Eigenvalue Decay Implies Polynomial-Time Learnability for Neural Networks

Surbhi Goel * Adam Klivans†

August 15, 2017

Abstract

We consider the problem of learning function classes computed by neural networks with various activations (e.g. ReLU or Sigmoid), a task believed to be computationally intractable in the worst-case. A major open problem is to understand the minimal assumptions under which these classes admit provably efficient algorithms. In this work we show that a natural distributional assumption corresponding to *eigenvalue decay* of the Gram matrix yields polynomial-time algorithms in the non-realizable setting for expressive classes of networks (e.g. feed-forward networks of ReLUs). We make no assumptions on the structure of the network or the labels. Given sufficiently-strong polynomial eigenvalue decay, we obtain *fully*-polynomial time algorithms in *all* the relevant parameters with respect to square-loss. Milder decay assumptions also lead to improved algorithms. This is the first purely distributional assumption that leads to polynomial-time algorithms for networks of ReLUs, even with one hidden layer. Further, unlike prior distributional assumptions (e.g., the marginal distribution is Gaussian), eigenvalue decay has been observed in practice on common data sets.

*Department of Computer Science, UT-Austin, surbhi@cs.utexas.edu. Work supported by a Microsoft Data Science Initiative Award.

†Department of Computer Science, UT-Austin, klivans@cs.utexas.edu. Part of this work was done while visiting the Simons Institute for Theoretical Computer Science.

1 Introduction

Understanding the computational complexity of learning neural networks from random examples is a fundamental problem in machine learning. Several researchers have proved results showing computational *hardness* for the worst-case complexity of learning various networks– that is, when no assumptions are made on the underlying distribution or the structure of the network [10, 16, 24, 29, 47]. As such, it seems necessary to take some assumptions in order to develop efficient algorithms for learning deep networks (the most expressive class of networks known to be learnable in polynomial-time without any assumptions is a sum of one hidden layer of sigmoids [16]). A major open question is to understand what are the "correct" or minimal assumptions to take in order to guarantee efficient learnability[1]. An oft-taken assumption is that the marginal distribution is equal to some smooth distribution such as a multivariate Gaussian. Even under such a distributional assumption, however, there is evidence that fully polynomial-time algorithms are still hard to obtain for simple classes of networks [22, 39]. As such, several authors have made further assumptions on the underlying structure of the model (and/or work in the noiseless or *realizable* setting).

In fact, in an interesting recent work, Shamir [37] has given evidence that both distributional assumptions and assumptions on the network structure are necessary for efficient learnability using gradient-based methods. Our main result is that under *only* an assumption on the marginal distribution, namely eigenvalue decay of the Gram matrix, there exist efficient algorithms for learning broad classes of neural networks even in the non-realizable (agnostic) setting with respect to square loss. Furthermore, eigenvalue decay has been observed often in real-world data sets, unlike distributional assumptions that take the marginal to be unimodal or Gaussian. As one would expect, stronger assumptions on the eigenvalue decay result in polynomial learnability for broader classes of networks, but even mild eigenvalue decay will result in savings in runtime and sample complexity.

The relationship between our assumption on eigenvalue decay and prior assumptions on the marginal distribution being Gaussian is similar in spirit to the dichotomy between the complexity of certain algorithmic problems on power-law graphs versus Erdős-Rényi graphs. Several important graph problems such as clique-finding become much easier when the underlying model is a random graph with appropriate power-law decay (as opposed to assuming the graph is generated from the classical $G(n,p)$ model) [6, 25]. In this work we prove that neural network learning problems become tractable when the underlying distribution induces an empirical gram matrix with sufficiently strong eigenvalue-decay.

1.1 Our Contributions

Our main result is quite general and holds for any function class that can be suitably embedded in an RKHS (Reproducing Kernel Hilbert Space) with corresponding kernel function k (we refer readers unfamiliar with kernel methods to [33]). Given m draws from a distribution $(\mathbf{x}_1, \ldots, \mathbf{x}_m)$ and kernel k, recall that the *Gram matrix* K is an $m \times m$ matrix where the i, j entry equals $k(\mathbf{x}_i, \mathbf{x}_j)$. For ease of presentation, we begin with an informal statement of our main result that highlights the relationship between the eigenvalue decay assumption and the run-time and sample complexity of our final algorithm.

[1]For example, a very recent paper of Song, Vempala, Xie, and Williams [39] asks "What form would such an explanation take, in the face of existing complexity-theoretic lower bounds?"

Theorem 1 (Informal)**.** *Fix function class $\mathcal{C}$ and kernel function k. Assume $\mathcal{C}$ is approximated in the corresponding RKHS with norm bound B. After drawing m samples, let K/m be the (normalized) $m \times m$ Gram matrix with eigenvalues $\{\lambda_1, \ldots, \lambda_m\}$. For error parameter $\epsilon > 0$,*

1. *If, for sufficiently large i, $\lambda_i \approx O(i^{-p})$, then $\mathcal{C}$ is efficiently learnable with $m = \tilde{O}(B^{1/p}/\epsilon^{2+3/p})$.*
2. *If, for sufficiently large i, $\lambda_i \approx O(e^{-i})$, then $\mathcal{C}$ is efficiently learnable with $m = \tilde{O}(\log B/\epsilon^2)$.*

We allow a failure probability for the event that the eigenvalues do not decay. In all prior work, the sample complexity m depends linearly on B, and for many interesting concept classes (such as ReLUs), B is exponential in one or more relevant parameters. Given Theorem 1, we can use known structural results for embedding neural networks into an RKHS to estimate B and take a corresponding eigenvalue decay assumption to obtain polynomial-time learnability. Applying bounds recently obtained by Goel et al. [16] we have

Corollary 2. *Let $\mathcal{C}$ be the class of all fully-connected networks of ReLUs with one-hidden layer of ℓ hidden ReLU activations feeding into a single ReLU output activation (i.e., two hidden layers or depth-3). Then, assuming eigenvalue decay of $O(i^{-\ell/\epsilon})$, $\mathcal{C}$ is learnable in polynomial time with respect to square loss on $\mathbb{S}^{n-1}$. If ReLU is replaced with sigmoid, then we require eigenvalue decay $O(i^{-\sqrt{\ell}\log(\sqrt{\ell}/\epsilon)})$.*

For higher depth networks, bounds on the required eigenvalue decay can be derived from structural results in [16]. Without taking an assumption, the fastest known algorithms for learning the above networks run in time exponential in the number of hidden units and accuracy parameter (but polynomial in the dimension) [16].

Our proof develops a novel approach for bounding the generalization error of kernel methods, namely we develop *compression schemes* tailor-made for classifiers induced by kernel-based regression, as opposed to current Rademacher-complexity based approaches. Roughly, a compression scheme is a mapping from a training set S to a small subsample S' and *side-information* $\mathcal{I}$. Given this compressed version of S, the decompression algorithm should be able to generate a classifier h. In recent work, David, Moran and Yehudayoff [13] have observed that if the size of the compression is much less than m (the number of samples), then the empirical error of h on S is close to its true error with high probability.

At the core of our compression scheme is a method for giving small description length (i.e., $o(m)$ bit complexity), approximate solutions to instances of kernel ridge regression. Even though we assume K has decaying eigenvalues, K is neither sparse nor low-rank, and even a single column or row of K has bit complexity at least m, since K is an $m \times m$ matrix! Nevertheless, we can prove that recent tools from Nyström sampling [31] imply a type of sparsification for solutions of certain regression problems involving K. Additionally, using preconditioning, we can bound the bit complexity of these solutions and obtain the desired compression scheme. At each stage we must ensure that our compressed solutions do not lose too much accuracy, and this involves carefully analyzing various matrix approximations. Our methods are the first compression-based generalization bounds for kernelized regression.

1.2 Related Work

Kernel methods [33] such as SVM, kernel ridge regression and kernel PCA have been extensively studied due to their excellent performance and strong theoretical properties. For large

data sets, however, many kernel methods become computationally expensive. The literature on approximating the Gram matrix with the overarching goal of reducing the time and space complexity of kernel methods is now vast. Various techniques such as random sampling [43], subspace embedding [2], and matrix factorization [15] have been used to find a low-rank approximation that is efficient to compute and gives small approximation error. The most relevant set of tools for our paper is Nyström sampling [43, 14], which constructs an approximation of K using a subset of the columns indicated by a selection matrix S to generate a positive semi-definite approximation. Recent work on leverage scores have been used to improve the guarantees of Nyström sampling in order to obtain linear time algorithms for generating these approximations [31].

The novelty of our approach is to use Nyström sampling in conjunction with compression schemes to give a new method for giving provable *generalization error* bounds for kernel methods. Compression schemes have typically been studied in the context of classification problems in PAC learning and for combinatorial problems related to VC dimension [26, 27]. Only recently have some authors considered compression schemes in a general, real-valued learning scenario [13]. Cotter, Shalev-Shwartz, and Srebro have studied compression in the context of classification using SVMs and prove that for general distributions, compressing classifiers with low generalization error is not possible [9].

