Superpolynomial Lower Bounds for Learning One-Layer Neural Networks using Gradient Descent

Surbhi Goel [1] Aravind Gollakota [1] Zhihan Jin [2] Sushrut Karmalkar [1] Adam Klivans [1]

Abstract

We give the first superpolynomial lower bounds for learning one-layer neural networks with respect to the Gaussian distribution for a broad class of algorithms. We prove that gradient descent run on any classifier with respect to square loss will fail to achieve small test error in polynomial time. Prior work held only for gradient descent run with small batch sizes and sufficiently smooth classifiers. For classification, we give a stronger result, namely that any statistical query (SQ) algorithm (including gradient descent) will fail to achieve small test error in polynomial time. Our lower bounds hold for commonly used activations such as ReLU and sigmoid. The core of our result relies on a novel construction of a simple family of neural networks that are exactly orthogonal with respect to all spherically symmetric distributions.

1. Introduction

A major challenge in the theory of deep learning is to understand when gradient descent can efficiently learn simple families of neural networks. The associated optimization problem is nonconvex and well known to be computationally intractable in the worst case. For example, cyphertexts from public-key cryptosystems can be encoded into a training set labeled by simple neural networks (Klivans & Sherstov, 2009), implying that the corresponding learning problem is as hard as breaking cryptographic primitives. These hardness results, however, rely on discrete representations and produce relatively unrealistic joint distributions.

Our Results. In this paper we give the first superpolynomial lower bounds for learning neural networks using gradient descent in arguably the simplest possible setting: we assume the marginal distribution is a spherical Gaussian, the labels are noiseless and are exactly equal to the output of a one-layer neural network (a linear combination of say ReLU or sigmoid activations), and the goal is to output a classifier whose test error (measured by square-loss) is small. We prove—unconditionally—that gradient descent fails to produce a classifier with small square-loss if it is required to run in polynomial time in the dimension. Our lower bound depends only on the algorithm used (gradient descent) and not on the architecture of the underlying classifier. That is, our results imply that current popular heuristics such as running gradient descent on an overparameterized network (for example, working in the NTK regime (Jacot et al., 2018)) will require superpolynomial time to achieve small test error.

Statistical Queries. We prove our lower bounds in the now well-studied statistical query (SQ) model of (Kearns, 1998) that captures most learning algorithms used in practice. For a loss function ℓ and a hypothesis h_θ parameterized by θ, the true population loss with respect to joint distribution D on $X \times Y$ is given by $\mathbb{E}_{(x,y)\sim D}[\ell(h_\theta(x), y)]$, and the gradient with respect to θ is given by $\mathbb{E}_{(x,y)\sim D}[\ell'(h_\theta(x), y)\nabla_\theta h_\theta(x)]$. In the SQ model, we specify a query function $\phi(x, y)$ and receive an estimate of $|\mathbb{E}_{(x,y)\sim D}[\phi(x, y)]|$ to within some tolerance parameter τ. An important special class of queries are correlational or inner-product queries, where the query function ϕ is defined only on X and we receive an estimate of $|\mathbb{E}_{(x,y)\sim D}[\phi(x) \cdot y]|$ within some tolerance τ. It is not difficult to see that (1) the gradient of a population loss can be approximated to within τ using statistical queries of tolerance τ and (2) for square-loss only inner-product queries are required.

Since the convergence analysis of gradient descent holds given sufficiently strong approximations of the gradient, lower bounds for learning in the SQ model (Kearns, 1998; Blum et al., 1994; Szörényi, 2009; Feldman, 2012; 2017) directly imply unconditional lower bounds on the running time for gradient descent to achieve small error. We give

[1] Department of Computer Science, University of Texas at Austin [2] Department of Computer Science, Shanghai Jiao Tong University. Correspondence to: Aravind Gollakota <aravindg@cs.utexas.edu>, Adam Klivans <klivans@cs.utexas.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

the first superpolynomial lower bounds for learning one-layer networks with respect to any Gaussian distribution for any SQ algorithm that uses inner product queries:

Theorem 1.1 (informal). *Let $\mathcal{C}$ be a class of real-valued concepts defined by one-layer single-output neural networks with input dimension n and m hidden units (ReLU or sigmoid); i.e., functions of the form $f(x) = \sum_{i=1}^{m} a_i \sigma(w_i \cdot x)$. Then learning $\mathcal{C}$ under the standard Gaussian $\mathcal{N}(0, I_n)$ in the SQ model with inner-product queries requires $n^{\Omega(\log m)}$ queries for any tolerance $\tau = n^{-\Omega(\log m)}$.*

In particular, this rules out any approach for learning one-layer neural networks in polynomial-time that performs gradient descent on any polynomial-size classifier with respect to square-loss or logistic loss. For classification, we obtain significantly stronger results and rule out general SQ algorithms that run in polynomial-time (e.g., gradient descent with respect to any polynomial-size classifier and any polynomial-time computable loss). In this setting, our labels are $\{\pm 1\}$ and correspond to the *softmax* of an unknown one-layer neural network. We prove the following:

Theorem 1.2 (informal). *Let $\mathcal{C}$ be a class of real-valued concepts defined by a one-layer neural network in n dimensions with m hidden units (ReLU or sigmoid) feeding into any odd, real-valued output node with range $[-1, 1]$. Let D' be a distribution on $\mathbb{R}^n \times \{\pm 1\}$ such that the marginal on $\mathbb{R}^n$ is the standard Gaussian $\mathcal{N}(0, I_n)$, and $\mathbb{E}[Y|X] = c(X)$ for some $c \in \mathcal{C}$. For some $b, C > 0$ and $\epsilon = Cm^{-b}$, outputting a classifier $f : \mathbb{R}^n \to \{\pm 1\}$ with $\mathbb{P}_{(X,Y)\sim D'}[f(X) \neq Y] \leq 1/2 - \epsilon$ requires $n^{\Omega(\log m)}$ statistical queries of tolerance $n^{-\Omega(\log m)}$.*

The above lower bound for classification rules out the commonly used approach of training a polynomial-size, real-valued neural network using gradient descent (with respect to any polynomial-time computable loss) and then taking the sign of the output of the resulting network.

Our techniques. At the core of all SQ lower bounds is the construction of a family of functions that are pairwise approximately orthogonal with respect to the underlying marginal distribution. Typically, these constructions embed 2^n parity functions over the discrete hypercube $\{-1, 1\}^n$. Since parity functions are perfectly orthogonal, the resulting lower bound can be quite strong. Here we wish to give lower bounds for more natural families of distributions, namely Gaussians, and it is unclear how to embed parity.

Instead, we use an alternate construction. For activation functions $\phi, \psi : \mathbb{R} \to \mathbb{R}$, define

$$f_S(x) = \psi\left(\sum_{w\in\{-1,1\}^k} \chi(w)\phi\left(\frac{w \cdot x_S}{\sqrt{k}}\right)\right).$$

Enumerating over every $S \subseteq [n]$ of size k gives a family of functions of size $n^{O(k)}$. Here x_S denotes the vector of x_i for $i \in S$ (typically we choose $k = \log m$ to produce a family of one-layer neural networks with m hidden units). Each of the $2^k = m$ inner weight vectors are all of unit norm, and all of the m outer weights have absolute value one. Note also that our construction uses activations with zero bias term.

