Journal of Machine Learning Research 19 (2018) 1-29 Submitted 9/17; Revised 3/18; Published 8/18

Distribution-Specific Hardness of Learning Neural Networks

Ohad Shamir OHAD.SHAMIR@WEIZMANN.AC.IL
Weizmann Institute of Science, Rehovot, Israel

Editor: Amir Globerson

Abstract

Although neural networks are routinely and successfully trained in practice using simple gradient-based methods, most existing theoretical results are negative, showing that learning such networks is difficult, in a worst-case sense over all data distributions. In this paper, we take a more nuanced view, and consider whether specific assumptions on the "niceness" of the input distribution, or "niceness" of the target function (e.g. in terms of smoothness, non-degeneracy, incoherence, random choice of parameters etc.), are sufficient to guarantee learnability using gradient-based methods. We provide evidence that neither class of assumptions alone is sufficient: On the one hand, for any member of a class of "nice" target functions, there are difficult input distributions. On the other hand, we identify a family of simple target functions, which are difficult to learn even if the input distribution is "nice". To prove our results, we develop some tools which may be of independent interest, such as extending Fourier-based hardness techniques developed in the context of statistical queries (Blum et al., 1994), from the Boolean cube to Euclidean space and to more general classes of functions.

Keywords: neural networks, computational hardness, distributional assumptions, gradient-based methods

1. Introduction

Artificial neural networks have seen a dramatic resurgence in recent years, and have proven to be a highly effective machine learning method in computer vision, natural language processing, and other challenging AI problems. Moreover, successfully training such networks is routinely performed using simple and scalable gradient-based methods, in particular stochastic gradient descent.

Despite this success, our theoretical understanding of the computational tractability of such methods is quite limited, with most results being negative. For example, as discussed in Livni et al. (2014), learning even depth-2 networks in a formal PAC learning framework is computationally hard in the worst case, and even if the algorithm is allowed to return arbitrary predictors. As common in such worst-case results, these are proven using rather artificial constructions, quite different than the real-world problems on which neural networks are highly successful. In particular, since the PAC framework focuses on *distribution-free* learning (where the distribution generating the examples is unknown and rather arbitrary), the hardness results rely on carefully crafted distributions, which allows one to relate the learning problem to (say) an NP-hard problem or breaking a cryptographic system. However, what if we insist on "natural" distributions? Is it possible to show that neural networks learning becomes computationally tractable? Can we show that they can

be learned using the standard heuristics employed in practice, such as stochastic gradient descent?

To understand what a "natural" distribution refers to, we need to separate the distribution over examples (given as input-output pairs $(\mathbf{x}, y)$) into two components:

- *The input distribution* $p(\mathbf{x})$: "Natural" input distributions on Euclidean space tend to have properties such as smoothness, non-degeneracy, incoherence etc.
- *The target function* $h(\mathbf{x})$: In PAC learning, it is assumed that the output y equals $h(\mathbf{x})$, where h is some unkown target function from the hypothesis class we are considering. In studying neural networks, it is common to consider the class of all networks which share some fixed architecture (e.g. feedforward networks of a given depth and width). However, one may argue that the parameters of real-world networks (e.g. the weights of each neuron) are not arbitrary, but exhibit various features such as non-degeneracy or some "random like" appearance. Indeed, networks with a random structure have been shown to be more amenable to analysis in various situations (see for instance Daniely et al., 2016; Arora et al., 2014; Choromanska et al., 2015, and references therein).

Empirical evidence seems to suggest that many pairs of input distributions and target functions are computationally tractable to learn, using standard methods. However, how do we characterize these pairs? Would appropriate assumptions on one of them be sufficient to show learnability?

In this paper, we investigate these two components, and provide evidence that *neither one* of them alone is generally enough to guarantee computationally tractable learning, at least with methods resembling those used in practice. Specifically, we focus on simple, shallow ReLU networks, assume that the data can be perfectly predicted by some such network, and even allow *over-parametrization* (a.k.a. over-specification or improper learning), in the sense that we allow the learning algorithm to output a predictor which is possibly larger and more complex than the target function (this technique increases the power of the learner, and was shown to make the learning problem easier in theory and in practice, e.g. Livni et al. 2014; Safran and Shamir 2016; Soudry and Carmon 2016). Even under such favorable conditions, we show the following:

- **Hardness for "natural" target functions.** For each individual target function coming from a simple class of small, shallow ReLU networks (even if its parameters are chosen randomly or in some other oblivious way), we show that no algorithm invariant to linear transformations can successfully learn it w.r.t. all input distributions in polynomial time (this corresponds, for instance, to standard gradient-based methods together with data whitening or preconditioning). This result is based on a reduction from learning intersections of halfspaces. Although that problem is known to be hard in the worst-case over both input distributions and target functions, we essentially show that invariant algorithms as above do not "distinguish" between worst-case and average-case: If one can learn a particular target function with such an algorithm, then the algorithm can learn nearly all target functions in that class.
- **Hardness for "natural" input distributions.** We show that target functions of the form $\mathbf{x} \mapsto \psi(\mathbf{w}^\top \mathbf{x})$ for any periodic ψ are generally difficult to learn using

gradient-based methods, even if the input distribution is fixed and belongs to a very broad class of smooth input distributions (including, for instance, Gaussians and mixtures of Gaussians). Note that such functions can essentially be constructed by simple shallow networks, and can be seen as an extension of generalized linear models (see McCullagh and Nelder 1989 for a survey). Unlike the previous result, which relies on a computational hardness assumption, the results here are geometric in nature, and imply that the gradient of the objective function, nearly everywhere, contains virtually no signal on the underlying target function. Therefore, any algorithm which relies on gradient information cannot learn such functions. Interestingly, the difficulty here is *not* in having a plethora of spurious local minima or saddle points—the associated stochastic optimization problem may actually have no such critical points. Instead, the objective function may exhibit properties such as *flatness* nearly everywhere, unless one is already very close to the global optimum. This highlights a potential pitfall in non-convex learning, which occurs already for a slight extension of generalized linear models, and even for "nice" input distributions.