The general phenomenon of eigenvalue decay of the Gram matrix has been studied from both a theoretical and applied perspective. Some empirical studies of eigenvalue decay and related discussion can be found in [30, 38, 41]. There has also been prior work relating eigenvalue decay to generalization error in the context of SVMs or Kernel PCA (e.g., [32, 38]). Closely related notions to eigenvalue decay are that of *local Rademacher complexity* due to Bartlett, Bousquet, and Mendelson [4] (see also [5]) and that of *effective dimensionality* due to Zhang [46].

The above works of Bartlett et al. and Zhang give improved generalization bounds via data-dependent estimates of eigenvalue decay of the kernel. At a high level, the goal of these works is to work under an assumption on the effective dimension and improve Rademacher-based generalization error bounds from $1/\sqrt{m}$ to $1/m$ (m is the number of samples) for functions embedded in an RKHS of unit norm. These works do not address the main obstacle of this paper, however, namely overcoming the complexity of the norm of the approximating RKHS. Their techniques are mostly incomparable even though the intent of using effective dimension as a measure of complexity is the same.

Shamir has shown that for general linear prediction problems with respect to square-loss and norm bound B, a sample complexity of $\Omega(B)$ is required for gradient-based methods [36]. Our work shows that eigenvalue decay can dramatically reduce this dependence, even in the context of kernel regression where we want to run in time polynomial in n, the dimension, rather than the (much larger) dimension of the RKHS.

1.3 Recent work on Learning Neural Networks

Due in part to the recent exciting developments in deep learning, there have been several works giving provable results for learning neural networks with various activations (threshold, sigmoid, or ReLU). For the most part, these results take various assumptions on either 1) the distribution (e.g., Gaussian or Log-Concave) or 2) the structure of the network architecture (e.g. sparse, random, or non-overlapping weight vectors) or both and often have a bad dependence on one or more of the relevant parameters (dimension, number of hidden units, depth, or accuracy).

Another way to restrict the problem is to work only in the noiseless/realizable setting. Works that fall into one or more of these categories include [23, 48, 44, 18, 34, 45, 11]. Kernel methods have been applied previously to learning neural networks [47, 29, 16, 12]. The current broadest class of networks known to be learnable in fully polynomial-time in all parameters with no assumptions is due to Goel et al. [16], who showed how to learn a sum of one hidden layer of sigmoids over the domain of $\mathbb{S}^{n-1}$, the unit sphere in n dimensions. We are not aware of other prior work that takes only a distributional assumption on the marginal and achieves fully polynomial-time algorithms for even simple networks (for example, one hidden layer of ReLUs).

Much work has also focused on the ability of gradient descent to succeed in parameter estimation for learning neural networks under various assumptions with an intense focus on the structure of local versus global minima [8, 20, 7, 40]. Here we are interested in the traditional task of learning in the non-realizable or agnostic setting and allow ourselves to output a hypothesis outside the function class (i.e., we allow improper learning). It is well known that for even simple neural networks, for example for learning a sigmoid with respect to square-loss, there may be many bad local minima [1]. Improper learning allows us to avoid these pitfalls.

2 Preliminaries

Notation. The input space is denoted by $\mathcal{X}$ and the output space is denoted by $\mathcal{Y}$. Vectors are represented with boldface letters such as $\mathbf{x}$. We denote a kernel function by $k_\psi(x, x') = \langle \psi(x), \psi(x') \rangle$ where ψ is the associated feature map and for the kernel and $\mathcal{K}_\psi$ is the corresponding reproducing kernel Hilbert space (RKHS). For necessary background material on kernel methods we refer the reader to [33].

2.1 Model and Generalization Bounds

We will work in the general non-realizable model of statistical learning theory also known as the *agnostic model of learning*. In this model, the labels presented to the learner are arbitrary, and the goal is to output a hypothesis that is competitive with the best fitting function from some fixed class:

Definition 3 (Agnostic Learning [21, 17])**.** *A concept class $\mathcal{C} \subseteq \mathcal{Y}^{\mathcal{X}}$ is agnostically learnable with respect to loss function $l : \mathcal{Y}' \times \mathcal{Y} \to \mathbb{R}^+$ (where $\mathcal{Y} \subseteq \mathcal{Y}'$) and distribution D over $\mathcal{X} \times \mathcal{Y}$, if for every $\delta, \epsilon > 0$ there exists a learning algorithm $\mathcal{A}$ given access to examples drawn from D, $\mathcal{A}$ outputs a hypothesis $h : \mathcal{X} \to \mathcal{Y}'$, such that with probability at least $1 - \delta$,*

$$E_{(\boldsymbol{x},y)\sim D}[l(h(\boldsymbol{x}), y)] \leq \min_{c \in C} E_{(\boldsymbol{x},y)\sim D}[l(c(\boldsymbol{x}), y)] + \epsilon. \tag{1}$$

Furthermore, we say that $\mathcal{C}$ is efficiently agnostically learnable to error ϵ *if $\mathcal{A}$ can output an h satisfying Equation* (1) *with running time polynomial in n, $1/\epsilon$ and $1/\delta$.*

The agnostic model generalizes Valiant's PAC model of learning [42], and so all of our results will hold for PAC learning as well. The following is a well known theorem for proving generalization based on Rademacher complexity.

Theorem 4 ([5])**.** *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$ and let $l : \mathcal{Y}' \times \mathcal{Y}$ be a b-bounded loss function that is L-Lispschitz in its first argument. Let $\mathcal{F}$ be a class of functions from $\mathcal{X}$ to $\mathcal{Y}'$*

and for any $f \in \mathcal{F}$, and $S = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)) \sim \mathcal{D}^m$ and $\delta > 0$, with probability at least $1 - \delta$ we have,

$$\left| E_{(x,y)\sim\mathcal{D}}[l(f(\boldsymbol{x}), y)] - \frac{1}{m}\sum_{i=1}^{m} l(f(\boldsymbol{x}_i), y_i) \right| \leq 4 \cdot L \cdot \mathcal{R}_m(\mathcal{F}) + 2 \cdot b \cdot \sqrt{\frac{\log(1/\delta)}{2m}}$$

where $\mathcal{R}_m(\mathcal{F})$ is the Rademacher complexity of the function class $\mathcal{F}$.

The Rademacher complexity of this linear class can be bounded by using the following theorem.

Theorem 5 ([19])**.** *Let $\mathcal{K}$ be a subset of a Hilbert space equipped with inner product $\langle \cdot, \cdot \rangle$ such that for each $x \in \mathcal{K}$, $\langle \boldsymbol{x}, \boldsymbol{x} \rangle \leq X^2$, and let $\mathcal{W} = \{\boldsymbol{x} \to \langle \boldsymbol{x}, \boldsymbol{w} \rangle \mid \langle \boldsymbol{w}, \boldsymbol{w} \rangle \leq W^2\}$ be a class of linear functions. Then it holds that*

$$\mathcal{R}_m(\mathcal{W}) \leq X \cdot W \cdot \sqrt{\frac{1}{m}}.$$

2.2 Selection and Compression Schemes

It is well known that in the context of PAC learning Boolean function classes, a suitable type of compression of the training data implies learnability [28]. Perhaps surprisingly, the details regarding the relationship between compression and ceratin other real-valued learning tasks have not been worked out until very recently. A convenient framework for us will be the notion of compression and selection schemes due to David et al. [13].

A selection scheme is a pair of maps (κ, ρ) where κ is the selection map and ρ is the reconstruction map. κ takes as input a sample $\mathcal{S} = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$ and outputs a sub-sample $\mathcal{S}'$ and a finite binary string b as side information. ρ takes this input and outputs a hypothesis h. The *size* of the selection scheme is defined to be $k(m) = |\mathcal{S}'| + |b|$. We present a slightly modified version of the definition of an approximate compression scheme due to [13]:

Definition 6 ((ϵ, δ)-approximate agnostic compression scheme)**.** *A selection scheme (κ, ρ) is an (ϵ, δ)-approximate agnostic compression scheme for hypothesis class $\mathcal{H}$ and sample satisfying property P if for all samples $\mathcal{S}$ that satisfy P with probability $1 - \delta$, $f = \rho(\kappa(S))$ satisfies $\sum_{i=1}^{m} l(f(\boldsymbol{x}_i), y_i) \leq \min_{h\in\mathcal{H}} \left(\sum_{i=1}^{m} l(h(\boldsymbol{x}_i), y_i)\right) + \epsilon$.*

Compression has connections to learning in the general loss setting through the following theorem which shows that as long as $k(m)$ is small, the selection scheme generalizes.

Theorem 7 (Theorem 30.2 [35], Theorem 3.2 [13])**.** *Let (κ, ρ) be a selection scheme of size $k = k(m)$, and let $A_{\mathcal{S}} = \rho(\kappa(\mathcal{S}))$. Given m i.i.d. samples drawn from any distribution $\mathcal{D}$ such that $k \leq m/2$, for constant bounded loss function $l : \mathcal{Y}' \times \mathcal{Y} \to \mathbb{R}^+$ with probability $1 - \delta$, we have*

$$\left| E_{(\boldsymbol{x},y)\sim\mathcal{D}}[l(A_{\mathcal{S}}(x), y)] - \sum_{i=1}^{m} l(A_{\mathcal{S}}(\boldsymbol{x}_i), y_i) \right| \leq \sqrt{\epsilon \cdot \left(\frac{1}{m}\sum_{i=1}^{m} l(A_{\mathcal{S}}(\boldsymbol{x}_i), y_i)\right)} + \epsilon$$

where $\epsilon = 50 \cdot \frac{k\log(m/k)+\log(1/\delta)}{m}$.