We give a complete characterization of the class of nonlinear activations for which these functions are orthogonal. In particular, the family is orthogonal for any activation with a nonzero Hermite coefficient of degree k or higher.

Apart from showing orthogonality, we must also prove that functions in these classes are nontrivial (i.e., are not exponentially close to the constant zero function). This reduces to proving certain lower bounds on the norms of one-layer neural networks. The analysis requires tools from Hermite and complex analysis.

SQ Lower Bounds for Real-Valued Functions. Another major challenge is that our function family is real-valued as opposed to boolean. Given an orthogonal family of (deterministic) *boolean* functions, it is straightforward to apply known results and obtain general SQ lower bounds for learning with respect to $0/1$ loss. For the case of real-valued functions, the situation is considerably more complicated. For example, the class of orthogonal Hermite polynomials on n variables of degree d has size $n^{O(d)}$, yet there *is* an SQ algorithm due to (Andoni et al., 2014) that learns this class with respect to the Gaussian distribution in time $2^{O(d)}$. More recent work due to (Andoni et al., 2019) shows that Hermite polynomials can be learned by an SQ algorithm in time polynomial in n and $\log d$.

As such, it is *impossible* to rule out general polynomial-time SQ algorithms for learning real-valued functions based solely on orthogonal function families. Fortunately, it is not difficult to see that the SQ reductions due to (Szörényi, 2009) hold in the real-valued setting as long as the learning algorithm uses only *inner-product queries* (and the norms of the functions are sufficiently large). Since performing gradient descent with respect to square-loss or logistic loss can be implemented using inner-product queries, we obtain our first set of desired results[1].

Still, we would like rule out *general* SQ algorithms for learning simple classes of neural networks. To that end, we consider the *classification* problem for one-layer neural networks and output labels after performing a *softmax* on a one-layer network. Concretely, consider a distribution on $\mathbb{R}^n \times \{-1, 1\}$ where $\mathbb{E}[Y|X] = \sigma(c(X))$ for some

[1] The algorithms of (Andoni et al., 2014) and (Andoni et al., 2019) do not use inner-product queries.

$c \in \mathcal{C}$ and $\sigma : \mathbb{R} \to [-1, 1]$ (for example, σ could be tanh). We describe two goals. The first is to estimate the conditional mean function, i.e., output a classifier h such that $\mathbb{E}[(h(x) - c(x))^2] \leq \epsilon$. The second is to directly minimize classification loss, i.e., output a boolean classifier h such that $\mathbb{P}_{X,Y\sim D}[h(X) \neq Y] \leq 1/2 - \epsilon$.

We give superpolynomial lower bounds for both of these problems in the general SQ model by making a new connection to *probabilistic concepts*, a learning model due to (Kearns & Schapire, 1994). Our key theorem gives a superpolynomial SQ lower bound for the problem of *distinguishing* probabilistic concepts induced by our one-layer neural networks from truly random labels. A final complication we overcome is that we must prove orthogonality and norm bounds on one-layer neural networks that have been composed with a nonlinear activation (e.g., tanh).

SGD and Gradient Descent Plus Noise. It is easy to see that our results also imply lower bounds for algorithms where the learner adds noise to the estimate of the gradient (e.g., Langevin dynamics). On the other hand, for technical reasons, it is known that SGD is *not* a statistical query algorithm (because it examines training points individually) and does not fall into our framework. That said, recent work by (Abbe & Sandon, 2020) shows that SGD is *universal* in the sense that it can encode all polynomial-time learners. This implies that proving unconditional lower bounds for SGD would give a proof that $\mathsf{P} \neq \mathsf{NP}$. Thus, we cannot hope to prove unconditional lower bounds on SGD (unless we can prove $\mathsf{P} \neq \mathsf{NP}$).

Independent Work. Independently, Diakonikolas et al. (Diakonikolas et al., 2020) have given stronger correlational SQ lower bounds for the same class of functions with respect to the Gaussian distribution. Their bounds are exponential in the number of hidden units while ours are quasipolynomial. We can plug in their result and obtain exponential general SQ lower bounds for the associated probabilistic concept using our framework.

Related Work. There is a large literature of results proving hardness results (or unconditional lower bounds in some cases) for learning various classes of neural networks (Blum & Rivest, 1989; Vu, 1998; Klivans & Sherstov, 2009; Livni et al., 2014; Goel et al., 2017).

The most relevant prior work is due to (Song et al., 2017), who addressed learning one-layer neural networks under logconcave distributions using Lipschitz queries. Specifically, let n be the input dimension, and let m be the number of hidden s-Lipschitz sigmoid units. For $m = \tilde{O}(s\sqrt{n})$, they construct a family of neural networks such that any learner using λ-Lipschitz queries with tolerance greater than $\Omega(1/(s^2 n))$ needs at least $2^{\Omega(n)}/(\lambda s^2)$ queries. Roughly speaking, their lower bounds hold for λ-Lipschitz queries due to the composition of their one-layer neural networks with a δ-function in order make the family more "boolean." Because of their restriction on the tolerance parameter, they cannot rule out gradient descent with large batch sizes. Further, the slope of the activations they require in their constructions scales inversely with the Lipschitz and tolerance parameters.

To contrast with (Song et al., 2017), note that our lower bounds hold for any inverse-polynomial tolerance parameter (i.e., *will* hold for polynomially-large batch sizes), do not require a Lipschitz constraint on the queries, and use only standard 1-Lipschitz ReLU and/or sigmoid activations (with zero bias) for the construction of the hard family. Our lower bounds are typically quasipolynomial in the number of hidden units; improving this to an exponential lower bound is an interesting open question. Both of our models capture square-loss and logistic loss.

In terms of techniques, (Song et al., 2017) build an orthogonal function family using univariate, periodic "wave" functions. Our construction takes a different approach, adding and subtracting activation functions with respect to overlapping "masks." Finally, aside from the (black-box) use of a theorem from complex analysis, our construction and analysis are considerably simpler than the proof in (Song et al., 2017).

A follow-up work (Vempala & Wilmes, 2019) gave SQ lower bounds for learning classes of degree d orthogonal polynomials in n variables with respect to the uniform distribution on the unit sphere (as opposed to Gaussians) using inner product queries of bounded tolerance (roughly $1/n^d$). To obtain superpolynomial lower bounds, each function in the family requires superpolynomial description length (their polynomials also take on very small values, $1/n^d$, with high probability).