Together, these results indicate that in order to explain the practical success of neural network learning with gradient-based methods, one would need to employ a careful combination of assumptions on both the input distribution and the target function, and that results with even a "partially" distribution-free flavor (which are common, for instance, in convex learning problems) may be difficult to attain here.

To prove our results, we develop some tools which may be of independent interest. In particular, the techniques used to prove hardness of learning functions of the form $\mathbf{x} \mapsto \psi(\mathbf{w}^\top \mathbf{x})$ are based on Fourier analysis, and have some close connections to hardness results on learning Boolean functions such as parities in the well-known framework of learning from statistical queries (Kearns, 1998): In both cases, one essentially shows that the Fourier transform of the target function has very small support, and hence does not "correlate" with most functions, making it difficult to learn using certain methods. However, we consider a more general and arguably more natural class of input distributions over Euclidean space, rather than distributions on the Boolean cube. In a sense, we show that learning general periodic functions over Euclidean space is difficult (at least with gradient-based methods), for the same reasons that learning parities over the Boolean cube is difficult in the statistical queries framework. This connection has recently been formalized and extended in Song et al. (2017) (see discussion below).

1.1 Related Work

Recent years have seen quite a few papers on the theory of neural network learning. Below, we only briefly mention those most relevant to our paper.

In a very elegant work, Janzamin et al. (2015) have shown that a certain method based on tensor decompositions allows one to provably learn simple neural networks by a combination of assumptions on the input distribution and the target function. However, a drawback of their method is that it requires rather precise knowledge of the input distribution and its derivatives, which is rarely available in practice. In contrast, our focus is on algorithms which do not utilize such knowledge. Other works which show computationally-efficient

learnability of certain neural networks under sufficiently strong distributional assumptions include Arora et al. (2014); Livni et al. (2014); Andoni et al. (2014); Zhang et al. (2015).

In the context of learning functions over the Boolean cube, it is known that even if we restrict ourself to a particular input distribution (as long as it satisfies some mild conditions), it is difficult to learn parity functions using statistical query algorithms (Kearns, 1998; Blum et al., 1994), which also include gradient-based methods (Feldman et al., 2015). Since parities can be implemented with small real-valued networks, this implies that for "most" input distributions on the Boolean cube, there are neural networks which are unlikely to be learnable with gradient-based methods. However, data provided to neural networks in practice are not in the form of Boolean vectors, but rather vectors of floating-point numbers. Moreover, some assumptions on the input distribution, such as smoothness and Gaussianity, only make sense once we consider the support to be Euclidean space rather than the Boolean cube. Perhaps these are enough to guarantee computational tractability? A contribution of this paper is to show that this is not the case, and to formally demonstrate how phenomena similar to the Boolean case also occurs in Euclidean space, using appropriate target functions and distributions.

Related to the above, Song et al. (2017) recently showed that statistical query algorithms indeed cannot learn certain neural networks, using target functions similar to those we consider in Sec. 4, and for log-concave input distributions[1]. In contrast, our result for such target functions is specific to gradient-based methods, but applies to a different large family of distributions, not necessarily log-concave. Furthermore, we note that the challenges in fitting ridge functions $\mathbf{x} \mapsto \psi(\mathbf{w}^\top \mathbf{x})$ for certain ψ (when $\mathbf{x}$ is standard Gaussian and one attempts to fit the target function using a function of the same form) was also studied in Donoho and Johnstone (1989).

Finally, we note that Klivans and Kothari (2014) provides improper-learning hardness results, which hold even for a standard Gaussian distribution on Euclidean space, and for any algorithm. However, unlike our paper, their focus is on hardness of agnostic learning (where the target function is arbitrary and does not have to correspond to a given class), the results are specific to the standard Gaussian distribution, and the proofs are based on a reduction from the Boolean case.

The paper is structured as follows: In Sec. 2, we formally present some notation and concepts used throughout the paper. In Sec. 3, we provide our hardness results for natural target functions, and in Sec. 4, we provide our hardness results for natural input distributions. All proofs are presented in Sec. 5.

2. Preliminaries

We generally let bold-faced letters denote vectors. Given a complex-valued number $z = a + ib$, we let $\overline{z} = a - ib$ denote its complex conjugate, and $|z| = \sqrt{a^2 + b^2}$ denote its modulus. Given a function f, we let ∇f denote its gradient and $\nabla^2 f$ denote its Hessian (assuming they exist).

Neural Networks. Our results focus on learning predictors which can be described by simple and shallow (depth 2 or 3) neural networks. A standard feedforward neural network

1. The arXiv technical report on which our paper is based was published in September 2016, whereas their arXiv technical report was published in July 2017.

is composed of neurons, each of which computes the mapping $\mathbf{x} \mapsto \sigma(\mathbf{w}^\top \mathbf{x} + b)$, where $\mathbf{w}, b$ are parameters and σ is a scalar activation function, for example the popular ReLU function $[z]_+ = \max\{0, z\}$. These neurons are arranged in parallel in layers, so the output of each layer can be compactly represented as $\mathbf{x} \mapsto \sigma(W^\top \mathbf{x} + \mathbf{b})$, where W is a matrix (each column corresponding to the parameter vector of one of the neurons), $\mathbf{b}$ is a vector, and σ applies an activation function on the coordinates of $W^\top \mathbf{x}$. In vanilla feedforward networks, such layers are connected to each other, so given an input $\mathbf{x}$, the output equals

$$\sigma_k(W_k^\top \sigma_{k-1}(W_{k-1}^\top \dots \sigma_2(W_2^\top \sigma_1(W_1^\top \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) \dots + \mathbf{b}_{k-1}) + \mathbf{b}_k),$$

where W_i, b_i, σ_i are parameter of the i-th layer. The number of layers k is denoted as the depth of the network, and the maximal number of columns in W_i is denoted as the width of the network. For simplicity, in this paper we focus on networks which output a real-valued number, and measure our performance with respect to the squared loss (that is, given an input-output example $(\mathbf{x}, y)$, where $\mathbf{x}$ is a vector and $y \in \mathbb{R}$, the loss of a predictor p on the example is $(p(\mathbf{x}) - y)^2$).