3 Problem Overview

In this section we give a general outline for our main result. Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ be a training set of samples drawn i.i.d. from some arbitrary distribution $\mathcal{D}$ on $\mathcal{X} \times [0,1]$ where $\mathcal{X} \subseteq \mathbb{R}^n$. Let us consider a concept class $\mathcal{C}$ such that for all $c \in \mathcal{C}$ and $\mathbf{x} \in \mathcal{X}$ we have $c(\mathbf{x}) \in [0,1]$. We wish to learn the concept class $\mathcal{C}$ with respect to the square loss, that is, we wish to find $c \in \mathcal{C}$ that approximately minimizes $E_{(\mathbf{x},y)\sim\mathcal{D}}[(c(\mathbf{x}) - y)^2]$. A common way of solving this is by solving the empirical minimization problem (ERM) given below and subsequently proving that it generalizes.

Optimization Problem 1

$$\underset{c\in\mathcal{C}}{\text{minimize}} \qquad \frac{1}{m}\sum_{i=1}^{m}(c(\mathbf{x}_i) - y_i)^2$$

Unfortunately, it may not be possible to efficiently solve the ERM in polynomial-time due to issues such as non-convexity. A way of tackling this is to show that the concept class can be approximately minimized by another hypothesis class of linear functions in a high dimensional feature space (this in turn presents new obstacles for proving generalization-error bounds, which is the focus of this paper).

Definition 8 (ϵ-approximation)**.** *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be function classes mapping domain $\mathcal{X}$ to $\mathbb{R}$. $\mathcal{C}_1$ is ϵ-approximated by $\mathcal{C}_2$ if for every $c \in \mathcal{C}_1$ there exists a $c' \in \mathcal{C}_2$ such that for all $x \in \mathcal{X}, |c(x) - c'(x)| \leq \epsilon$.*

Suppose $\mathcal{C}$ can be ϵ-approximated in the above sense by the hypothesis class $H_\psi = \{\mathbf{x} \to \langle \mathbf{v}, \psi(\mathbf{x})\rangle | \mathbf{v} \in \mathcal{K}_\psi, \langle \mathbf{v}, \mathbf{v}\rangle \leq B\}$ for some B and kernel function k_ψ. We further assume that the kernel is bounded, that is, $|k_\psi(\mathbf{x}, \mathbf{x'})| \leq M$ for some $M > 0$ for all $\mathbf{x}, \mathbf{x'} \in \mathcal{X}$. Thus, the problem relaxes to the following,

Optimization Problem 2

$$\underset{v\in\mathcal{K}_\psi}{\text{minimize}} \qquad \frac{1}{m}\sum_{i=1}^{m}(\langle \mathbf{v}, \psi(\mathbf{x}_i)\rangle - y_i)^2 \qquad \text{subject to} \qquad \langle \mathbf{v}, \mathbf{v}\rangle \leq B$$

Using the Representer theorem, we have that the optimum solution for the above is of the form $\mathbf{v}^* = \sum_{i=1}^{m} \alpha_i \psi(\mathbf{x}_i)$ for some $\alpha \in \mathbb{R}^n$. Denoting the sample kernel matrix be K such that $K_{i,j} = k_\psi(\mathbf{x}_i, \mathbf{x}_j)$, the above optimization problem is equivalent to the following optimization problem,

Optimization Problem 3

$$\underset{\alpha\in\mathbb{R}^m}{\text{minimize}} \qquad \frac{1}{m}||K\alpha - Y||_2^2 \qquad \text{subject to} \qquad \alpha^T K \alpha \leq B$$

where Y is the vector corresponding to all y_i and $||Y||_\infty \leq 1$ since $\forall i \in [m], y_i \in [0,1]$. Let α_B be the optimal solution of the above problem. This is known to be efficiently solvable in $\mathsf{poly}(m,n)$ time as long as the kernel function is efficiently computable.

Applying Rademacher complexity bounds to $\mathcal{H}_\psi$ yields generalization error bounds that decrease, roughly, on the order of $B/\sqrt{m}$ (Theorem 4 and 5). If B is exponential in $1/\epsilon$, the accuracy parameter, or in n, the dimension, as in the case of bounded depth networks of ReLUs, then this dependence leads to exponential sample complexity. As mentioned in Section 1.2, in the context of eigenvalue decay, various results [46, 4, 5] have been obtained to improve the dependence of $B/\sqrt{m}$ to B/m, but little is known about improving the dependence on B.

Our goal is to show that eigenvalue decay of the empirical Gram matrix does yield generalization bounds with better dependence on B. The key is to develop a novel compression scheme for kernelized ridge regression. We give a step-by-step analysis for how to generate an approximate, compressed version of the solution to Optimization Problem 3. Then, we will carefully analyze the bit complexity of our approximate solution and realize our compression scheme. Finally, we can put everything together and show how quantitative bounds on eigenvalue decay directly translate into compressions schemes with low generalization error.

4 Compressing the Kernel Solution

Through a sequence of steps, we will sparsify α to find a solution of much smaller bit complexity that is still an approximate solution (to within a small additive error). The quality and size of the approximation will depend on the eigenvalue decay.

4.1 Lagrangian Relaxation

We relax Optimization Problem 3 and consider the Lagrangian version of the problem to account for the norm bound constraint. This version is convenient for us, as it has a nice closed-form solution.

Optimization Problem 4

$$\underset{\alpha \in \mathbb{R}^m}{\text{minimize}} \qquad \frac{1}{m}||K\alpha - Y||_2^2 + \lambda \alpha^T K \alpha$$

We will later set λ such that the error of considering this relaxation is small. It is easy to see that the optimal solution for the above lagrangian version is $\alpha = (K + \lambda m I)^{-1} Y$.

4.2 Preconditioning

To avoid extremely small or non-zero eigenvalues, we consider a perturbed version of K, $K_\gamma = K + \gamma m I$. This gives us that the eigenvalues of K_γ are always greater than or equal to γm. This property is useful for us in our later analysis. Henceforth, we consider the following optimization problem on the perturbed version of K:

Optimization Problem 5

$$\underset{\alpha \in \mathbb{R}^m}{\text{minimize}} \qquad \frac{1}{m}||K_\gamma \alpha - Y||_2^2 + \lambda \alpha^T K_\gamma \alpha$$

The optimal solution for perturbed version is $\alpha_\gamma = (K_\gamma + \lambda m I)^{-1} Y = (K + (\lambda + \gamma) m I)^{-1} Y$.

4.3 Sparsifying the Solution via Nyström Sampling

We will now use tools from Nyström Sampling to sparsify the solution obtained from Optimzation Problem 5. To do so, we first recall the definition of effective dimension or degrees of freedom for the kernel [46]:

Definition 9 (η-effective dimension)**.** *For a positive semidefinite $m \times m$ matrix K and parameter η, the η-effective dimension of K is defined as $d_\eta(K) = tr(K(K + \eta m I)^{-1})$.*

Various kernel approximation results have relied on this quantity, and here we state a recent result due to [31] who gave the first application independent result that shows that there is an efficient way of computing a set of columns of K such that $\bar{K}$, a matrix constructed from the columns is close in terms of 2-norm to the matrix K. More formally,

Theorem 10 ([31])**.** *For kernel matrix K, there exists an algorithm that gives a set of $O\left(d_\eta(K) \log\left(d_\eta(K)/\delta\right)\right)$ columns, such that $\bar{K} = KS(S^T K S)^\dagger S^T K$ where S is the matrix that selects the specific columns, satisfies with probability $1 - \delta$, $\bar{K} \preceq K \preceq \bar{K} + \eta m I$.*

It can be shown that $\bar{K}$ is positive semi-definite. Also, the above implies $||K - \bar{K}||_2 \leq \eta m$. We use the decay to approximate the Kernel matrix with a low-rank matrix constructed using the columns of K. Let $\bar{K}_\gamma$ be the matrix obtained by applying Theorem 10 to K_γ for $\eta > 0$ and consider the following optimization problem,

Optimization Problem 6

$$\underset{\alpha \in \mathbb{R}^m}{\text{minimize}} \qquad \frac{1}{m}||\bar{K}_\gamma \alpha - Y||_2^2 + \lambda \alpha^T \bar{K}_\gamma \alpha$$

The optimal solution for the above is $\bar{\alpha}_\gamma = \left(\bar{K}_\gamma + \lambda m I\right)^{-1} Y$. Since $\bar{K}_\gamma = K_\gamma S(S^T K_\gamma S)^\dagger S^T K_\gamma$, solving for the above enables us to get a solution $\alpha^* = S(S^T K_\gamma S)^\dagger S^T K_\gamma \bar{\alpha}_\gamma$, which is a k-sparse vector for $k = O\left(d_\eta(K_\gamma) \log\left(d_\eta(K_\gamma)/\delta\right)\right)$.

4.4 Bounding the Error of the Sparse Solution

We bound the additional error incurred by our sparse hypothesis α^* compared to α_B. To do so, we bound the error for each of the approximations: sparsification, preconditioning and lagrangian relaxation in the following lemma.