Shamir (Shamir, 2018) (see also the related work of (Shalev-Shwartz et al., 2017)) proves hardness results (and lower bounds) for learning neural networks using gradient descent with respect to square-loss. His results are separated into two categories: (1) hardness for learning "natural" target families (one layer ReLU networks) or (2) lower bounds for "natural" input distributions (Gaussians). We achieve lower bounds for learning problems with *both* natural target families and natural input distributions. Additionally, our lower bounds hold for any nonlinear activations (as opposed to just ReLUs) and for broader classes of algorithms (SQ).

Recent work due to (Goel et al., 2019) gives hardness results for learning a ReLU with respect to Gaussian distributions. Their results require the learner to output a single ReLU as its output hypothesis and require the learner to succeed in the agnostic model of learning. (Klivans & Kothari, 2014) prove hardness results for learning a threshold function with respect to Gaussian distributions, but they also require the learner to succeed in the agnostic model. Very recent work due to Daniely and Vardi (Daniely & Vardi, 2020) gives hardness results for learning randomly chosen two-layer networks. The hard distributions in their case are not Gaussians, and they require a nonlinear clipping output activation.

Positive Results. Many recent works give algorithms for learning one-layer ReLU networks using gradient descent with respect to Gaussians under various assumptions (Zhong et al., 2017; Zhang et al., 2017; Brutzkus & Globerson, 2017; Zhang et al., 2019) or use tensor methods (Janzamin et al., 2015; Ge et al., 2018). These results depend on the hidden weight vectors being sufficiently orthogonal, or the coefficients in the second layer being positive, or both. Our lower bounds explain why these types of assumptions are necessary.

2. Preliminaries

We use $[n]$ to denote the set $\{1, \dots, n\}$, and $S \subseteq_k T$ to indicate that S is a k-element subset of T. We denote euclidean inner products between vectors u and v by $u \cdot v$. We denote the element-wise product of vectors u and v by $u \circ v$, that is, $u \circ v$ is the vector $(u_1 v_1, \dots, u_n v_n)$.

Let X be an arbitrary domain, and let D be a distribution on X. Given two functions $f, g : X \to \mathbb{R}$, we define their L_2 inner product with respect to D to be $\langle f, g \rangle_D = \mathbb{E}_D[fg]$. The corresponding L_2 norm is given by $\|f\|_D = \sqrt{\langle f, f \rangle_D} = \sqrt{\mathbb{E}_D[f^2]}$.

A real-valued concept on $\mathbb{R}^n$ is a function $c : \mathbb{R}^n \to \mathbb{R}$. We denote the induced labeled distribution on $\mathbb{R}^n \times \mathbb{R}$, i.e. the distribution of $(x, c(x))$ for $x \sim D$, by D_c. A probabilistic concept, or p-concept, on X is a concept that maps each point x to a random $\{\pm 1\}$-valued label in such a way that $\mathbb{E}[Y|X] = c(X)$ for a fixed function $c : \mathbb{R}^n \to [-1, 1]$, known as the conditional mean function. Given a distribution D on the domain, we abuse D_c to denote the induced labeled distribution on $X \times \{\pm 1\}$ such that the marginal distribution on $\mathbb{R}^n$ is D and $\mathbb{E}[Y|X] = c(X)$ (equivalently the label is $+1$ with probability $\frac{1+c(x)}{2}$ and -1 otherwise).

The SQ model A statistical query is specified by a query function $h : \mathbb{R}^d \times Y \to \mathbb{R}$. The SQ model allows access to an SQ oracle that accepts a query h of specified tolerance τ, and responds with a value in $[\mathbb{E}_{x,y}[h(x, y)] - \tau, \mathbb{E}_{x,y}[h(x, y)] + \tau]$.[2] To disallow arbitrary scaling, we will require that for each y, the function $x \mapsto h(x, y)$ has norm at most 1. In the real-valued setting, a query h is called a correlational or inner product query if it is of the form $h(x, y) = g(x) \cdot y$ for some function g, so that $\mathbb{E}_{D_c}[h] = \mathbb{E}_D[gc] = \langle g, c \rangle_D$. Here we assume $\|g\| \leq 1$ when stating lower bounds, again to disallow arbitrary scaling.

Gradient descent with respect to squared loss is captured by inner product queries, since the gradient is given by

$$\begin{aligned} \mathbb{E}_{x,y}[\nabla_\theta (h_\theta(x) - y)^2] &= \mathbb{E}_{x,y}[2(h_\theta(x) - y)\nabla_\theta h_\theta(x)] \\ &= 2\mathbb{E}_x[h_\theta(x)\nabla_\theta h_\theta(x)] \\ &\quad - 2\mathbb{E}_{x,y}[y \nabla_\theta h_\theta(x)]. \end{aligned}$$

Here the first term can be estimated directly using knowledge of the distribution, while the latter is a vector each of whose elements is an inner product query.

We now formally define the learning problems we consider.

Definition 2.1 (SQ learning of real-valued concepts using inner product queries)**.** Let $\mathcal{C}$ be a class of p-concepts over a domain X, and let D be a distribution on X. We say that a learner learns $\mathcal{C}$ with respect to D up to L_2 error ϵ using inner product quiers (equivalently squared loss ϵ^2) if, given only SQ oracle access to D_c for some unknown $c \in \mathcal{C}$, and using only inner product queries, it is able to output $\tilde{c} : X \to [-1, 1]$ such that $\|c - \tilde{c}\|_D \leq \epsilon$.

For the classification setting, we consider two different notions of learning p-concepts. One is learning the target up to small L_2 error, to be thought of as a strong form of learning. The other, weaker form, is achieving a nontrivial inner product (i.e. unnormalized correlation) with the target. We prove lower bounds on both in order to capture different learning goals.

Definition 2.2 (SQ learning of p-concepts)**.** Let $\mathcal{C}$ be a class of p-concepts over a domain X, and let D be a distribution on X. We say that a learner learns $\mathcal{C}$ with respect to D up to L_2 error ϵ if, given only SQ oracle access to D_c for some unknown $c \in \mathcal{C}$, and using arbitrary queries, it is able to output $\tilde{c} : X \to [-1, 1]$ such that $\|c - \tilde{c}\|_D \leq \epsilon$. We say that a learner weakly learns $\mathcal{C}$ with respect to D with advantage ϵ if it is able to output $\tilde{c} : X \to [-1, 1]$ such that $\langle \tilde{c}, c \rangle_D \geq \epsilon$.

Note that the best achievable advantage is $\mathbb{E}_{x \sim D}[|c(x)|]$, achieved by $\tilde{c}(x) = \text{sign}(c(x))$. Note also that $\|c\|_D^2 \leq \mathbb{E}_D[|c|] \leq \|c\|_D$, and therefore a norm lower bound on functions in $\mathcal{C}$ implies an upper bound on the achievable advantage.

[2] In the SQ literature, this is referred to as the STAT oracle. A variant called VSTAT is also sometimes used, known to be equivalent up to small polynomial factors (Feldman, 2017). While it makes no substantive difference to our superpolynomial lower bounds, our arguments can be extended to VSTAT as well.