Gradient-Based Methods. Gradient-based methods are a class of optimization algorithms for solving problems of the form $\min_{\mathbf{w} \in \mathcal{W}} F(\mathbf{w})$ (for some given function F and assuming $\mathbf{w}$ is a vector in Euclidean space), based on computing $\nabla F(\mathbf{w})$ of approximations of $\nabla F(\mathbf{w})$ at various points $\mathbf{w}$. Perhaps the simplest such algorithm is gradient descent, which initializes deterministically or randomly at some point $\mathbf{w}_1$, and iteratively performs updates of the form $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla F(\mathbf{w}_t)$, where $\eta_t > 0$ is a step size parameter. In the context of statistical supervised learning problems, we are usually interested in solving problems of the form $\min_{\mathbf{w} \in \mathcal{W}} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}}[\ell(f(\mathbf{w}, \mathbf{x}), h(\mathbf{x}))]$, where $\{\mathbf{x} \mapsto f(\mathbf{w}, \mathbf{x}) : \mathbf{w} \in \mathcal{W}\}$ is some class of predictors, h is a target function, and ℓ is some loss function. Since the distribution $\mathcal{D}$ is generally unknown, one cannot compute the gradient of this function w.r.t. $\mathbf{w}$ directly, but can still compute approximations, e.g. by sampling one $\mathbf{x}$ at random and computing the gradient (or sub-gradient) of $\ell(f(\mathbf{w}, \mathbf{x}), h(\mathbf{x}))$. The same approach can be used to solve empirical approximations of the above, i.e. $\min_{\mathbf{w} \in \mathcal{W}} \frac{1}{m} \sum_{i=1}^m \ell(f(\mathbf{w}, \mathbf{x}_i), h(\mathbf{x}_i))$ for some data set $\{(\mathbf{x}_i, h(\mathbf{x}_i))\}_{i=1}^m$. These are generally known as stochastic gradient methods, and are one of the most popular and scalable machine learning methods in practice.

PAC Learning. For the results of Sec. 3, we will rely on the following standard definition of PAC learning with respect to Boolean functions: Given a hypothesis class $\mathcal{H}$ of functions from $\{0,1\}^d$ to $\{0,1\}$, we say that a learning algorithm PAC-learns $\mathcal{H}$ if for any $\epsilon \in (0,1)$, any distribution $\mathcal{D}$ over $\{0,1\}^d$, and any $h^\star \in \mathcal{H}$, if the algorithm is given oracle access to i.i.d. samples $(\mathbf{x}, h^\star(\mathbf{x}))$ where $\mathbf{x}$ is sampled according to $\mathcal{D}$, then in time $\mathrm{poly}(d, 1/\epsilon)$, the algorithm returns a function $f : \{0,1\}^d \mapsto \{0,1\}$ (which can be evaluated in $\mathrm{poly}(d)$ time) such that $\Pr_{\mathbf{x} \sim \mathcal{D}}(f(\mathbf{x}) \neq h^\star(\mathbf{x})) \leq \epsilon$ with high probability (for our purposes, it will be enough to consider any constant close to 1). Note that in the definition above, we allow f not to belong to the hypothesis class $\mathcal{H}$. This is often denoted as "improper" learning, and allows the learning algorithm more power than in "proper" learning, where f must be a member of $\mathcal{H}$.

3. Natural Target Functions

In this section, we consider simple target functions parameterized by vectors $\mathbf{w}_1, \ldots, \mathbf{w}_n$, of the form

$$\mathbf{x} \;\mapsto\; \left[\sum_{i=1}^{n}[\langle \mathbf{w}_i, \mathbf{x}\rangle]_+\right]_+ - \left[\sum_{i=1}^{n}[\langle \mathbf{w}_i, \mathbf{x}\rangle]_+ - 1\right]_+ ,$$

where $[z]_+ = \max\{0, z\}$ is the ReLU function, which correspond to standard depth-3 ReLU networks with $2n$ neurons in the first layer and 2 neurons in the second layer. Equivalently, these functions can also be written as

$$\mathbf{x} \mapsto \left[\sum_{i=1}^{n}[\langle \mathbf{w}_i, \mathbf{x}\rangle]_+\right]_{[0,1]} ,$$

where $[z]_{[0,1]} = \min\{1, \max\{0, z\}\}$ is the clipping operation on the interval $[0, 1]$. For the rest of this section, we will use the latter formulation for convenience. Letting $W = [\mathbf{w}_1, \ldots, \mathbf{w}_n]$, we can write such predictors as $\mathbf{x} \mapsto h(W^\top \mathbf{x})$ for an appropriate fixed function h. Our goal would be to show that *individually* for any such target function (regardless of how W is chosen, as long as it has full column rank), and any polynomial-time learning algorithm satisfying some conditions, there exists an input distribution on which it must fail.