Lemma 11. *The errors due to the following approximations can be bounded as follows.*

1. *Error due to sparsification:* $||\bar{K}_\gamma \bar{\alpha}_\gamma - Y||_2 \leq ||K_\gamma \alpha_\gamma - Y||_2 + \frac{\eta\sqrt{m}}{\lambda + \gamma}$

2. *Error due to preconditioning:* $||K_\gamma \alpha_\gamma - Y||_2 \leq ||K\alpha - Y||_2 + \frac{\gamma\sqrt{m}}{\lambda+\gamma}$

3. *Error due to lagrangian relaxation:* $||K\alpha - Y||_2 \leq ||K\alpha_B - Y||_2 + \sqrt{\lambda m B}$

Proof. The errors can be bounded as follows.

1. We have,

$$\begin{aligned}
&||\bar{K}_\gamma \bar{\alpha}_\gamma - Y||_2 - ||K_\gamma \alpha_\gamma - Y||_2 \\
&\quad \leq ||\bar{K}_\gamma \bar{\alpha}_\gamma - K_\gamma \alpha_\gamma||_2 && (2)\\
&\quad = ||\bar{K}_\gamma \left(\bar{K}_\gamma + \lambda m I\right)^{-1} Y - K_\gamma \left(K_\gamma + \lambda m I\right)^{-1} Y||_2 && (3)\\
&\quad = \lambda m || \left(- \left(\bar{K}_\gamma + \lambda m I\right)^{-1} + \left(K_\gamma + \lambda m I\right)^{-1} \right) Y||_2 && (4)\\
&\quad = \lambda m || \left(\bar{K}_\gamma + \lambda m I\right)^{-1} \left(\bar{K}_\gamma - K_\gamma\right) \left(K_\gamma + \lambda m I\right)^{-1} Y||_2 && (5)\\
&\quad \leq \lambda m || \left(\bar{K}_\gamma + \lambda m I\right)^{-1} ||_2 || \bar{K}_\gamma - K_\gamma||_2 || \left(K + (\lambda+\gamma) m I\right)^{-1} ||_2 ||Y||_2 && (6)\\
&\quad \leq \frac{||\bar{K}_\gamma - K_\gamma||_2}{(\lambda+\gamma)\sqrt{m}} \leq \frac{\eta\sqrt{m}}{\lambda+\gamma}. && (7)
\end{aligned}$$

   Here 2 follows from triangle inequality, 3 follows from substitution and 4 follows from using $A\left(A + cI\right)^{-1} = \left(A + cI - cI\right)\left(A + cI\right)^{-1} = I - c\left(A + cI\right)^{-1}$. 5 follows from $a^{-1} - b^{-1} = -a^{-1}\left(a - b\right)b^{-1}$ and 6 follows from $||AB||_2 \leq ||A||_2 ||B||_2$. Lastly 7 follows from $||A^{-1}||_2 = \lambda_{min}\left(A\right)^{-1}$, $\lambda_{min}\left(A + cI\right) \geq c$ for psd A. We also use $K_\gamma = K + \gamma m I$ and $||Y||_2 \leq \sqrt{m}$.

2. Similar to the above proof, we have,

$$\begin{aligned}
&||K_\gamma \alpha_\gamma - Y||_2 - ||K\alpha - Y||_2 \\
&\quad \leq ||K_\gamma \alpha_\gamma - K(K + \lambda m I)^{-1} Y||_2 && (8)\\
&\quad = ||K_\gamma \left(K_\gamma + \lambda m I\right)^{-1} Y - K \left(K + \lambda m I\right)^{-1} Y||_2 && (9)\\
&\quad = \lambda m || \left(K_\gamma + \lambda m I\right)^{-1} \left(K_\gamma - K\right) \left(K + \lambda m I\right)^{-1} Y||_2 && (10)\\
&\quad \leq \lambda m || \left(K + (\lambda+\gamma) m I\right)^{-1} ||_2 ||\gamma m I||_2 || \left(K + \lambda m I\right)^{-1} ||_2 ||Y||_2 && (11)\\
&\quad \leq \frac{\gamma\sqrt{m}}{\lambda+\gamma}. && (12)
\end{aligned}$$

3. Since α minimizes Optimization Problem 4, we have

$$\begin{aligned}
||K\alpha - Y||_2^2 \leq & ||K\alpha - Y||_2^2 + \lambda m \alpha^T K \alpha && (13)\\
& \leq ||K\alpha_B - Y||_2^2 + \lambda m \alpha_B^T K \alpha_B && (14)\\
& \leq ||K\alpha_B - Y||_2^2 + \lambda m B && (15)
\end{aligned}$$

   where the last inequality follows from $\alpha_B^T K \alpha_B \leq B$ by the constraint of the bounded optimization problem. Taking the square-root, we get,

$$||K\alpha - Y||_2 \leq \sqrt{||K\alpha_B - Y||_2^2 + \lambda m B} \leq ||K\alpha_B - Y||_2 + \sqrt{\lambda m B} \quad (16)$$

□

We now combine the above to give the following theorem.

Theorem 12 (Total Error). *For $\lambda = \frac{\epsilon^2}{81B}$, $\eta \leq \frac{\epsilon^3}{729B}$ and $\gamma \leq \frac{\epsilon^3}{729B}$, we have*

$$\frac{1}{m}||K_\gamma \alpha^* - Y||_2^2 \leq \frac{1}{m}||K\alpha_B - Y||_2^2 + \epsilon.$$

Proof. Note that $\bar{K}\bar{\alpha}_\gamma = K_\gamma \alpha^*$ by the definition of α^*, from the previous lemma, we have,

$$||\bar{K}\bar{\alpha}_\gamma - Y||_2 - ||K\alpha_B - Y||_2 \leq \frac{\eta\sqrt{m}}{\lambda+\gamma} + \frac{\gamma\sqrt{m}}{\lambda+\gamma} + \sqrt{\lambda m B} = \beta \tag{17}$$

where $\beta = \frac{(\eta+\gamma)\sqrt{m}}{\lambda+\gamma} + \sqrt{\lambda m B}$. Squaring and then dividing by m on both sides, we get

$$\frac{1}{m}||\bar{K}_\gamma\bar{\alpha}_\gamma - Y||_2^2 \leq \frac{1}{m}||K\alpha_B - Y||_2^2 + 2\frac{\beta}{m}||K\alpha_B - Y||_2 + \frac{\beta^2}{m} \tag{18}$$

$$\leq \frac{1}{m}||K\alpha_B - Y||_2^2 + 2\frac{\beta}{\sqrt{m}} + \frac{\beta^2}{m} \tag{19}$$

$$\leq \frac{1}{m}||K\alpha_B - Y||_2^2 + 3\frac{\beta}{\sqrt{m}} \tag{20}$$

The second inequality follows from $||K\alpha_B - Y||_2^2 \leq ||Y||_2^2 \leq m$ since 0 is a feasible solution for Optimization Problem 3. The last inequality follows from assuming $\frac{\beta}{\sqrt{m}} \leq 1$ which holds for our choice of β. Setting the values in the lemma satisfies the last inequality gives us $\beta \leq \frac{\epsilon\sqrt{m}}{3}$ giving us the desired bound. □

4.5 Computing the Sparsity of the Solution

To compute the sparsity of the solution, we need to bound $d_\eta(K_\beta)$. We consider the following different eigenvalue decays.

Definition 13 (Eigenvalue Decay). *Let the real eigenvalues of a symmetric $m \times m$ matrix A be denoted by $\lambda_1 \geq \cdots \geq \lambda_m$.*

1. *A is said to have (C,p)-**polynomial eigenvalue decay** if for all $i \in \{1,\ldots,m\}$, $\lambda_i \leq Ci^{-p}$.*
2. *A is said to have C-**exponential eigenvalue decay** if for all $i \in \{1,\ldots,m\}$, $\lambda_i \leq Ce^{-i}$.*

Note that in the above definitions C and p are not necessarily constants. We allow C and p to depend on other parameters (the choice of these parameters will be made explicit in subsequent theorem statements). We can now bound the effective dimension in terms of eigenvalue decay:

Theorem 14 (Bounding effective dimension). *For $\gamma m \leq \eta$, the η-effective dimension of K_γ can be bounded as follows,*

1. *If K/m has (C,p)-**polynomial eigenvalue decay** for $p > 1$ then $d_\eta(K_\gamma) \leq \left(\frac{C}{(p-1)\eta}\right)^{1/p} + 2$.*

*2. If K/m has C-**exponential eigenvalue decay** then $d_\eta(K_\gamma) \leq \log\left(\frac{C}{(e-1)\eta}\right) + 2$.*

Proof. Observe that,

$$\begin{aligned} d_\eta(K_\gamma) &= tr(K_\gamma(K_\gamma + \eta m I)^{-1}) \\ &= \sum_{i=1}^{m} \frac{\lambda_i(K_\gamma)}{\lambda_i(K_\gamma) + \eta m} \\ &\leq \sum_{i=1}^{j} \frac{\lambda_i(K_\gamma)}{\lambda_i(K_\gamma)} + \sum_{i=j+1}^{m} \frac{\lambda_i(K_\gamma)}{\eta m} \\ &\leq j + \sum_{i=j+1}^{m} \frac{\gamma m + \lambda_i(K)}{\eta m} \\ &\leq j + 1 + \sum_{i=j+1}^{m} \frac{\lambda_i(K)}{\eta m} \end{aligned}$$

Here the second equality follows from trace of matrix being equal to the sum of the eigenvalues and the last follows from $\gamma m \leq \eta$.