Remark 2.3 (Learning with L_2 error implies weak learning). If the functions in our class satisfy a norm lower bound, say $\|c\|_D^2 \geq (1+\alpha)\epsilon^2$, then a simple calculation shows that learning with L_2 error ϵ implies weak learning with advantage $\alpha\epsilon^2/2$.

Our definition of weak learning also captures the standard boolean sense of weak learning, in which the learner is required to output a boolean hypothesis with 0/1 loss bounded away from $1/2$. Indeed, by an easy calculation, the 0/1 loss of a function $f : X \to \{\pm 1\}$ satisfies

$$\mathbb{P}_{(x,y)\sim D_c}[f(x) \neq y] = \frac{1}{2} - \frac{\langle c, f\rangle_D}{2}.$$

The difficulty of learning a concept class in the SQ model is captured by a parameter known as the statistical dimension of the class.

Definition 2.4 (Statistical dimension)**.** Let $\mathcal{C}$ be a concept class of either real-valued concepts or p-concepts (i.e. their corresponding conditional mean functions) on a domain X, and let D be a distribution on X. The (un-normalized) *correlation* of two concepts $c, c' \in \mathcal{C}$ under D is $|\langle c, c'\rangle_D|$.[3] The *average correlation* of $\mathcal{C}$ is defined to be

$$\rho_D(\mathcal{C}) = \frac{1}{|\mathcal{C}|^2} \sum_{c,c' \in \mathcal{C}} |\langle c, c'\rangle_D|.$$

The *statistical dimension on average* at threshold γ, $\mathrm{SDA}_D(\mathcal{C}, \gamma)$, is the largest d such that for all $\mathcal{C}' \subseteq \mathcal{C}$ with $|\mathcal{C}'| \geq |\mathcal{C}|/d$, $\rho_D(\mathcal{C}') \leq \gamma$.

Remark 2.5. For any general and large concept class $\mathcal{C}^*$ (such as all one-layer neural nets), we may consider a specific subclass $\mathcal{C} \subseteq \mathcal{C}^*$ and prove lower bounds on learning $\mathcal{C}$ in terms of the SDA of $\mathcal{C}$. These lower bounds extend to $\mathcal{C}^*$ because if it is hard to learn a subset, then it is hard to learn the whole class.

We will mainly be interested in the statistical dimension in a setting where bounds on pairwise correlations are known. In that case the following lemma holds.

Lemma 2.6 (adapted from (Feldman et al., 2017), Lemma 3.10)**.** *Suppose a concept class $\mathcal{C}$ has pairwise correlation γ, i.e. $|\langle c, c'\rangle_D| \leq \gamma$ for $c \neq c' \in \mathcal{C}$, and squared norm at most β, i.e. $\|c\|_D^2 \leq \beta$ for all $c \in \mathcal{C}$. Then for any $\gamma' > 0$,* $\mathrm{SDA}_D(\mathcal{C}, \gamma + \gamma') \geq |\mathcal{C}|\frac{\gamma'}{\beta - \gamma}$*. In particular, if $\mathcal{C}$ is a class of orthogonal concepts (i.e. $\gamma = 0$) with squared norm bounded by β, then* $\mathrm{SDA}(\mathcal{C}, \gamma') \geq |\mathcal{C}|\frac{\gamma'}{\beta}$.

Proof. Let $d = |\mathcal{C}|\frac{\gamma'}{\beta-\gamma}$, and observe that for any subset $\mathcal{C}' \subseteq \mathcal{C}$ satisfying $|\mathcal{C}'| \geq |\mathcal{C}|/d = \frac{\beta-\gamma}{\gamma'}$,

$$\begin{aligned}\rho_D(\mathcal{C}') &= \frac{1}{|\mathcal{C}|^2} \sum_{c,c' \in \mathcal{C}} |\langle c, c'\rangle_D| \\ &\leq \frac{1}{|\mathcal{C}'|^2}(|\mathcal{C}'|\beta + (|\mathcal{C}'|^2 - |\mathcal{C}'|)\gamma) \\ &= \gamma + \frac{\beta - \gamma}{|\mathcal{C}'|} \\ &= \gamma + \gamma'.\end{aligned}$$

□

3. Orthogonal Family of Neural Networks

We consider neural networks with one hidden layer with activation function $\phi : \mathbb{R} \to \mathbb{R}$, and with one output node that has some activation function $\psi : \mathbb{R} \to \mathbb{R}$. If we take the input dimension to be n and the number of hidden nodes to be m, then such a neural network is a function $f : \mathbb{R}^n \to \mathbb{R}$ given by

$$f(x) = \psi\left(\sum_{i=1}^m a_i \phi(w_i \cdot x)\right),$$

where $w_i \in \mathbb{R}^n$ are the weights feeding into the i^{th} hidden node, and $a_i \in \mathbb{R}$ are the weights feeding into the output node. If ψ takes values in $[-1, 1]$, we may also view f as defining a p-concept in terms of its conditional mean function.

For our construction, we need our functions to be orthogonal, and we need a lower bound on their norms. For the first property we only need the distribution on the domain to satisfy a relaxed kind of spherical symmetry that we term sign-symmetry, which says that the distribution must look identical on all orthants. To lower bound the norms, we need to assume that the distribution is Gaussian $\mathcal{N}(0, I)$.

Assumption 3.1 (Sign-symmetry)**.** *For any $z \in \{\pm 1\}^n$ and $x \in \mathbb{R}^n$, let $x \circ z$ denote $(x_1 z_1, \ldots, x_n z_n)$. A distribution D on $\mathbb{R}^n$ is* sign-symmetric *if for any $z \in \{\pm 1\}^n$ and x drawn from D, x and $x \circ z$ have the same distribution D.*

Assumption 3.2 (Odd outer activation)**.** *The outer activation ψ is an odd, increasing function, i.e. $\psi(-x) = -\psi(x)$.*

Note that ψ could be the identity function.

Assumption 3.3 (Inner activation)**.** *The inner activation $\phi \in L_2(\mathcal{N}(0, I))$.*

The construction of our orthogonal family of neural networks is simple and exploits sign-symmetry.

Definition 3.4 (Family of Orthogonal Neural Networks)**.** Let the domain be $\mathbb{R}^n$, let $\phi : \mathbb{R} \to \mathbb{R}$ be any well-behaved

[3] In the p-concept setting, it is instructive to note that in the notation of (Feldman et al., 2017), this correlation is precisely the distributional correlation $\chi_{D_0}(D_c, D_{c'})$ of the induced labeled distributions D_c and $D_{c'}$ under the reference distribution $D_0 = D \times \mathrm{Unif}\{\pm 1\}$.

activation function, and let $\psi : \mathbb{R} \to \mathbb{R}$ be any odd function. For an index set $S \subseteq [n]$, let $x_S \in \mathbb{R}^{|S|}$ denote the vector of x_i for $i \in S$. Fix any $k > 0$. For any sign-pattern $z \in \{\pm 1\}^k$, let $\chi(z)$ denote the parity $\prod_i z_i$. For any index set $S \subseteq_k [n]$, define a one-layer neural network with $m = 2^k$ hidden nodes,

$$\begin{aligned} g_S(x) &= \sum_{w \in \{-1,1\}^k} \chi(w)\phi\left(\frac{w \cdot x_S}{\sqrt{k}}\right) \\ f_S(x) &= \psi\left(g_S(x)\right). \end{aligned}$$

Our orthogonal family is

$$\mathcal{C}_{\mathrm{orth}}(n,k) = \{f_S \mid S \subseteq_k [n]\}.$$

Notice that the size of this family is $\binom{n}{k} = n^{\Theta(k)}$ (for appropriate k), which is $n^{\Theta(\log m)}$ in terms of m. We will take $k = \Theta(\log n)$, so that $m = \mathrm{poly}(n)$ and thus the neural networks are $\mathrm{poly}(n)$-sized, and the size of the family is $n^{\Theta(\log n)}$, i.e. quasipolynomial in n.