We begin by noting that some algorithmic assumption is *necessary* to get such a target-function-specific result. Indeed, if we fix the target function in advance, we can always "learn" by the following algorithm: return the target function, regardless of the training data. To avoid such trivial scenarios, we will consider algorithms which exhibit certain natural invariances to the coordinate system used. One very natural invariance is with respect to orthogonal transformations: For example, if we rotate the input instances $\mathbf{x}_i$ in a fixed manner, then an orthogonally-invariant algorithm will return a predictor which still makes the same predictions on those instances. Formally, this invariance is defined as follows:

Definition 1 *Let $\mathcal{A}$ be an algorithm which inputs a data set $(\{\mathbf{x}_i, y_i\})_{i=1}^m$ (where $\mathbf{x}_i \in \mathbb{R}^d$) and outputs a predictor $\mathbf{x} \mapsto f(W^\top \mathbf{x})$ (for some function f and matrix W dependent on the data set). We say that $\mathcal{A}$ is* orthogonally-invariant*, if for any orthogonal matrix $M \in \mathbb{R}^{d\times d}$, if we feed the algorithm with $\{M\mathbf{x}_i, y_i\}_{i=1}^m$, the algorithm returns a predictor $\mathbf{x} \mapsto f(W_M^\top \mathbf{x})$, where f is the same as before and W_M is such that $W_M^\top M\mathbf{x}_i = W^\top \mathbf{x}_i$ for all $\mathbf{x}_i$.*

Remark 2 *The definition as stated refers to deterministic algorithms. For stochastic algorithms, we will understand orthogonal invariance to mean orthogonal invariance conditioned on any realization of the algorithm's random coin flips.*

For example, standard gradient and stochastic gradient descent methods for optimizing W (possibly with coordinate-oblivious regularization, such as L_2 regularization) can be easily shown to be orthogonally-invariant[2]. However, for our results we will need to make

2. Essentially, this is because the gradient of any function $g(W^\top \mathbf{x}) = g(\langle \mathbf{w}_1, \mathbf{x}\rangle, \ldots, \langle \mathbf{w}_k, \mathbf{x}\rangle)$ w.r.t. any $\mathbf{w}_i$ is proportional to $\mathbf{x}$. Thus, if we multiply $\mathbf{x}$ by an orthogonal M, the gradient also gets multiplied by M. Since $M^\top M = I$, the inner products of instances $\mathbf{x}$ and gradients remain the same. Therefore, by induction, it can be shown that any algorithm which operates by incrementally updating some iterate by linear combinations of gradients will be rotationally invariant.

a somewhat stronger invariance assumption, namely invariance to general invertible linear transformations of the data (not necessarily just orthogonal). This is formally defined as follows:

Definition 3 *An algorithm $\mathcal{A}$ is* linearly-invariant*, if it satisfies Definition 1 for any invertible matrix $M \in \mathbb{R}^{d\times d}$ (rather than just orthogonal ones).*

One well-known example of such an algorithm (which is also invariant to affine transformations) is the Newton method (Boyd and Vandenberghe, 2004). More relevant to our purposes, linear invariance occurs whenever an orthogonally-invariant algorithm preconditions or "whitens" the data so that its covariance has a fixed structure (e.g. the identity matrix, possibly after a dimensionality reduction if the data is rank-deficient). For example, even though gradient descent methods are not linearly invariant, they become so if we precede them by such a preconditioning step. This is formalized in the following theorem:

Theorem 4 *Let $\mathcal{A}$ be any algorithm which given $\{\mathbf{x}_i, y_i\}_{i=1}^m$, computes the whitening matrix $P = D^{-1}U^\top$ (where $X = [\mathbf{x}_1\ \mathbf{x}_2\ \ldots\ \mathbf{x}_m]$, $X = UDV^\top$ is a thin[3] SVD decomposition of X), feeds $\{P\mathbf{x}_i, y_i\}_{i=1}^m$ to an orthogonally-invariant algorithm, and given the output predictor $\mathbf{x} \mapsto f(W^\top\mathbf{x})$, returns the predictor $\mathbf{x} \mapsto f((P^\top W)^\top\mathbf{x})$. Then $\mathcal{A}$ is linearly-invariant.*

It is easily verified that the covariance matrix of the transformed instances $P\mathbf{x}_1, \ldots, P\mathbf{x}_m$ is the $r \times r$ identity matrix (where $r = \mathrm{Rank}(X)$), so this is indeed a whitening transform. We note that whitening is a very common preprocessing heuristic, and even when not done explicitly, scalable approximate whitening and preconditioning methods are very common and widely recognized as useful for training neural networks (for example, Adagrad and batch normalization, see Duchi et al. 2011 and Ioffe and Szegedy 2015).

To show our result, we rely on a reduction from a PAC-learning problem known to be computationally hard, namely learning intersections of halfspaces. These are Boolean predictors parameterized by $\mathbf{w}_1, \ldots, \mathbf{w}_n \in \mathbb{R}^d$ and $b_1, \ldots, b_n \in \mathbb{R}$, which compute a mapping of the form

$$\mathbf{x} \ \rightarrow \ \bigwedge_{i=1}^{n} \left(\langle \mathbf{w}_i, \mathbf{x}\rangle \geq b_i\right)$$

(where we let 1 correspond to 'true' and 0 to 'false'). The problem of PAC-learning intersections of halfspaces over the Boolean cube ($\mathbf{x} \in \{0,1\}^d$) has been well-studied. In particular, two known hardness results are the following:

- Klivans and Sherstov (2009) show that under a certain well-studied cryptographic assumption (hardness of finding unique shortest vectors in a high-dimensional lattice), no algorithm can PAC-learn intersection of $n_d = d^\delta$ halfspaces (where δ is any positive constant), even if the coordinates of $\mathbf{w}_i$ and b_i are all integers, and $\max_i \|(\mathbf{w}_i, b_i)\| \leq \mathrm{poly}(d)$.

3. That is, if X is of size $d \times m$, then U is of size $d \times \mathrm{Rank}(X)$, D is of size $\mathrm{Rank}(X) \times \mathrm{Rank}(X)$, and V is of size $m \times \mathrm{Rank}(X)$.

- Daniely and Shalev-Shwartz (2016) show that under an assumption related to the hardness of refuting random K-SAT formulas, no algorithm can PAC-learn intersections of $n_d = \omega(\log(d))$ halfspaces (as $d \to \infty$), even if the coordinates of $\mathbf{w}_i$ and b_i are all integers, and $\max_i \|(\mathbf{w}_i, b_i)\| = \mathcal{O}(d)$.

In the theorem below, we will use the result of Daniely and Shalev-Shwartz (2016), which applies to an intersection of a smaller number of halfspaces, and with smaller norms. However, similar results can be shown using Klivans and Sherstov (2009), at the cost of worse polynomial dependencies on d.

