k \quad 1)!!}{2k!}\prod_{i=1}^{d}\frac{c_j!}{(c_j=2)!} = \frac{1}{2^k}\prod_{i=1}^{d}(c_j \quad 1)!!2_j^{c=2} = \prod_{i=1}^{d}(c_j \quad 1)!!$$

because $\sum_j c_j = 2k$, which completes the proof. ■

Definition 37 Let $fh_{kl}g$ and $fh_{kl}^{1}g$ denote the change of basis matrices between Hermite polynomials and monomials, i.e.

$$He_k(x) = \sum_{lk} h_{kl}x^l \quad \text{and} \quad x^k = \sum_{lk} h_{kl}^{1}He_l(x):$$

Note that

$$h_{kl} = \begin{cases}(1)^{\frac{k \; l}{2}}(k \quad l \quad 1)!!\binom{k}{l} & 2 j k \quad l \\ 0 & 2 - k \quad l\end{cases} \quad \text{and} \quad h_{kl}^{1} = \begin{cases}(k \quad l \quad 1)!!\binom{k}{l} & 2 j k \quad l \\ 0 & 2 - k \quad l\end{cases}:$$

Lemma 38 Let T be a symmetric p-tensor and let w $N(0; I_d)$. Then for k p,

$$E\|T(w^{k})\|_F^2 = \sum_{2lk}(k \quad 2l)!((2l \quad 1)!!)^2\binom{k}{2l}^2\|T(I^{l})\|_F^2:$$

Proof Let $T = \sum_i c_i v_i^p$ with $\|v_i\| = 1$. Using the change of basis $x^k \; ! \; \sum_{lk} h_{kl}^{1}He_l(x)$,

$$\begin{aligned}E\|T(w^{k})\|_F^2 &= \sum_{ij} c_i c_j \, E[(w \; v_i)^k(w \; v_j)^k](v_i \; v_j)^{p \quad k} \\ &= \sum_{lk} l!(h_{kl}^{1})^2 \sum_{ij} c_i c_i (v_i \; v_j)^{p \quad k+l} \\ &= \sum_{2lk}(k \quad 2l)!((2l \quad 1)!!)^2\binom{k}{2l}^2\|T(I^{l})\|_F^2:\end{aligned}$$

■

Corollary 39 Let T be a symmetric p-tensor with dim(span(T)) = r. For k p,

$$E\|T(w^{k})\|_F^2 \lesssim r^{\lfloor\frac{k}{2}\rfloor}\|T\|_F^2:$$

Proof The proof follows directly from Lemma 38 and the inequality $\|T(I^{l})\|_F^2 = \|T(_{span(T)}^{l})\|_F^2 \quad \|T\|_F^2 \|_{span(T)}\|_F^{2l} = r^l\|T\|_F^2$ for $2l$ k. ■

Corollary 40 Let T be a symmetric p-tensor with $\dim(\mathrm{span}(T)) = r$. With probability at least $1 \quad 2e \quad ,$

$$q \overline{} \quad \underline{k}$$

$$kT(w$$

$$^{k})k_F \ . \ kTk_F \qquad r^{b}{}_{2}{}^{ck}:$$

Proof Note that $F(w) = T(w$ $\sim$

$^{k})^2$ is a polynomial of degree $2k$. For $^{F}k \quad p$, let T_k be the $(k;k)$ tensor which comes from contracting the last d k indices of T

T, i.e.

$$(\tilde{T}_k)^{j_1;:::;j_k}_{i_1;:::;i_k} = T_{i_1;:::;i_k;i_{k+1};:::;i_p} T^{j_1;:::;j_k;i_{k+1};:::;i_p}:$$

Note that $F(w) = E_w[\tilde{T}_k(w$ $\sim$

$^{2k})]$ and $T_{k_F} \quad kTk^2$. Then by Theorem 38, ^{F}

$$E_w[F(w)^2] \ . \ \sum Sym(T_k)\tilde{}(I^l)^2_F \ . \ \sum T(I^l)^4 \quad kTk^4_F r^{2b\frac{k}{}c}:_F lk \quad \underline{pk}$$

Therefore by Theorem 30, with probability at least $1 \quad ^{2}2e \quad ,$ $F(w) \ . \ kTk^2_F r^{b}{}_{2}{}^{\frac{k}{}ck}$ and taking square roots completes the proof. ■

Corollary 41 For $k \quad p$,

$$E\,kT(w \quad ^{F}$$

$$^{k})k^2 \quad EhT;w$$

$$^{p}i^2:$$

Proof This follows immediately from Lemma 38 and $(k \quad 2l)! \ \frac{l}{2}^{2} \quad (p \quad 2l)! \ \frac{p}{2}^{l^2}$. ■