We now prove that our functions are orthogonal under any sign-symmetric distribution.

Theorem 3.5. *Let the domain be $\mathbb{R}^n$, and let D be a sign-symmetric distribution on $\mathbb{R}^n$. Fix any $k > 0$. Then $\langle f_S, f_T\rangle_D = 0$ for any two distinct $f_S, f_T \in \mathcal{C}_{\mathrm{orth}}(n,k)$.*

Proof. For the proof, the key property of our construction that we will use is the following: for any sign-pattern $z \in \{\pm 1\}^n$ and any $x \in \mathbb{R}^n$,

$$f_S(x \circ z) = \chi_S(z) f_S(x), \tag{1}$$

where $\chi_S(z) = \prod_{i \in S} z_i = \chi(z_S)$ is the parity on S of z. Indeed, observe first that

$$\begin{aligned} g_S(x \circ z) &= \sum_{w \in \{-1,1\}^k} \chi(w)\phi\left(\frac{w \cdot (x \circ z)_S}{\sqrt{k}}\right) \\ &= \sum_{w \in \{-1,1\}^k} \chi(w)\phi\left(\frac{(w \circ z_S) \cdot x_S}{\sqrt{k}}\right) \\ &= \sum_{w \in \{-1,1\}^k} \chi(w \circ z_S)\chi(z_S)\phi\left(\frac{(w \circ z_S) \cdot x_S}{\sqrt{k}}\right) \\ &= \chi(z_S) \sum_{w \in \{-1,1\}^k} \chi(w)\phi\left(\frac{w \cdot x_S}{\sqrt{k}}\right) \\ &\qquad\qquad \text{(replacing } w \circ z_S \text{ with } w) \\ &= \chi(z_S) g_S(x). \end{aligned}$$

The property then follows since ψ is odd and $\psi(av) = a\psi(v)$ for any $a \in \{\pm 1\}$ and $v \in \mathbb{R}$.

Consider f_S and f_T for any two distinct $S, T \subseteq_k [n]$. Recall that by the definition of sign-symmetry, for any $z \in \{\pm 1\}^n$ and x drawn from D, x and $x \circ z$ has the same distribution. Using this and Eq. (1), we have

$$\begin{aligned} \langle f_S, f_T\rangle_D &= \mathbb{E}_{x\sim D}[f_S(x) f_T(x)] \\ &= \mathbb{E}_{z\sim\{\pm1\}^n}\, \mathbb{E}_{x\sim D}[f_S(x\circ z) f_T(x \circ z)] \\ &\qquad\qquad \text{(sign-symmetry)} \\ &= \mathbb{E}_{z\sim\{\pm1\}^n}\, \mathbb{E}_{x\sim D}[\chi_S(z) f_S(x) \chi_T(z) f_T(x)] \\ &\qquad\qquad \text{(Eq. (1))} \\ &= \mathbb{E}_{x\sim D}\left[f_S(x) f_T(x)\, \mathbb{E}_{z\sim\{\pm1\}^n} \chi_S(z)\chi_T(z)\right] \\ &= 0, \end{aligned}$$

since $\mathbb{E}_{z\sim\{\pm1\}^n}\chi_S(z)\chi_T(z) = 0$ for any two distinct parities χ_S, χ_T. □

Remark 3.6. Our proof actually shows that any family of functions satisfying Eq. (1) is an orthogonal family under any sign-symmetric distribution.

We still need to establish that our functions are nonzero. For this we need to specialize to the Gaussian distribution, as well as consider specific activation functions (a similar analysis can in principle be carried out for other sign-symmetric distributions). For any n and k, it follows from Lemma A.1 that if the inner activation ϕ has a nonzero Hermite coefficient of degree k or higher, then the functions in $\mathcal{C}_{\mathrm{orth}}(n,k)$ are nonzero. The sigmoid, ReLU and sign functions all satisfy this property.

Corollary 3.7. *Let the domain be $\mathbb{R}^n$, and let D be any sign-symmetric distribution on $\mathbb{R}^n$. For any $\gamma > 0$,*

$$\mathrm{SDA}_D(\mathcal{C}_{\mathrm{orth}}(n,k), \gamma) \geq |\mathcal{C}_{\mathrm{orth}}(n,k)|\gamma = \binom{n}{k}\gamma.$$

Here we also assume that all $c \in \mathcal{C}_{\mathrm{orth}}(n,k)$ are nonzero for our distribution D.

Proof. Follows from Theorem 3.5 and Lemma 2.6, using a loose upper bound of 1 on the squared norm. □

We also need to prove norm lower bounds on our functions for our notions of learning to be meaningful. In Appendix A, we prove the following.

Theorem 3.8. *Let the inner activation function ϕ be* ReLU *or sigmoid, and let the outer activation function ψ be any odd, increasing, continuous function. Let the underlying distribution D be $\mathcal{N}(0, I_n)$. Then $\|f_S\| = \Omega(e^{-\Theta(k)})$, where the hidden constants depend on ψ and ϕ, for any $f_S \in \mathcal{C}_{\mathrm{orth}}(n,k)$.*

With this in hand, we now state our main SQ lower bounds.

Theorem 3.9. *Let the input dimension be n, and let the underlying distribution be $\mathcal{N}(0, I_n)$. Consider $\mathcal{C}_{\mathrm{orth}}(n,k)$ instantiated with $\phi =$ ReLU *or sigmoid and ψ any odd, increasing function (including the identity function), and* *let $m = 2^k$ be the hidden layer size of each neural net. Let A be an SQ learner using only inner product queries of tolerance τ. For any $k \in \mathbb{N}$, there exists $\tau = 1/n^{-\Theta(k)}$ such that A requires at least $n^{\Omega(k)}$ queries of tolerance τ to learn $\mathcal{C}_{\mathrm{orth}}(n,k)$ with advantage $1/\exp(k)$.*

In particular, there exist $k = \Theta(\log n)$ and $\tau = 1/n^{\Theta(\log n)}$ such that A requires at least $n^{\Omega(\log n)}$ queries of tolerance τ to learn $\mathcal{C}_{\mathrm{orth}}(n,k)$ with advantage $1/\operatorname{poly}(n)$. In this case $m = \operatorname{poly}(n)$, so that each function in the family has polynomial size. This is our main superpolynomial lower bound.