The main result of this section is the following:

Theorem 5 *Consider any network $h(W_\star^\top \mathbf{x}) = [\sum_{i=1}^{n_d} [\langle \mathbf{w}_i^\star, \mathbf{x}\rangle]_+]_{[0,1]}$ (where the columns of $W_\star$ are $\mathbf{w}_1^\star \ldots \mathbf{w}_n^\star$), which satisfies the following:*

- *$n_d = \omega(\log(d))$ as $d \to \infty$*
- *$\max_i \|\mathbf{w}_i^\star\| = \mathcal{O}(d)$*
- *$\mathbf{w}_1^\star \ldots \mathbf{w}_n^\star$ are linearly independent, so the smallest singular value $s_{\min}(W_\star)$ of $W_\star$ is strictly positive.*

Then under the assumption stated in Daniely and Shalev-Shwartz (2016), there is no linearly-invariant algorithm which for any $\epsilon > 0$ and any distribution $\mathcal{D}$ over vectors of norm $\frac{\mathcal{O}(d\sqrt{dn_d})}{\min\{1, s_{\min}(W_\star)\}}$, given only access to samples $(\mathbf{x}, h(W_\star^\top \mathbf{x}))$ where $\mathbf{x} \sim \mathcal{D}$, runs in time poly$(d, 1/\epsilon)$ and returns with high probability a predictor $\mathbf{x} \mapsto f(W^\top \mathbf{x})$ such that

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}} \left[\left(f(W^\top \mathbf{x}) - h(W_\star^\top \mathbf{x}) \right)^2 \right] \leq \epsilon.$$

Note that the result holds even if the returned predictor $f(W^\top \mathbf{x})$ has a different structure than $h(W_\star^\top \mathbf{x})$, and W is of a larger size than $W_\star$. Thus, it applies even if the algorithm implements over-parametrization and attempts to train a network larger than $h(W_\star^\top \mathbf{x})$.

The proof (which is provided in Sec. 5) can be sketched as follows: First, the hardness assumption for learning intersection of halfspaces is shown to imply hardness of learning networks $\mathbf{x} \mapsto h(W^\top \mathbf{x})$ as described above (and even if W has linearly independent columns—a restriction which will be important later). However, this only implies that no algorithm can learn $\mathbf{x} \mapsto h(W^\top \mathbf{x})$ for *all* W and all input distributions $\mathcal{D}$. In contrast, we want to show that learning would be difficult even for *some fixed* $W_\star$. To do so, we show that if an algorithm is linearly invariant, then the ability to learn with respect to some W and all distributions $\mathcal{D}$ means that we can learn with respect to all W and all $\mathcal{D}$. Roughly speaking, we argue that for linearly-invariant algorithms, "average-case" and "worst-case" hardness are the same here. Intuitively, this is because given some arbitrary $W, \mathcal{D}$, we can create a different input distribution $\tilde{\mathcal{D}}$, so that $W, \tilde{\mathcal{D}}$ "look like" $W_\star, \mathcal{D}$ under some linear transformation (see Figure 1 for an illustration). Therefore, a linearly-invariant algorithm which succeeds on one will also succeed on the other.

A bit more formally, let us fix some $W_\star$ (with linearly independent columns), and suppose we have a linearly-invariant algorithm which can successfully learn $\mathbf{x} \mapsto h(W_\star^\top \mathbf{x})$ with

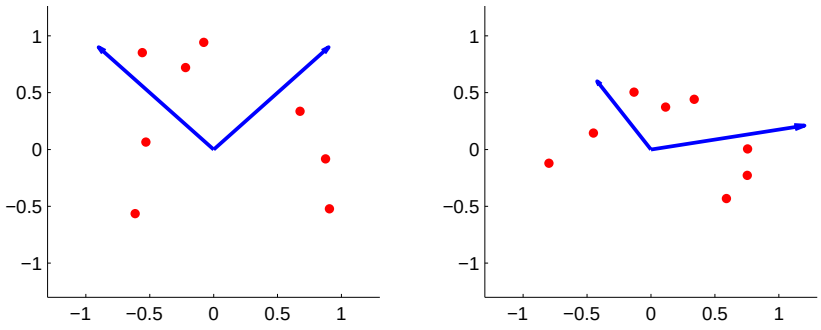

Figure 1: Correspondence between $W_\star, \mathcal{D}$ (left figure) and $W, \tilde{\mathcal{D}}$ (right figure). Arrows correspond to columns of $W_\star$ and W, and dots correspond to the support of $\mathcal{D}$ and $\tilde{\mathcal{D}}$. $\tilde{\mathcal{D}}$ is constructed so that the same linear transformation mapping $W_\star$ to W also maps $\mathcal{D}$ to $\tilde{\mathcal{D}}$.

respect to any input distribution. Let $W, \mathcal{D}$ be some other matrix and distribution with respect to which we wish to learn (where W has full column rank and is of the same size as $W_\star$). Then it can be shown that there is an invertible matrix M such that $W = M^\top W_\star$. Since the algorithm successfully learns $\mathbf{x} \mapsto h(W_\star^\top \mathbf{x})$ with respect to any input distribution, it would also successfully learn if we use the input distribution $\tilde{\mathcal{D}}$ defined by sampling $\mathbf{x} \sim \mathcal{D}$ and returning $M\mathbf{x}$. This means that the algorithm would succesfully learn from data distributed as

$$(\mathbf{x}, h(W_\star^\top \mathbf{x}))\ ,\ \mathbf{x} \sim \tilde{\mathcal{D}} \iff (M\mathbf{x}, h(W_\star^\top (M\mathbf{x})))\ ,\ \mathbf{x} \sim \mathcal{D} \iff (M\mathbf{x}, h(W^\top \mathbf{x}))\ ,\ \mathbf{x} \sim \mathcal{D}.$$

Since the algorithm is linearly-invariant, it can be shown that this implies successful learning from $(\mathbf{x}, h(W^\top \mathbf{x}))$ where $\mathbf{x} \sim \mathcal{D}$, as required.