Corollary 42 Let $w \quad N(0;I_d)$. Then,

$$E[w$$

$$^{2k}] \quad k!_{Sym^k(R^d)}$$

where $_{Sym^k(R^d)}$ denotes the projection onto symmetric k-tensors. ■

Proof Considering only the $l = 0$ term in the above expansion of $E[w$

$^{2k}](T;T)$ gives

$$E[w \qquad ^{2}$$

$$^{2k}](T;T) \quad k!kTk_F:$$

Lemma 43 (Theorem 4.3 in (Prato and Tubaro, 2007)) Let f be a polynomial of degree p. Then

$$E_{w N(0;I_d)}[f(w)^k] \quad O_{k;p}(1) \quad E_{w N(0;I_d)}[f(w)^2]^{k=2}:$$

E.2. Sphere Lemmas

Lemma 44 Let (d). Then,

$$E[^{2k}] = \prod_{j=0}^{k\ 1}(d+2j) = d(d+2)\,(d+2k \quad 2) = (d^k):$$

Lemma 45 Let $\overline{w}$ $S^{d\ \ 1}$. Then,

$$E\left[\overline{w}^{2k}\right] = \frac{E_{wN(0;I_d)}[w^{2k}]}{E_{(d)}[^{2k}]}:$$

Proof This follows from the decomposition $w = \overline{w}$ with (d); $\overline{w}$ S^{d-1} independent. ■

Corollary 46 Let T be a symmetric p-tensor with dim(span(T)) = r. With probability at least $1 \quad 2e$,

$$\|T(\overline{w}^{k})\|_F \lesssim \|T\|_F \sqrt{\frac{r^{\lfloor\frac{k}{2}\rfloor}\ ^k}{d^k}}:$$

Corollary 47 Let $\overline{w}$ $S^{d\ \ 1}$. For k p,

$$E\|T(\overline{w}^{k})\|_F^2 \lesssim d^{p\ \ k}E\langle T;\overline{w}^{p}\rangle^2:$$

E.3. Rademacher Complexity Bounds

Lemma 48 Let $f = a^T(W^?x + b)$ be a two layer neural network. For fixed $W; b$, Let F

$$= \left\{f_{(a;W;b)} : \|a\|_2 B_a\right\}:$$

Then,

$$R_n(F) \sqrt{\frac{B_a^2(\|W\|_F^2 + \|b\|^2)}{n}}:$$

Proof

$$\begin{aligned}
R_n(F) &= E_{x;}\left[\sup_{f\in F}\left[\frac{1}{n}\sum_i {}_i\,a^T(Wx_i+b)\right]\right]\\
&= \frac{B_a}{n}E_{x;}\left[\left\|\sum_i {}_i(Wx_i+b)\right\|\right]\\
&\ \frac{B_a}{n}\sqrt{E_{x;}\left[\left\|\sum_i {}_i(Wx_i+b)\right\|_2^2\right]}\\
&= \frac{B_a}{\sqrt{n}}\sqrt{E_x\|(Wx_1+b)\|_2^2}\\
&\ \sqrt{\frac{B_a^2(\|W\|_F^2+\|b\|^2)}{n}}:
\end{aligned}$$

■

Lemma 49 Let $= (a; W; b)$ and let $f = a^T(Wx + b)$ be a two layer neural network. Let

$$F = ff \quad : kak_2 \quad B_a; kw_jk \quad B_wg:$$

Then,

$$R_n(F) \quad 2B_aB_w \; r\,\frac{md}{n}$$

Proof

$$
\begin{aligned}
R_n(F) &= E_{x;} \;"\sup_{f2F} \;"\frac{1}{n}\sum_i \;{}_i\, a^T(Wx_i + b)\;\#\# \\
&= \frac{B_a}{n} E_{x;} \;"\sup_{f2F} \sum_i {}_i(Wx_i + b)\;\# \\
& \frac{B_a \; p\,m}{n} E_{x;} \;"\sup_{f2F} \sum_i {}_i(Wx_i + b)\;{}^2\;\# \\
&= \frac{B_a \; p\,m}{n} E_{x;} \;"\sup_{f2F} \sum_i {}_i(w_j \; x_i + b_j)\;{}^1\# \\
& \frac{2B_a \; p\,m}{n} E_{x;} \;"\sup_{f2F} \sum_i {}_i(w_j \; x_i + b_j)\;\# \\
& \frac{2B_a \; p\,m}{n} E_{x;} \;"\sup \sum_i {}_i(w_j \; x_i)\;{}_{f2F}\;\# \\
& 2B_aB_w \; r\,\frac{md}{n}:
\end{aligned}
$$

■