1. For (C, p)-polynomial eigenvalue decay with $p > 1$,

$$\sum_{i=k+1}^{m} \frac{\lambda_i(K)}{\eta m} = \sum_{i=k+1}^{m} \frac{Ci^{-p}}{\eta} \leq \frac{C}{\eta} \int_{k+1}^{\infty} i^{-p} di = \frac{C(k+1)^{-p+1}}{(p-1)\eta}$$

   Substituting $j = \left(\frac{C}{(p-1)\eta}\right)^{1/p}$ we get the required bound.

2. For C-exponential eigenvalue decay,

$$\sum_{i=k+1}^{m} \frac{\lambda_i(K)}{\eta m} = \sum_{i=k+1}^{m} \frac{Ce^{-i}}{\eta} \leq \sum_{i=k+1}^{\infty} \frac{Ce^{-i}}{\eta} = \frac{Ce^{-k}}{(e-1)\eta}$$

   Substituting $j = \log\left(\frac{C}{(e-1)\eta}\right)$ we get the required bound.

□

Remark: *Based on the above analysis, observe that we only need the eigenvalue decay to hold after the jth eigenvalue for j defined above. Thus the top $j-1$ eigenvalues need not be constrained.*

5 Bounding the Size of the Compression Scheme

The above analysis gives us a sparse solution for the problem and, in turn, an ϵ-approximation for the error on the overall sample $\mathcal{S}$ with probability $1-\delta$. We can now fully define our compression scheme for the hypothesis class H_ψ with respect to samples satisfying the eigenvalue decay property.

- **Selection Scheme** κ: Given input $\mathcal{S} = (\mathbf{x}_i, y_i)_{i=1}^m$,
  1. Use RLS-Nyström Sampling [31] to compute $\bar{K_\gamma} = K_\gamma S(S^T K_\gamma S)^\dagger S^T K_\gamma$ for $\eta = \frac{\epsilon^3}{5832B}$ and $\gamma = \frac{\epsilon^3}{5832Bm}$. Let $\mathcal{I}$ be the sub-sample corresponding to the columns selected using S.
  2. Solve Optimization Problem 6 for $\lambda = \frac{\epsilon^2}{324B}$ to get $\bar{\alpha}_\gamma$.
  3. Compute the $|\mathcal{I}|$-sparse vector $\alpha^* = S(S^T K_\gamma S)^\dagger S^T K_\gamma \bar{\alpha}_\gamma = K_\gamma^{-1} \bar{K}_\gamma \bar{\alpha}_\gamma$ (K_γ is invertible as all eigenvalues are non-zero).
  4. Output subsample $\mathcal{I}$ along with $\tilde{\alpha}^*$ which is α^* truncated to precision $\frac{\epsilon}{4M|\mathcal{I}|}$ per non-zero index.
- **Reconstruction Scheme** ρ: Given input subsample $\mathcal{I}$ and $\tilde{\alpha}^*$, output hypothesis,

$$h_{\mathcal{S}}(\mathbf{x}) = clip_{0,1}(\mathbf{w}^T \tilde{\alpha}^*)$$

  where $\mathbf{w}$ is a vector with entries $K(\mathbf{x}_i, \mathbf{x}) + \gamma m \mathbb{1}[\mathbf{x} = \mathbf{x}_i]$ for $i \in \mathcal{I}$ and 0 otherwise where $\gamma = \frac{\epsilon^3}{5832Bm}$. Note, $clip_{a,b}(x) = \max(a, \min(b, x))$ for some $a < b$.

The size of the above scheme can be bounded using the following lemma.

Lemma 15. *The bit complexity of the side information of the selection scheme κ given above is $O\left(d \log\left(\frac{d}{\delta}\right) \log\left(\frac{\sqrt{m}BMd\log(d/\delta)}{\epsilon^4}\right)\right)$ where d is the η-effective dimension of K_γ for $\eta = \frac{\epsilon^3}{5832B}$ and $\gamma = \frac{\epsilon^3}{5832Bm}$.*

Proof. From the selection scheme we can bound the norm of $\alpha^* = K_\gamma^{-1} \bar{K_\gamma} \bar{\alpha}_\gamma$ for $\gamma = \frac{\epsilon^3}{5832Bm}$, the side information, as follows,

$$\begin{aligned}
||\alpha^*||_2 &= ||K_\gamma^{-1} \bar{K_\gamma} \bar{\alpha}_\gamma||_2 && (21)\\
&= ||K_\gamma^{-1} \bar{K_\gamma} (\bar{K_\gamma} + \lambda m I)^{-1} Y||_2 && (22)\\
&\leq ||K_\gamma^{-1}||_2 ||\bar{K_\gamma}(\bar{K_\gamma} + \lambda m I)^{-1}||_2 ||Y||_2 && (23)\\
&\leq \frac{1}{\gamma m} \cdot 1 \cdot \sqrt{m} && (24)\\
&= \frac{1}{\gamma\sqrt{m}} = \frac{5832\sqrt{m}B}{\epsilon^3}. && (25)
\end{aligned}$$

Thus we can upper bound the bit complexity of the non-decimal part of α^* as,

$$\begin{aligned}
\sum_{i \in \mathcal{I}} \log\left(|\alpha_i^*|\right) &= \frac{1}{2} \sum_{i=1}^{|\mathcal{I}|} \log\left((\alpha_i^*)^2\right)\\
&\leq \frac{|\mathcal{I}|}{2} \log\left(\frac{\sum_{i=1}^{|\mathcal{I}|} (\alpha_i^*)^2}{|\mathcal{I}|}\right)\\
&\leq |\mathcal{I}| \log\left(\frac{||\alpha^*||_2}{\sqrt{|\mathcal{I}|}}\right) \leq |\mathcal{I}| \log\left(\frac{5832\sqrt{m}B}{\epsilon^3}\right)
\end{aligned}$$

where $|\mathcal{I}| = O\left(d \log\left(\frac{d}{\delta}\right)\right)$ according to Theorem 10. Since each non-zero index has $\frac{\epsilon}{4M|\mathcal{I}|}$ precision, we need $|\mathcal{I}| \log\left(\frac{4M|\mathcal{I}|}{\epsilon}\right)$ bits for the decimal part. Combining the two-parts we get the required bound. ∎

The following theorem shows that the above is a compression scheme for $\mathcal{H}_\psi$.

Theorem 16. *(κ, ρ) is an (ϵ, δ)-approximate agnostic compression scheme for the hypothesis class H_ψ for sample $\mathcal{S}$ of size $k(m, \epsilon, \delta, B, M) = O\left(d \log\left(\frac{d}{\delta}\right) \log\left(\frac{\sqrt{m}BMd\log(d/\delta)}{\epsilon^4}\right)\right)$ where d is the η-effective dimension of K_γ for $\eta = \frac{\epsilon^3}{5832B}$ and $\gamma = \frac{\epsilon^3}{5832Bm}$.*

Proof. For $\mathcal{S} = (\mathbf{x}_i, y_i)_{i=1}^m$ and $h_\mathcal{S}$ the output of the compression scheme, we have

$$\frac{1}{m}\sum_{i=1}^{m}(h_\mathcal{S}(\mathbf{x}_i) - y_i)^2 \leq \frac{1}{m}\sum_{i=1}^{m}\left(\sum_{j\in\mathcal{I}}(K(\mathbf{x}_j, \mathbf{x}_i) + \gamma m \mathbb{1}[\mathbf{x}_j = \mathbf{x}_i])\tilde{\alpha}_j^* - y_i\right)^2 \tag{26}$$

$$\leq \frac{1}{m}\sum_{i=1}^{m}\left(\sum_{j\in\mathcal{I}}(K(\mathbf{x}_j, \mathbf{x}_i) + \gamma m \mathbb{1}[\mathbf{x}_j = \mathbf{x}_i])\alpha_j^* - y_i\right)^2 + \frac{\epsilon}{2} \tag{27}$$

$$= \frac{1}{m}||K_\gamma \alpha^* - Y||_2^2 + \frac{\epsilon}{2} \tag{28}$$

$$= \frac{1}{m}||\bar{K}_\gamma \bar{\alpha}_\gamma - Y||_2^2 + \frac{\epsilon}{2} \tag{29}$$

$$= \frac{1}{m}||K\alpha_B - Y||_2^2 + \frac{\epsilon}{2} + \frac{\epsilon}{2} \tag{30}$$

$$= \min_{h\in H_\psi}\left(\frac{1}{m}\sum_{i=1}^{m}(h(\mathbf{x}_i) - y_i)^2\right) + \epsilon \tag{31}$$

Here 26 follows from the fact that since the output is in $[0, 1]$ clipping only reduces the loss, 27 follows from the precision used while compressing and since square loss is 2-Lipschitz, 28 follows from representing it in the matrix form, 29 follows since $\alpha^* = K_\gamma^{-1}\bar{K}_\gamma\bar{\alpha}_\gamma$ by definition, 30 follows from Theorem 12 with the given parameters satisfying the theorem for $\epsilon/2$ and lastly 31 follows from the definition of α_B. Thus, this gives us our result. □

6 Putting It All Together: From Compression to Learning

We now present our final algorithm: *Compressed Kernel Regression* (Algorithm 1). Note that the algorithm is efficient and takes at most $O(m^3)$ time.