In the sketch above, we have ignored some technical issues. For example, we need to be careful that M has a bounded spectral norm, so that it induces a linear transformation which does not distort norms by too much (as all our arguments apply for input distributions supported on a bounded domain). A second issue is that if we apply a linearly-invariant algorithm on a data set transformed by M, then the invariance is only with respect to the data, not necessarily with respect to new instances $\mathbf{x}$ sampled from the same distribution (and this restriction is necessary for results such as Thm. 4 to hold without further assumptions). However, it can be shown that if the data set is large enough, invariance will still occur with high probability over the sampling of $\mathbf{x}$, which is sufficient for our purposes.

4. Natural Input Distributions

In this section, we consider the difficulty of gradient-based methods to learn certain target functions, even with respect to smooth, well-behaved distributions over $\mathbb{R}^d$. Specifically, we will consider functions of the form $\mathbf{x} \mapsto \psi(\mathbf{w}^{\star\top} \mathbf{x})$, where $\mathbf{w}^\star$ is a vector of bounded norm and ψ is a periodic function. Note that if ψ is continuous and piecewise linear, then $\psi(\mathbf{w}^{\star\top} \mathbf{x})$ can be implemented by a depth-2 neural ReLU network on any bounded subset

of the domain. More generally, any continuous periodic function can be approximated arbitrarily well by such networks.

Our formal results rely on Fourier analysis and are a bit technical. Hence, we precede them with an informal description, outlining the main ideas and techniques, and presenting a specific case study which may be of independent interest (Subsection 4.1). The formal results are presented in Subsection 4.2.

4.1 Informal Description of Results and Techniques

Consider a target function of the form $\mathbf{x} \mapsto \psi(\mathbf{w}^{\star\top}\mathbf{x})$, and an input distribution with density function $\varphi^2(\cdot)$, where φ is some non-negative function (we consider the density as the square of some function in order to simplify notation later on). Suppose we attempt to learn this target function (with respect to the squared loss) using *some* hypothesis class, which can be parameterized by a bounded-norm vector $\mathbf{v}$ in some subset $\mathcal{V}$ of an Euclidean space (not necessarily of the same dimensionality as $\mathbf{w}^\star$), so each predictor in the class can be written as $\mathbf{x} \mapsto f(\mathbf{v}, \mathbf{x})$ for some fixed mapping f. Thus, our goal is essentially to solve the stochastic optimization problem

$$\min_{\mathbf{v}:\mathbf{v}\in\mathcal{V}} \ \mathbb{E}_{\mathbf{x}\sim\varphi^2}\left[\left(f(\mathbf{v},\mathbf{x}) - \psi(\mathbf{w}^{\star\top}\mathbf{x})\right)^2\right]. \tag{1}$$

In this section, we study the geometry of this objective function, and show that under mild conditions on f, and assuming the norm of $\mathbf{w}^\star$ is reasonably large, the gradient of the objective function with respect to $\mathbf{w}$ is almost independent of $\mathbf{w}^\star$, in the following sense: If we fix any $\mathbf{w}$ and choose $\mathbf{w}^\star$ uniformly at random, the gradient will be extremely concentrated around a fixed value which is independent of $\mathbf{w}^\star$ (e.g. exponentially small in $\|\mathbf{w}^\star\|^2$ for a Gaussian or a mixture of Gaussians). Therefore, assuming $\|\mathbf{w}^\star\|$ is reasonably large, any standard gradient-based method will follow a trajectory nearly independent of $\mathbf{w}^\star$. In fact, in practice we do not even have access to exact gradients of Eq. (1), but only to noisy and biased versions of it (e.g. if we perform stochastic gradient descent, and certainly if we use finite-precision computations). In that case, the noise will completely obliterate the exponentially small signal about $\mathbf{w}^\star$ in the gradients, and will make the trajectory essentially independent of $\mathbf{w}^\star$. As a result, assuming ψ and the distribution is such that the function $\psi(\mathbf{w}^{\star\top}\mathbf{x})$ is sensitive to the direction of $\mathbf{w}^\star$, it follows that these methods will fail to optimize Eq. (1) successfully. Finally, we note that in practice, it is common to solve not Eq. (1) directly, but rather its empirical approximation with respect to some fixed finite training set. Still, by concentration of measure, this empirical objective would converge to the one in Eq. (1) given enough data, so the same issues will occur.

An important feature of our results is that they make virtually no structural assumptions on the predictors $\mathbf{x} \mapsto f(\mathbf{v}, \mathbf{x})$. In particular, they can represent arbitrary classes of neural networks (as well as other predictor classes). Thus, our results imply that target functions of the form $\mathbf{x} \mapsto \psi(\mathbf{w}^{\star\top}\mathbf{x})$, where ψ is periodic, would be difficult to learn using gradient-based methods, even if we allow improper learning and consider predictor classes of a different structure.

To explain how such results are attained, let us study a concrete special case (not necessarily in the context of neural networks). Consider the target function $\mathbf{x} \mapsto \cos(2\pi\mathbf{w}^{\star\top}\mathbf{x})$,