Proof. The proof amounts to careful choices of the parameters ϵ, γ and τ in Corollary 3.7 and Corollary 4.6. Recall that $\mathrm{SDA}(\mathcal{C}_{\mathrm{orth}}(n,k), \gamma) \geq n^{\Theta(k)}\gamma$. We pick $\gamma = n^{-\Theta(k)}$ appropriately such that $d = \mathrm{SDA}(\mathcal{C}_{\mathrm{orth}}(n,k), \gamma)$ is still $n^{\Theta(k)}$. Theorem 3.8 gives us a norm lower bound of $\exp(-\Theta(k))$, allowing us to take $\epsilon = \exp(-\Theta(k))$ and $\tau = \sqrt{\gamma} = n^{-\Theta(k)}$ in Corollary 4.6. □

4. SQ Lower Bounds

SQ Lower Bounds for Real-valued Functions Prior work (Szörényi, 2009; Feldman, 2012) has already established the following fundamental result, which we phrase in terms of our definition of statistical dimension. For the reader's convenience, we include a proof in Appendix B.

Theorem 4.1. *Let D be a distribution on X, and let $\mathcal{C}$ be a real-valued concept class over a domain X such that $\|c\|_D > \epsilon$ for all $c \in \mathcal{C}$. Consider any SQ learner that is allowed to make only inner product queries to an SQ oracle for the labeled distribution D_c for some unknown $c \in \mathcal{C}$. Let $d = \mathrm{SDA}_D(\mathcal{C}, \gamma)$. Then any such SQ learner needs at least $\Omega(d)$ queries of tolerance $\sqrt{\gamma}$ to learn $\mathcal{C}$ up to L_2 error ϵ.*

SQ Lower Bounds for *p*-concepts It turns out to be fruitful to view our learning problem in terms of a decision problem over distributions. We define the problem of distinguishing a valid labeled distribution from a randomly labeled one, and show a lower bound for this problem. We then show that learning is at least as hard as distinguishing, thereby extending the lower bound to learning as well. Our analysis closely follows that of (Feldman et al., 2017).

Definition 4.2 (Distinguishing between labeled and uniformly random distributions)**.** Let $\mathcal{C}$ be a class of p-concepts over a domain X, and let D be a distribution on X. Let $D_0 = D_{c_0}$ be the randomly labeled distribution $D \times \mathrm{Unif}\{\pm 1\}$. Suppose we are given SQ access either to a labeled distribution D_c for some $c \in \mathcal{C}$ such that $c \neq c_0$ or to D_0. The problem of distinguishing between labeled and uniformly random distributions is to decide which.

Remark 4.3. Given access to D_c for some truly boolean concept $c : X \to \{\pm 1\}$, it is easy to distinguish any other boolean function c' from c since $\|c - c'\|_D^2 = 2 - 2\langle c, c'\rangle_D$ (which is information-theoretically optimal as a distinguishing criterion) can be computed using a single inner product query. However, if c and c' are p-concepts, $\|c\|_D$ and $\|c'\|_D$ are not 1 in general and may be difficult to estimate. It is not obvious how best to distinguish the two, short of directly learning the target.

Considering the distinguishing problem is useful because if we can show that distinguishing itself is hard, then any reasonable notion of learning will be hard as well, including weak learning. We give simple reductions for both our notions of learning.

Lemma 4.4 (Learning is as hard as distinguishing)**.** *Let D be a distribution over the domain X, and let $\mathcal{C}$ be a p-concept class over X. Suppose there exists either*

(a) a weak SQ learner capable of learning $\mathcal{C}$ up to advantage ϵ using q queries of tolerance τ, where $\tau \leq \epsilon/2$; or,

(b) an SQ learner capable of learning $\mathcal{C}$ (assume $\|c\|_D \geq 3\epsilon$ for all $c \in \mathcal{C}$) up to L_2 error ϵ using q queries of tolerance τ, where $\tau \leq \epsilon^2$. Then there exists a distinguisher that is able to distinguish between an unknown D_c and D_0 using at most $q+1$ queries of tolerance τ.

Proof. (a) Run the weak learner to obtain $\tilde{c}$. If $c \neq c_0$, we know that $\langle \tilde{c}, c\rangle_D \geq \epsilon$, whereas if $c = c_0$, then $\langle \tilde{c}, c\rangle_D = 0$ no matter what $\tilde{c}$ is. A single additional query $(h(x,y) = \tilde{c}(x)y)$ of tolerance $\epsilon/2$ distinguishes between the two cases.

(b) Run the learner to obtain $\tilde{c}$. If $c \neq c_0$, i.e. $\|c\|_D \geq 3\epsilon$, we know that $\|\tilde{c} - c\|_D \leq \epsilon$, so that by the triangle inequality, $\|\tilde{c}\|_D \geq \|c\|_D - \|\tilde{c} - c\|_D \geq 2\epsilon$. But if $c = c_0$, then $\|\tilde{c}\|_D \leq \epsilon$. An additional query $(h(x,y) = \tilde{c}(x)^2)$ of tolerance ϵ^2 suffices to distinguish the two cases. □

We now prove the main lower bound on distinguishing.

Theorem 4.5. *Let D be a distribution over the domain X, and let $\mathcal{C}$ be a p-concept class over X. Then any SQ algorithm needs at least $d = \mathrm{SDA}(\mathcal{C}, \gamma)$ queries of tolerance $\sqrt{\gamma}$ to distinguish between D_c and D_0 for an unknown $c \in \mathcal{C}$. (We will consider deterministic SQ algorithms that always succeed, for simplicity.)*

Proof. Consider any successful SQ algorithm A. Consider the adversarial strategy where to every query $h : X \times \{-1,1\} \to [-1,1]$ of A (with tolerance $\tau = \sqrt{\gamma}$), we respond with $\mathbb{E}_{D_0}[h]$. We can pretend that this is a valid answer with respect to any $c \in \mathcal{C}$ such that $|\mathbb{E}_{D_c}[h] - \mathbb{E}_{D_0}[h]| \leq \tau$. Our argument will be based on showing that each such query rules out fairly few distributions, so that the number of queries required in total is large.

Since we assumed that A is a deterministic algorithm that always succeeds, it eventually correctly guesses that it is D_0 that it is getting answers from. Say it takes q queries to do so. For the k^{th} query h_k, let S_k be the set of concepts in $\mathcal{C}$ that are ruled out by our response $\mathbb{E}_{D_0}[h_k]$:

$$S_k = \{c \in \mathcal{C} \mid |\mathbb{E}_{D_c}[h] - \mathbb{E}_{D_0}[h]| \ > \tau\}.$$

We'll show that

(a) on the one hand, $\cup_{k=1}^{q} S_k = \mathcal{C}$, so that $\sum_{k=1}^{q} |S_k| \geq |\mathcal{C}|$,

(b) while on the other, $|S_k| \leq |\mathcal{C}|/d$ for every k. Together, this will mean that $q \geq d$.