For our learnability result, we restrict distributions to those that satisfy eigenvalue decay. More formally,

Definition 17 (Distribution Satisfying Eigenvalue Decay)**.** *Consider distribution $\mathcal{D}$ over $\mathcal{X}$ and kernel function k_ψ. Let $\mathcal{S}$ be a sample drawn i.i.d. from the distribution $\mathcal{D}$ and K be the empirical gram matrix corresponding to kernel function k_ψ on $\mathcal{S}$.*

- *$\mathcal{D}$ is said to satisfy (C, p, N)-polynomial eigenvalue decay if with probability $1-\delta$ over the drawn sample of size $m \geq N$, K/m satisfies (C, p)-polynomial eigenvalue decay.*
- *$\mathcal{D}$ is said to satisfy (C, N)-exponential eigenvalue decay if with probability $1-\delta$ over the drawn sample of size $m \geq N$, K/m satisfies C-exponential eigenvalue decay.*

Our main theorem proves generalization of the hypothesis output by Algorithm 1 for distributions satisfying eigenvalue decay in the above sense.

Algorithm 1 Compressed Kernel Regression

Input: Samples $\mathcal{S} = (\mathbf{x}_i, y_i)_{i=1}^m$, gram matrix K on $\mathcal{S}$, constants $\epsilon, \delta > 0$, norm bound B and maximum kernel function value M on $\mathcal{X}$.

1: Using RLS-Nyström Sampling [31] with input $(K_\gamma, \eta m)$ for $\gamma = \frac{\epsilon^3}{5832Bm}$ and $\eta = \frac{\epsilon^3}{5832B}$ compute $\bar{K}_\gamma = K_\gamma S(S^T K_\gamma S)^\dagger S^T K_\gamma$. Let $\mathcal{I}$ be the subsample corresponding to the columns selected using S. Note that the number of columns selected depends on the η effective dimension of K_γ.

2: Solve Optimization Problem 6 for $\lambda = \frac{\epsilon^2}{324B}$ to get $\bar{\alpha}_\gamma$ over $\mathcal{S}$

3: Compute $\alpha^* = S(S^T K_\gamma S)^\dagger S^T K_\gamma \bar{\alpha}_\gamma = \bar{K}_\gamma^{-1} \bar{K}_\gamma \bar{\alpha}_\gamma$

4: Compute $\tilde{\alpha}^*$ by truncating each entry of α^* up to precision $\frac{\epsilon}{4M|\mathcal{I}|}$

Output: $h_\mathcal{S}$ such that for all $x \in \mathcal{X}$, $h_\mathcal{S}(\mathbf{x}) = clip_{0,1}(\mathbf{w}^T \tilde{\alpha}^*)$ where $\mathbf{w}$ is a vector with entries $K(\mathbf{x}_i, \mathbf{x}) + \gamma m \mathbb{1}[\mathbf{x} = \mathbf{x}_i]$ for $i \in \mathcal{I}$ and 0 otherwise.

Theorem 18 (Formal for Theorem 1)**.** *Fix function class $\mathcal{C}$ with output bounded in $[0,1]$ and M-bounded kernel function k_ψ such that $\mathcal{C}$ is ϵ_0-approximated by $H_\psi = \{\boldsymbol{x} \to \langle \boldsymbol{v}, \psi(\boldsymbol{x})\rangle | \boldsymbol{v} \in \mathcal{K}_\psi, \langle \boldsymbol{v}, \boldsymbol{v}\rangle \le B\}$ for some ψ, B. Consider a sample $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)_{i=1}^m\}$ drawn i.i.d. from $\mathcal{D}$ on $\mathcal{X} \times [0,1]$. There exists an algorithm $\mathcal{A}$ that outputs hypothesis $h_\mathcal{S} = \mathcal{A}(\mathcal{S})$, such that,*

1. *If $\mathcal{D}_\mathcal{X}$ satisfies (C, p, m)-polynomial eigenvalue decay with probability $1 - \delta/4$ then with probability $1 - \delta$ for $m = \tilde{O}((CB)^{1/p} \log(M)/\epsilon^{2+3/p})$,*

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}(h_\mathcal{S}(\boldsymbol{x}) - y)^2 \le \min_{c\in\mathcal{C}} \left(\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}(c(\boldsymbol{x}) - y)^2\right) + 2\epsilon_0 + \epsilon$$

2. *If $\mathcal{D}_\mathcal{X}$ satisfies (C, m)-exponential eigenvalue decay with probability $1 - \delta/4$ then with probability $1 - \delta$ for $m = \tilde{O}(\log CB \log(M)/\epsilon^2)$,*

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}(h_\mathcal{S}(\boldsymbol{x}) - y)^2 \le \min_{c\in\mathcal{C}} \left(\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}(c(\boldsymbol{x}) - y)^2\right) + 2\epsilon_0 + \epsilon$$

Algorithm $\mathcal{A}$ runs in time $\mathsf{poly}(m, n)$.

Proof. Since $\mathcal{C}$ is ϵ_0-approximated by H_ψ we have,

$$\min_{h\in H_\psi} \left(\frac{1}{m}\sum_{i=1}^m (h(\mathbf{x}_i) - y_i)^2\right) \le \min_{c\in\mathcal{C}} \left(\frac{1}{m}\sum_{i=1}^m (c(\mathbf{x}_i) - y_i)^2\right) + 2\epsilon_0 \le \frac{1}{m}\sum_{i=1}^m (c^*(\mathbf{x}_i) - y_i)^2 + 2\epsilon_0$$

where $c^* \in \mathcal{C}$ be such that it minimizes $\mathbb{E}_{(x,y)\sim\mathcal{D}}(c(x) - y)^2$ over all $c \in \mathcal{C}$. The first inequality follows from square loss being 2-Lipschitz and the last inequality follows from c^* being a feasible solution.

Let K be the empirical gram matrix corresponding to k_ψ on $\mathcal{S}$. Let $h_\mathcal{S}$ be the hypothesis output by Algorithm 1 with input $(\mathcal{S}, K, \epsilon_1, \delta/4, B, M)$ for $\epsilon_1 > 0$ chosen later. From Theorem 16 with probability $1 - \delta/4$, we have

$$\frac{1}{m}\sum_{i=1}^m (h_\mathcal{S}(\mathbf{x}_i) - y_i)^2 \le \min_{h\in H_\psi} \left(\frac{1}{m}\sum_{i=1}^m (h(\mathbf{x}_i) - y_i)^2\right) + \epsilon_1.$$

We know that for every $c \in \mathcal{C}$, the square loss is bounded by 1, thus using Chernoff-Hoeffding inequality, with probability $1 - \delta/4$, we have

$$\frac{1}{m}\sum_{i=1}^{m}(c^*(\mathbf{x}_i) - y_i)^2 \leq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}(c^*(\mathbf{x}) - y)^2 + \epsilon_2$$

where $\epsilon_2 = \sqrt{\frac{\log(4/\delta)}{2m}}$.

Now the output of $h_{\mathcal{S}}$ lies in $[0,1]$ thus for all $(\mathbf{x}, y)$, $(y - h_{\mathcal{S}}(\mathbf{x}))^2$ lies in $[0,1]$. Thus viewing $h_{\mathcal{S}}$ as the output of the compression scheme (κ, ρ) of size k (Theorem 16), by Theorem 7, we have with probability $1 - \delta/4$,

$$\left|\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}(h_{\mathcal{S}}(\mathbf{x}) - y)^2 - \frac{1}{m}\sum_{i=1}^{m}(h_S(\mathbf{x}_i) - y_i)^2\right| \leq \sqrt{\frac{\epsilon_3}{m}\sum_{i=1}^{m}(h_{\mathcal{S}}(\mathbf{x}_i) - y_i)^2} + \epsilon_3 \leq \epsilon_3 + \sqrt{\epsilon_3} \leq 2\sqrt{\epsilon_3}$$

where $\epsilon_3 = 50 \cdot \frac{k\log(m/k) + \log(4/\delta)}{m}$.