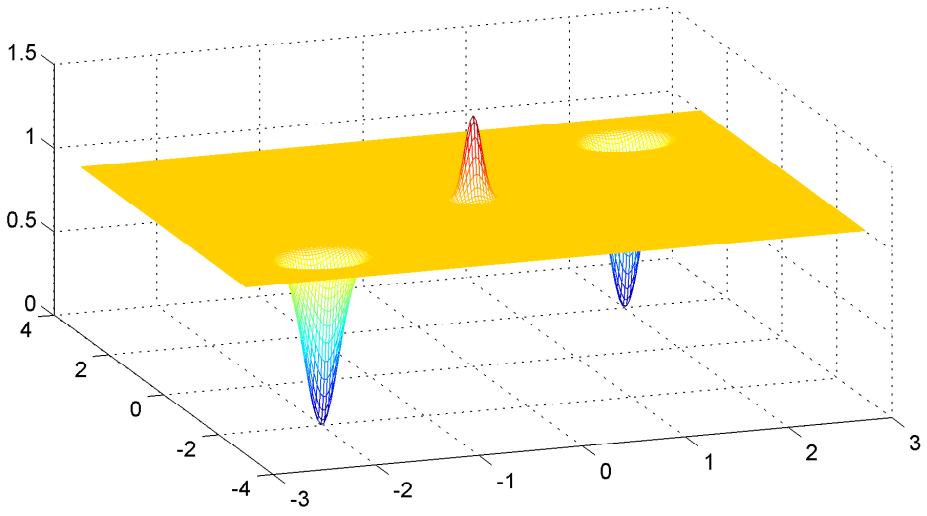

Figure 2: Graphical depiction of the objective function in Eq. (2), in 2 dimensions and where $\mathbf{w}^\star = (2, 2)$.

and the hypothesis class (parameterized by $\mathbf{w}$) of functions $\mathbf{x} \mapsto \cos(2\pi \mathbf{w}^\top \mathbf{x})$. Thus, Eq. (1) takes the form

$$\min_{\mathbf{w}} \mathbb{E}_{\mathbf{x}\sim\varphi^2} \left[\left(\cos(2\pi \mathbf{w}^\top \mathbf{x}) - \cos(2\pi \mathbf{w}^{\star\top} \mathbf{x}) \right)^2 \right]. \tag{2}$$

Furthermore, suppose the input distribution φ^2 is a standard Gaussian on $\mathbb{R}^d$. In two dimensions and for $\mathbf{w}^\star = (2, 2)$, the objective function in Eq. (1) turns out to have the form illustrated in Figure 2. This objective function has only three critical points: A global maximum at $\mathbf{0}$, and two global minima at $\mathbf{w}^\star$ and $-\mathbf{w}^\star$. Nevertheless, it would be difficult to optimize using gradient-based methods, since it is extremely *flat* everywhere except close to the critical points. As we will see shortly, the same phenomenon occurs in higher dimensions. In high dimensions, if the direction of $\mathbf{w}^\star$ is chosen randomly, we will be overwhelmingly likely to initialize far from the global minima, and hence will start in a flat plateau in which most gradient-based methods will stall[4].

We now turn to explain why Eq. (2) has the form shown in Figure 2. This will also help to illustrate our proof techniques, which apply much more generally. The main idea is to analyze the Fourier transform of Eq. (2). Letting $\cos_{\mathbf{w}}$ denote the function $\mathbf{x} \mapsto \cos(2\pi \mathbf{w}^\top \mathbf{x})$, we can write Eq. (2) as

$$\int \left(\cos(2\pi \mathbf{w}^\top \mathbf{x}) - \cos(2\pi \mathbf{w}^{\star\top} \mathbf{x}) \right)^2 \varphi^2(\mathbf{x}) d\mathbf{x} \;=\; \left\| \cos_{\mathbf{w}} \cdot \varphi - \cos_{\mathbf{w}^\star} \cdot \varphi \right\|^2,$$

where $\|\cdot\|$ is the standard norm over the space $L^2(\mathbb{R}^d)$ of square integrable functions. By standard properties of the Fourier transform (as described in Sec. 2), this squared norm

4. Although there are techniques to overcome flatness (e.g. by normalizing the gradient, see Nesterov 1984; Hazan et al. 2015), in our case the normalization factor will be huge and require extremely precise gradient information, which as discussed earlier, is unrealistic here.

of a function equals the squared norm of the function's Fourier transform, which equals in turn

$$\left\|\widehat{\cos_{\mathbf{w}}\cdot\varphi} - \widehat{\cos_{\mathbf{w}}\cdot\varphi}\right\|^2 \;=\; \left\|\widehat{\cos_{\mathbf{w}}} * \hat{\varphi} - \widehat{\cos_{\mathbf{w}^\star}} * \hat{\varphi}\right\|^2.$$

$\widehat{\cos_{\mathbf{w}}}(\boldsymbol{\xi})$ can be shown to equal $\frac{1}{2}\left(\delta(\boldsymbol{\xi}-\mathbf{w})+\delta(\boldsymbol{\xi}+\mathbf{w})\right)$, where $\delta(\cdot)$ is Dirac's delta function (a "generalized" function which satisfies $\delta(\mathbf{z})=0$ for all $\mathbf{z}\neq\mathbf{0}$, and $\int\delta(\mathbf{z})d\mathbf{z}=1$). Plugging this into the above and simplifying, we get

$$\begin{aligned}&\frac{1}{4}\left\|\hat{\varphi}(\cdot-\mathbf{w})+\hat{\varphi}(\cdot+\mathbf{w})-\hat{\varphi}(\cdot-\mathbf{w}^\star)-\hat{\varphi}(\cdot+\mathbf{w}^\star)\right\|^2\\ &\quad=\;\frac{1}{4}\int_{\boldsymbol{\xi}}\left|\hat{\varphi}(\boldsymbol{\xi}-\mathbf{w})+\hat{\varphi}(\boldsymbol{\xi}+\mathbf{w})-\hat{\varphi}(\boldsymbol{\xi}-\mathbf{w}^\star)-\hat{\varphi}(\boldsymbol{\xi}+\mathbf{w}^\star)\right|^2 d\boldsymbol{\xi},\end{aligned} \tag{3}$$

where $\hat{\varphi}(\cdot-\mathbf{w})$ stands for the function $\mathbf{x}\mapsto\hat{\varphi}(\mathbf{x}-\mathbf{w})$, etc. If φ^2 is a standard Gaussian, $\hat{\varphi}(\boldsymbol{\xi})$ can be shown to equal the Gaussian-like function $(4\pi)^{d/2}a^{-\|\boldsymbol{\xi}\|^2}$ where $a=\exp(4\pi^2)$. Plugging back, the expression above is proportional to

$$\int_{\boldsymbol{\xi}}\left(\left(a^{-\|\boldsymbol{\xi}-\mathbf{w}\|^2}+a^{-\|\boldsymbol{\xi}+\mathbf{w}\|^2}\right)-\left(a^{-\|\boldsymbol{\xi}-\mathbf{w}^\star\|^2}+a^{-\|\boldsymbol{\xi}+\mathbf{w}^\star\|^2}\right)\right)^2 d\boldsymbol{\xi}. \tag{4}$$