For the first claim, suppose $\cup_{k=1}^{q} S_k$ were not all of $\mathcal{C}$, and indeed say $c \in \mathcal{C} \setminus (\cup_{k=1}^{q} S_k)$. This is a distribution that our answers were consistent with throughout, yet one that A's solution (D_0) is incorrect for. But A always succeeds, so for it not to have ruled out this D_c is impossible.

For the second claim, suppose for the sake of contradiction that for some k, $|S_k| \ > \ |\mathcal{C}|/d$. By Definition 2.4, this means we know that $\rho_D(S_k) \leq \gamma$. One of the key insights in the proof of (Szörényi, 2009) is that by expressing query expectations entirely in terms of inner products, we gain the ability to apply simple algebraic techniques. To this end, for any query function h, let $\widehat{h}(x) = (h(x,1) - h(x,-1))/2$. Observe that for any p-concept c,

$$\begin{aligned}
\langle \widehat{h}, c\rangle_D &= \mathbb{E}_{x\sim D}\left[h(x,1)\frac{c(x)}{2}\right] - \mathbb{E}_{x\sim D}\left[h(x,-1)\frac{c(x)}{2}\right] \\
&= \mathbb{E}_{x\sim D}\left[h(x,1)\frac{1+c(x)}{2}\right] \\
&\quad + \mathbb{E}_{x\sim D}\left[h(x,-1)\frac{1-c(x)}{2}\right] \\
&\quad - \mathbb{E}_{x\sim D}\left[h(x,1)\frac{1}{2}\right] - \mathbb{E}_{x\sim D}\left[h(x,-1)\frac{1}{2}\right] \\
&= \mathbb{E}_{D_c}[h] - \mathbb{E}_{D_0}[h],
\end{aligned}$$

the difference between the query expectations wrt D_c and D_0. Here we have expanded each $\mathbb{E}_{D_c}[h]$ using the fact that the label for x is 1 with probability $(1 + c(x))/2$ and -1 otherwise. Thus $|\langle \widehat{h_k}, c\rangle_D|$, where h_k is the k^{th} query, is greater than τ for any $c \in S_k$, since S_k are precisely those concepts ruled out by our response. We will show contradictory upper and lower bounds on the following quantity:

$$\Phi = \left\langle \widehat{h_k}, \sum_{c\in S_k} c \cdot \operatorname{sign}(\langle \widehat{h_k}, c\rangle_D) \right\rangle_D .$$

Note that since every query h satisfies $\|h(\cdot, y)\|_D \leq 1$ for all y, it follows by the triangle inequality that $\|\widehat{h}\|_D \leq 1$. So by Cauchy-Schwarz and our observation that $\rho_D(S_k) \leq \gamma$,

$$\begin{aligned}
\Phi^2 &\leq \|\widehat{h_k}\|_D^2 \cdot \left\| \sum_{c\in S_k} c \cdot \operatorname{sign}(\langle \widehat{h_k}, c\rangle) \right\|_D^2 \\
&\leq \sum_{c,c'\in S_k} |\langle c, c'\rangle_D| = |S_k|^2 \rho_D(S_k) \leq |S_k|^2 \gamma.
\end{aligned}$$

However since $|\langle \widehat{h_k}, c\rangle_D| \ > \tau$, we also have that $\Phi = \sum_{c\in S_k} |\langle \widehat{h_k}, c\rangle_D| \ > |S_k|\tau$. Since $\tau = \sqrt{\gamma}$, this contradicts our upper bound and in turn completes the proof of our second claim. And as noted earlier, the two claims together imply that $q \geq d$. □

The final lower bounds on learning thus obtained are stated as a corollary for convenience. The proof follows directly from Lemma 4.4 and Theorem 4.5.

Corollary 4.6. *Let D be a distribution over the domain X, and let $\mathcal{C}$ be a p-concept class over X. Let γ, τ be such that $\sqrt{\gamma} \leq \tau$. Let $d = \mathrm{SDA}(\mathcal{C}, \gamma)$.*

(a) Let ϵ be such that $\tau \leq \epsilon^2$, and assume $\|c\|_D \geq 3\epsilon$ for all $c \in \mathcal{C}$. Then any SQ learner learning $\mathcal{C}$ up to L_2 error ϵ requires at least $d - 1$ queries of tolerance τ.

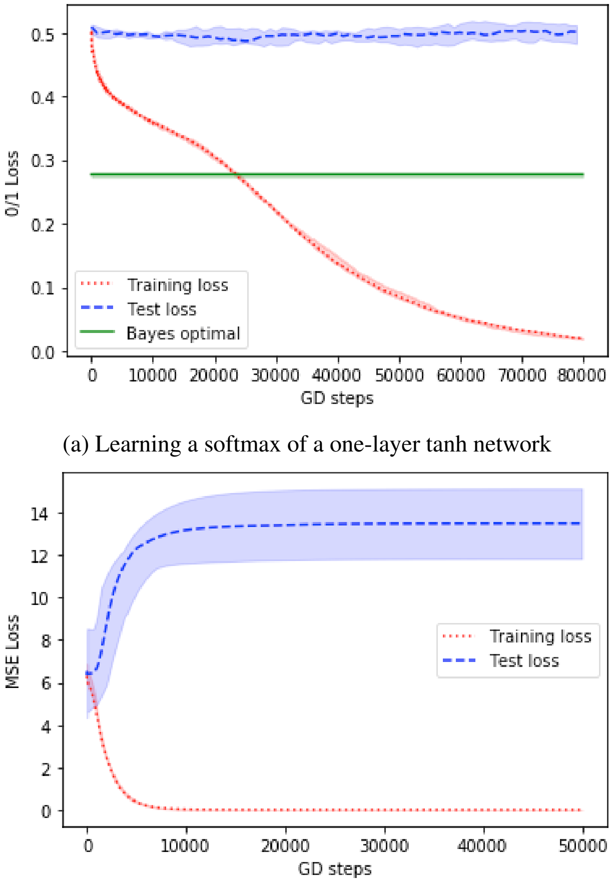
(b) Let ϵ be such that $\tau \leq \epsilon/2$. Then any weak SQ learner learning $\mathcal{C}$ up to advantage ϵ requires at least $d - 1$ queries of tolerance τ.

5. Experiments

We include experiments for both regression and classification. We train an overparameterized neural network on data from our function class, using gradient descent. We find that we are able to achieve close to zero training error, while test error remains high. This is consistent with our lower bound for these classes of functions.