Combining the above, we have with probability $1 - \delta$,

$$\begin{aligned}
\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}(h_{\mathcal{S}}(\mathbf{x}) - y)^2 &\leq \frac{1}{m}\sum_{i=1}^{m}(h_S(\mathbf{x}_i) - y_i)^2 + 2\sqrt{\epsilon_3} && (32)\\
&\leq \min_{h\in H_\psi}\left(\frac{1}{m}\sum_{i=1}^{m}(h(\mathbf{x}_i) - y_i)^2\right) + \epsilon_1 + 2\sqrt{\epsilon_3} && (33)\\
&\leq \frac{1}{m}\sum_{i=1}^{m}(c^*(\mathbf{x}_i) - y_i)^2 + 2\epsilon_0 + \epsilon_1 + 2\sqrt{\epsilon_3} && (34)\\
&\leq \min_{c\in\mathcal{C}}\left(\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}(c(\mathbf{x}) - y)^2\right) + 2\epsilon_0 + \epsilon_1 + \epsilon_2 + 2\sqrt{\epsilon_3} && (35)
\end{aligned}$$

Using Theorem 14 we can bound k depending on the different eigenvalue decay assumption. Now we set $\epsilon_1 = \epsilon/3$ and substituting for m. Recall that ϵ_2 and ϵ_3 are functions of m and for the chosen m, they are bounded by $\epsilon/3$ giving us the desired bound. Since Algorithm 1 runs in time $\mathsf{poly}(\mathsf{m}, \mathsf{n})$ we get the required time complexity. □

Remark: The above theorem can be extended to different rates of eigenvalue decay. For example, it can be shown that *finite* rank r would give a bound independent of B but dependent instead on r. Also, as in the proof of Theorem 14, it suffices for the eigenvalue decay to hold only for i sufficiently large.

7 Learning Neural Networks

Here we apply our main theorem to the problem of learning neural networks. For technical definitions of neural networks, we refer the reader to [47]. We define the class of neural networks as follows.

Definition 19 (Class of Neural Networks [16])**.** *Let $\mathcal{N}[\sigma, D, W, T]$ be the class of fully-connected, feed-forward networks with D hidden layers, activation function σ and quantities W and T described as follows:*

1. *Weight vectors in layer* 0 *have* 2*-norm bounded by* T*.*

2. *Weight vectors in layers* $1, \ldots, D$ *have* 1*-norm bounded by* W*.*

3. *For each hidden unit* $\sigma(\boldsymbol{w} \cdot \boldsymbol{z})$ *in the network, we have* $|\boldsymbol{w} \cdot \boldsymbol{z}| \leq T$ *(by* $\boldsymbol{z}$ *we denote the input feeding into unit* σ *from the previous layer).*

We consider activation functions $\sigma_{relu}(x) = \max(0, x)$ and $\sigma_{sig} = \frac{1}{1+e^{-x}}$, though other activation functions fit within our framework. Goel et al. [16] showed that the class of ReLUs/Sigmoids along with their compositions can be approximated by linear functions in a high dimensional Hilbert space (corresponding to a particular type of polynomial kernel). We use the following theorem that follows directly from the structural results in [16] (and uses the composed-kernel technique of Zhang et al. [47]).

Theorem 20. *Consider the following hypothesis class* $\mathcal{H}_{\mathsf{MK}_d} = \{\boldsymbol{x} \to \langle \boldsymbol{v}, \psi(\boldsymbol{x})\rangle | \boldsymbol{v} \in \mathcal{K}_{\mathsf{MK}_d}, \langle \boldsymbol{v}, \boldsymbol{v}\rangle \leq B\}$ *where* $\mathcal{K}_{\mathsf{MK}_d}$ *is the Hilbert space corresponding to the Multinomial Kernel* [2] *and* ψ *is the corresponding feature vector. For* $D > 0$*, consider the composed class* $\mathcal{H}^{(D)} = \{\boldsymbol{x} \to \langle v, \psi^{(D)}(\boldsymbol{x})\rangle | \boldsymbol{v} \in \mathcal{K}^{(D)}, \langle \boldsymbol{v}, \boldsymbol{v}\rangle \leq B\}$ *where* $\psi^{(D)}$ *is the feature vector of the* D*-times composed kernel* $K^{(D)}$ [3]*. Then for* $\mathcal{X} = \mathbb{S}^{n-1}$*,*

1. ***Single ReLU****:* $\mathcal{C}_{relu} = \mathcal{N}[\sigma_{relu}, 0, \cdot, 1]$ *is* ϵ*-approximated by* $\mathcal{H}_d$ *for* $d = O(1/\epsilon)$ *and* $B = 2^{(\tau/\epsilon)}$ *with* $M = d + 1$*,*

2. ***Network of ReLUs****:* $\mathcal{C}_{relu-D} = \mathcal{N}[\sigma_{relu}, D, W, T]$ *is* ϵ*-approximated by* $\mathcal{H}_{(D)}$ *for* $B = 2^{(\tau W^D DT/\epsilon)^D}$ *with* $M = 2$*,*

3. ***Network of Sigmoids****:* $\mathcal{C}_{sig-D} = \mathcal{N}[\sigma_{sig}, D, W, T]$ *is* ϵ*-approximated by* $\mathcal{H}_{(D)}$ *for* $B = 2^{(\tau T \log(W^D D/\epsilon))^D}$ *with* $M = 2$*,*

for some sufficiently large constant $\tau > 0$*.*

As mentioned earlier, the sample complexity of prior work depends linearly on B, which, for even a single ReLU, is exponential in $1/\epsilon$. Assuming sufficiently strong eigenvalue decay, we can show that we can obtain fully polynomial time algorithms for the above classes.

Theorem 21. *For* $\epsilon, \delta > 0$*, consider* $\mathcal{D}$ *on* $\mathbb{S}^{n-1} \times [0, 1]$ *such that,*

1. *For* $\mathcal{C}_{relu}$*,* $\mathcal{D}_{\mathcal{X}}$ *satisfies* (C, p, m)*-polynomial eigenvalue decay for* $p \geq \xi/\epsilon$*,*

2. *For* $\mathcal{C}_{relu-D}$*,* $\mathcal{D}_{\mathcal{X}}$ *satisfies* (C, p, m)*-polynomial eigenvalue decay for* $p \geq (\xi W^D DT/\epsilon)^D$*,*

3. *For* $\mathcal{C}_{sig-D}$*,* $\mathcal{D}_{\mathcal{X}}$ *satisfies* (C, p, m)*-polynomial eigenvalue decay for* $p \geq (\xi T \log(W^D D/\epsilon)))^D$*,*

where $\mathcal{D}_{\mathcal{X}}$ *is the marginal distribution on* $\mathcal{X} = \mathbb{S}^{n-1}$*,* $\xi > 0$ *is some sufficiently large constant and* $C \leq (n \cdot 1/\epsilon)^{\zeta p}$ *for some constant* $\zeta > 0$*. The value of* m *is obtained from Theorem 18 as* $m = \tilde{O}((CB)^{1/p} \log(M)/\epsilon^{2+3/p})$ *where the values of* B, M *are derived from Theorem 20.*

Each decay assumption above implies an algorithm for agnostically learning the corresponding class on $\mathbb{S}^{n-1} \times [0, 1]$ *with respect to the square loss in time* $\mathsf{poly}(n, 1/\epsilon, \log(1/\delta))$*.*

[2] The multinomial kernel defined by [16] is $\mathsf{MK}_d(\mathbf{x}, \mathbf{x}') = \sum_{i=0}^{d} (\mathbf{x} \cdot \mathbf{x}')^i$.

[3] [47] defined kernel $K^{(1)}(\mathbf{x}, \mathbf{x}\prime) = \frac{1}{2-(\mathbf{x} \cdot \mathbf{x}\prime)}$. The corresponding composed kernel function is defined as $K^{(D)}(\mathbf{x}, \mathbf{x}\prime) = \frac{1}{2-K^{(D-1)}(\mathbf{x}, \mathbf{x}\prime)}$

Proof. The proof follows from applying Theorem 18 to the appropriate kernel from Theorem 20 and substituting the corresponding eigenvalue decays to compute the sample size needed by Algorithm 1 for learnability. For example, for the case of single ReLU, $M = \mathsf{poly}(1/\epsilon)$, $B = 2^{(\tau/\epsilon)}$ and we take $p \geq \xi/\epsilon$. So for any $C = (n \cdot 1/\epsilon)^{\zeta p}$, we obtain sample complexity $m = \tilde{O}((C2^{(\tau/\epsilon)})^{1/p} \log(M)/\epsilon^{2+3/p}) = \mathsf{poly}(n, 1/\epsilon)$. Since the algorithm takes time at most $\mathsf{poly}(m, n)$, we obtain the required result. □

Note that assuming an exponential eigenvalue decay (stronger than polynomial) will result in efficient learnability for much broader classes of networks.

Since it is not known how to agnostically learn even a single ReLU with respect to arbitrary distributions on $\mathbb{S}^{n-1}$ in polynomial-time[4], much less a network of ReLUs, we state the following corollary highlighting the decay we require to obtain efficient learnability for simple networks:

Corollary 22 (Restating Corollary 2)**.** *Let $\mathcal{C}$ be the class of all fully-connected networks of ReLUs with one-hidden layer of size ℓ feeding into a final output ReLU activation where the 2-norms of all weight vectors are bounded by 1. Then, (suppressing the parameter m for simplicity), assuming $(C, i^{-\ell/\epsilon})$-polynomial eigenvalue decay for $C = \mathsf{poly}(n, 1/\epsilon, \ell)$, $\mathcal{C}$ is learnable in polynomial time with respect to square loss on $\mathbb{S}^{n-1}$. If ReLU is replaced with sigmoid, then we require eigenvalue decay of $i^{-\sqrt{\ell}\log(\sqrt{\ell}/\epsilon)}$.*

Proof. By assumption the 2-norm of each weight vector is bounded by 1, which implies that the 1-norm of the weight vector to the one hidden unit at layer two is at most $\sqrt{\ell}$. Also observe that, the maximum 2-norm of any input vector $\mathbf{z}$ to a hidden unit with weight vector $\mathbf{w}$ is bounded by $\sqrt{\ell}$ hence $|\mathbf{w} \cdot \mathbf{x}| \leq \sqrt{\ell}$. Using these properties we can apply Theorem 21 with parameters $W = \sqrt{\ell}$, $T = \sqrt{\ell}$ and $D = 1$ to obtain the required result. □

8 Conclusions and Future Work

We have proposed the first set of distributional assumptions that guarantee fully polynomial-time algorithms for learning expressive classes of neural networks (without restricting the structure of the network). The key abstraction was that of a *compression scheme* for kernel approximations, specifically Nyström sampling. We proved that eigenvalue decay of the Gram matrix reduces the dependence on the norm B in the kernel regression problem.

Prior distributional assumptions, such as the underlying marginal equaling a Gaussian, neither lead to fully polynomial-time algorithms nor are representative of real-world data sets[5]. Eigenvalue decay, on the other hand, has been observed in practice and does lead to provably efficient algorithms for learning neural networks.

A natural criticism of our assumption is that the rate of eigenvalue decay we require is too strong. In some cases, especially for large depth networks with many hidden units, this may be true[6]. Note, however, that our results show that even moderate eigenvalue decay will

[4]Goel et al. [16] show that agnostically learning a single ReLU over $\{-1, 1\}^n$ is as hard as learning sparse parities with noise. This reduction can be extended to the case of distributions over $\mathbb{S}^{n-1}$ [3].

[5]Despite these limitations, we still think uniform or Gaussian assumptions are worthwhile and have provided highly nontrivial learning results.

[6]It is useful to keep in mind that agnostically learning even a single ReLU with respect to all distributions seems computationally intractable, and that our required eigenvalue decay in this case is only a function of the accuracy parameter ϵ.

lead to improved algorithms. Further, it is quite possible our assumptions can be relaxed. An obvious question for future work is what is the minimal rate of eigenvalue decay needed for efficient learnability? Another direction would be to understand how these eigenvalue decay assumptions relate to other distributional assumptions.

Acknowledgements. We would like to thank Misha Belkin and Nikhil Srivastava for very helpful conversations regarding kernel ridge regression and eigenvalue decay. We also thank Daniel Hsu, Karthik Sridharan, and Justin Thaler for useful feedback. The analogy between eigenvalue decay and power-law graphs is due to Raghu Meka.

References

[1] Peter Auer, Mark Herbster, and Manfred K. Warmuth. Exponentially many local minima for single neurons. In *Advances in Neural Information Processing Systems*, volume 8, pages 316–322. The MIT Press, 1996.

[2] Haim Avron, Huy Nguyen, and David Woodruff. Subspace embeddings for the polynomial kernel. In *Advances in Neural Information Processing Systems*, pages 2258–2266, 2014.

[3] Peter Bartlett, Daniel Kane, and Adam Klivans. personal communication.

[4] Peter L. Bartlett, Olivier Bousquet, and Shahar Mendelson. Local rademacher complexities. 33(4), August 16 2005.

[5] Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

[6] Pawel Brach, Marek Cygan, Jakub Lacki, and Piotr Sankowski. Algorithmic complexity of power law networks. *CoRR*, abs/1507.02426, 2015.

[7] Alon Brutzkus and Amir Globerson. Globally optimal gradient descent for a convnet with gaussian inputs. *CoRR*, abs/1702.07966, 2017.

[8] Anna Choromanska, Mikael Henaff, Michaël Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *AISTATS*, volume 38 of *JMLR Workshop and Conference Proceedings*. JMLR.org, 2015.

[9] Andrew Cotter, Shai Shalev-Shwartz, and Nati Srebro. Learning optimally sparse support vector machines. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 266–274, 2013.

[10] Amit Daniely. Complexity theoretic limitations on learning halfspaces. In *STOC*, pages 105–117. ACM, 2016.

[11] Amit Daniely. SGD learns the conjugate kernel class of the network. *CoRR*, abs/1702.08503, 2017.

[12] Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *NIPS*, pages 2253–2261, 2016.

[13] Ofir David, Shay Moran, and Amir Yehudayoff. On statistical learning via the lens of compression. *arXiv preprint arXiv:1610.03592*, 2016.

[14] Petros Drineas and Michael W Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *journal of machine learning research*, 6(Dec):2153–2175, 2005.

[15] Petros Drineas, Michael W Mahoney, and S Muthukrishnan. Relative-error cur matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, 2008.

[16] Surbhi Goel, Varun Kanade, Adam Klivans, and Justin Thaler. Reliably learning the relu in polynomial time. *arXiv preprint arXiv:1611.10258*, 2016.

[17] David Haussler. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Inf. Comput.*, 100(1):78–150, 1992.

[18] Majid Janzamin, Hanie Sedghi, and Anima Anandkumar. Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. *arXiv preprint arXiv:1506.08473*, 2015.

[19] Sham M Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in neural information processing systems*, pages 793–800, 2009.

[20] Kenji Kawaguchi. Deep learning without poor local minima. In *NIPS*, pages 586–594, 2016.

[21] Michael J. Kearns, Robert E. Schapire, and Linda M. Sellie. Toward efficient agnostic learning. *Mach. Learn.*, 17(2-3):115–141, 1994.

[22] Adam R. Klivans and Pravesh Kothari. Embedding hard learning problems into gaussian space. In *APPROX-RANDOM*, volume 28 of *LIPIcs*, pages 793–809. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014.

[23] Adam R. Klivans and Raghu Meka. Moment-matching polynomials. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:8, 2013.

[24] Adam R. Klivans and Alexander A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. *J. Comput. Syst. Sci*, 75(1):2–12, 2009.

[25] Anton Krohmer. Finding Cliques in Scale-Free Networks. Master's thesis, Saarland University, Germany, 2012.

[26] Dima Kuzmin and Manfred K. Warmuth. Unlabeled compression schemes for maximum classes. *Journal of Machine Learning Research*, 8:2047–2081, 2007.

[27] Nick Littlestone and Manfred Warmuth. Relating data compression and learnability. Technical report, Technical report, University of California, Santa Cruz, 1986.

[28] Nick Littlestone and Manfred Warmuth. Relating data compression and learnability. Technical report, 1986.

[29] Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. In *Advances in Neural Information Processing Systems*, pages 855–863, 2014.

[30] Siyuan Ma and Mikhail Belkin. Diving into the shallows: a computational perspective on large-scale shallow learning. *CoRR*, abs/1703.10622, 2017.

[31] Cameron Musco and Christopher Musco. Recursive sampling for the nyström method. *arXiv preprint arXiv:1605.07583*, 2016.

[32] B. Schölkopf, J. Shawe-Taylor, AJ. Smola, and RC. Williamson. Generalization bounds via eigenvalues of the gram matrix. Technical Report 99-035, NeuroCOLT, 1999.

[33] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[34] Hanie Sedghi and Anima Anandkumar. Provable methods for training neural networks with sparse connectivity. *arXiv preprint arXiv:1412.2693*, 2014.

[35] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[36] Ohad Shamir. The sample complexity of learning linear predictors with the squared loss. *Journal of Machine Learning Research*, 16:3475–3486, 2015.

[37] Ohad Shamir. Distribution-specific hardness of learning neural networks. *arXiv preprint arXiv:1609.01037*, 2016.

[38] John Shawe-Taylor, Christopher KI Williams, Nello Cristianini, and Jaz Kandola. On the eigenspectrum of the gram matrix and the generalization error of kernel-pca. *IEEE Transactions on Information Theory*, 51(7):2510–2522, 2005.

[39] Le Song, Santosh Vempala, John Wilmes, and Bo Xie. On the complexity of learning neural networks. *arXiv preprint arXiv:1707.04615*, 2017.

[40] Daniel Soudry and Yair Carmon. No bad local minima: Data independent training error guarantees for multilayer neural networks. *CoRR*, abs/1605.08361, 2016.

[41] Ameet Talwalkar and Afshin Rostamizadeh. Matrix coherence and the nystrom method. *CoRR*, abs/1408.2044, 2014.

[42] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27:1134–1142, 1984.

[43] Christopher KI Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, pages 661–667. MIT press, 2000.

[44] Bo Xie, Yingyu Liang, and Le Song. Diversity leads to generalization in neural networks. *CoRR*, abs/1611.03131, 2016.

[45] Qiuyi Zhang, Rina Panigrahy, and Sushant Sachdeva. Electron-proton dynamics in deep learning. *CoRR*, abs/1702.00458, 2017.

[46] Tong Zhang. Effective dimension and generalization of kernel learning. In *Advances in Neural Information Processing Systems*, pages 471–478, 2003.

[47] Yuchen Zhang, Jason D Lee, and Michael I Jordan. l1-regularized neural networks are improperly learnable in polynomial time. In *International Conference on Machine Learning*, pages 993–1001, 2016.

[48] Yuchen Zhang, Jason D. Lee, Martin J. Wainwright, and Michael I. Jordan. Learning halfspaces and neural networks with random initialization. *CoRR*, abs/1511.07948, 2015.