For classification, we use a training set of size T of data corresponding to $f \in \mathcal{C}_{\text{orth}}(n,k)$ instantiated with $\phi = \tanh$ and $\psi = \tanh$. We draw $x \sim \mathcal{N}(0, I_n)$. For each x, y is picked randomly from $\{\pm 1\}$ in such a way that $\mathbb{E}[y|x] = f(x)$. Since the outer activation ψ is $\tanh$, this can be thought of as applying a softmax to the network's output, or as the Boolean label corresponding to a logit output. We train a sum of tanh network (i.e. a network in which the inner activation is $\tanh$ and no outer activation is applied) on this data using gradient descent on squared loss, threshold the output, and plot the resulting 0/1 loss. See Fig. 1(a). This setup models a common way in which neural networks are trained for classification problems in practice.

For regression, we use a training set of size T of data corresponding to $f \in \mathcal{C}_{\text{orth}}(n,k)$ instantiated with $\phi = \tanh$



(a) Learning a softmax of a one-layer tanh network

(b) Learning a linear combination of tanhs

Figure 1. In (a) the target function is a softmax (± 1 labels) of a sum of 2^7 tanh activations with $n = 14$; in (b) the labels are obtained similarly but without the softmax. In both cases, we train a 1-layer neural network with $5 \cdot 2^7 = 640$ tanh units (hence 10241 parameters) using a training set of size 6000 and a test set of size 1000, with the learning rate set to 0.01. For (a) we take the sign of this trained network and measure its training and testing 0/1 loss; for (b) we measure the train and test square-loss of the learned network directly. In (a) we also plot the test error of the bayes optimal network (sign of the target function).

and ψ being the identity. We draw $x \sim \mathcal{N}(0, I_n)$, and $y = f(x)$. We train a sum of tanh network on this data using gradient descent on squared loss, which we plot in Fig. 1(b). This setup models the natural way of using neural networks for regression problems.

In both cases, we train neural networks whose number of parameters considerably exceeds the amount of training data. In all our experiments, we plot the median over 10 trials and shade the inter-quartile range of the data.

Similar results hold with the inner activation ϕ being ReLU instead of $\tanh$, and are included in the supplementary material.

References

Abbe, E. and Sandon, C. Poly-time universality and limitations of deep learning. *arXiv preprint arXiv:2001.02992*, 2020.

Andoni, A., Panigrahy, R., Valiant, G., and Zhang, L. Learning sparse polynomial functions. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pp. 500–510. SIAM, 2014.

Andoni, A., Dudeja, R., Hsu, D., and Vodrahalli, K. Attribute-efficient learning of monomials over highly-correlated variables. In *Thirtieth International Conference on Algorithmic Learning Theory*, 2019.

Blum, A. and Rivest, R. L. Training a 3-node neural network is NP-complete. In *Advances in neural information processing systems*, pp. 494–501, 1989.

Blum, A., Furst, M., Jackson, J., Kearns, M., Mansour, Y., and Rudich, S. Weakly learning dnf and characterizing statistical query learning using fourier analysis. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pp. 253–262, 1994.

Brutzkus, A. and Globerson, A. Globally optimal gradient descent for a convnet with gaussian inputs. *CoRR*, abs/1702.07966, 2017.

Daniely, A. and Vardi, G. Hardness of learning neural networks with natural weights. *arXiv preprint arXiv:2006.03177*, 2020.

Diakonikolas, I., Kane, D., Kontonis, V., and Zarifis, N. Algorithms and SQ Lower Bounds for PAC Learning One-Hidden-Layer ReLU Networks. In *Conference on Learning Theory*, 2020. To appear.

Feldman, V. A complete characterization of statistical query learning with applications to evolvability. *Journal of Computer and System Sciences*, 78(5):1444–1459, 2012.

Feldman, V. A general characterization of the statistical query complexity. In *Conference on Learning Theory*, pp. 785–830, 2017.

Feldman, V., Grigorescu, E., Reyzin, L., Vempala, S. S., and Xiao, Y. Statistical algorithms and a lower bound for detecting planted cliques. *Journal of the ACM (JACM)*, 64(2):8, 2017.

Ge, R., Lee, J. D., and Ma, T. Learning one-hidden-layer neural networks with landscape design. In *ICLR*. OpenReview.net, 2018.

Goel, S., Kanade, V., Klivans, A. R., and Thaler, J. Reliably learning the relu in polynomial time. In *COLT*, pp. 1004–1042, 2017.

Goel, S., Karmalkar, S., and Klivans, A. Time/accuracy tradeoffs for learning a relu with respect to gaussian marginals. In *Advances in Neural Information Processing Systems*, pp. 8582–8591, 2019.

Jacot, A., Hongler, C., and Gabriel, F. Neural tangent kernel: Convergence and generalization in neural networks. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *NeurIPS*, pp. 8580–8589, 2018.

Janzamin, M., Sedghi, H., and Anandkumar, A. Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. *arXiv preprint arXiv:1506.08473*, 2015.

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

Kearns, M. J. and Schapire, R. E. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, 48(3):464–497, 1994.

Klivans, A. R. and Kothari, P. Embedding hard learning problems into gaussian space. In *APPROX-RANDOM*, volume 28 of *LIPIcs*, pp. 793–809. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014. ISBN 978-3-939897-74-3.

Klivans, A. R. and Sherstov, A. A. Cryptographic hardness for learning intersections of halfspaces. *Journal of Computer and System Sciences*, 75(1):2–12, 2009.

Livni, R., Shalev-Shwartz, S., and Shamir, O. On the computational efficiency of training neural networks. In *Advances in Neural Information Processing Systems*, pp. 855–863, 2014.

Shalev-Shwartz, S., Shamir, O., and Shammah, S. Failures of gradient-based deep learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3067–3075, 2017.

Shamir, O. Distribution-specific hardness of learning neural networks. *J. Mach. Learn. Res*, 19:32:1–32:29, 2018.

Song, L., Vempala, S., Wilmes, J., and Xie, B. On the complexity of learning neural networks. In *Advances in Neural Information Processing Systems*, pp. 5514–5522, 2017.

Szörényi, B. Characterizing statistical query learning: simplified notions and proofs. In *International Conference on Algorithmic Learning Theory*, pp. 186–200. Springer, 2009.

Vempala, S. and Wilmes, J. Gradient descent for one-hidden-layer neural networks: Polynomial convergence and sq lower bounds. In *Conference on Learning Theory*, pp. 3115–3117, 2019.

Vu, V. H. On the infeasibility of training neural networks with small mean-squared error. *IEEE Transactions on Information Theory*, 44(7):2892–2900, 1998.

Zhang, Q., Panigrahy, R., and Sachdeva, S. Electron-proton dynamics in deep learning. *CoRR*, abs/1702.00458, 2017.

Zhang, X., Yu, Y., Wang, L., and Gu, Q. Learning one-hidden-layer relu networks via gradient descent. In Chaudhuri, K. and Sugiyama, M. (eds.), *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pp. 1524–1534. PMLR, 2019.

Zhong, K., Song, Z., Jain, P., Bartlett, P. L., and Dhillon, I. S. Recovery guarantees for one-hidden-layer neural networks. In *ICML*, volume 70, pp. 4140–4149. JMLR.org, 2017.