The expression in each inner parenthesis can be viewed as a mixture of two Gaussian-like functions, with centers at $\mathbf{w},-\mathbf{w}$ (or $\mathbf{w}^\star,-\mathbf{w}^\star$). Thus, if $\mathbf{w}$ is far from $\mathbf{w}^\star$, these two mixtures will have nearly disjoint support, and Eq. (4) will have nearly the same value regardless of $\mathbf{w}$—in other words, it is very flat. Since this equation is nothing more than a re-formulation of the original objective function in Eq. (2) (up to a constant), we get a similar behavior for Eq. (2) as well.

This behavior extends, however, much more generally than the specific objective in Eq. (2). First of all, we can replace the standard Gaussian distribution φ^2 by any distribution such that $\hat{\varphi}$ has a localized support. This would still imply that Eq. (3) refers to the difference of two functions with nearly disjoint support, and the same flatness phenomenon will occur. Second, we can replace the cos function by any periodic function ψ. By properties of the Fourier transform of periodic functions, we still get localized functions in the Fourier domain (more precisely, the Fourier transform will be localized around integer multiples of $\mathbf{w}$, up to scaling). Finally, instead of considering hypothesis classes of predictors $\mathbf{x}\mapsto\psi(\mathbf{w}^\top\mathbf{x})$ similar to the target function, we can consider quite arbitrary mappings $\mathbf{x}\mapsto f(\mathbf{w},\mathbf{x})$. Even though this function may no longer be localized in the Fourier domain, it is enough that only the target function $\mathbf{x}\mapsto\psi({\mathbf{w}^\star}^\top\mathbf{x})$ will be localized: That implies that regardless how f looks like, under a random choice of $\mathbf{w}^\star$, only a minuscule portion of the L_2 mass of f overlaps with the target function, hence getting sufficient signal on $\mathbf{w}^\star$ will be difficult.

As mentioned in the introduction, these techniques and observations are closely related to hardness results in the statistical queries literature, and can indeed be applied to that framework as shown recently in Song et al. (2017).

4.2 Formal Results

We now turn to provide a more formal statement of our results. The distributions we will consider consist of arbitrary mixtures of densities, whose square roots have Fourier transforms with rapidly decaying tails. More precisely, we have the following definition:

Definition 6 *Let $\epsilon(r)$ be some function from $[0,\infty)$ to $[0,1]$. A function $\varphi^2 : \mathbb{R}^d \to \mathbb{R}$ is* $\epsilon(r)$ Fourier-concentrated *if its square root φ belongs to $L^2(\mathbb{R}^d)$, and satisfies*

$$\|\hat{\varphi} \cdot \mathbf{1}_{\geq r}\| \;\; \leq \;\; \|\hat{\varphi}\|\, \epsilon(r),$$

where $\mathbf{1}_{\geq r}$ is the indicator function of $\{\mathbf{x} : \|\mathbf{x}\| \geq r\}$.

A canonical example is Gaussian distributions: Given a (non-degenerate, zero-mean) Gaussian density function φ^2 with covariance matrix Σ, its square root φ is proportional to a Gaussian with covariance 2Σ, and its Fourier transform $\hat{\varphi}$ is well-known to be proportional to a Gaussian with covariance $(2\Sigma)^{-1}$. By standard Gaussian concentration results, it follows that φ^2 is Fourier-concentrated with $\epsilon(r) = \exp(-\Omega(\lambda_{\min} r^2))$ where $\lambda_{\min}$ is the minimal eigenvalue of Σ. A similar bound can be shown when the Gaussian has some arbitrary mean. More generally, it is well-known that smooth functions (differentiable to sufficiently high order with integrable derivatives) have Fourier transforms with rapidly decaying tails. For example, if we consider the broad class of *Schwartz* functions (characterized by having values and all derivatives decaying faster than polynomially in r), then the Fourier transform of any such function is also a Schwartz function, which implies super-polynomial decay of $\epsilon(r)$ (see for instance Hunter and Nachtergaele 2001, Chapter 11 and Proposition 11.25).

We now formally state our main result for this section. We consider *any* predictor of the form $\mathbf{x} \mapsto f(\mathbf{v}, \mathbf{x})$, where f is some fixed function and $\mathbf{v}$ is a parameter vector coming from some domain $\mathcal{V}$, which we will assume w.l.o.g. to be a subset of some Euclidean space[5] (for example, f can represent a network of a given architecture, with weights specified by $\mathbf{v}$). When learning f based on data coming from an underlying distribution, we are essentially attempting to solve the optimization problem

$$\min_{\mathbf{v} \in \mathcal{V}} F_{\mathbf{w}^\star}(\mathbf{v}) := \mathbb{E}_{\mathbf{x} \sim \varphi^2} \left[\left(f(\mathbf{v}, \mathbf{x}) - \psi({\mathbf{w}^\star}^\top \mathbf{x}) \right)^2 \right].$$

Assuming that F is differentiable w.r.t. $\mathbf{w}$, any gradient-based method to solve this problem proceeds by computing (or approximating) $\nabla F_{\mathbf{w}^\star}(\mathbf{v})$ at various points $\mathbf{v}$. However, the following theorem shows that at *any* $\mathbf{v}$, and *regardless* of the type of predictor or network one is attempting to train, the gradient at $\mathbf{v}$ is virtually independent of the underlying target function, and hence provides very little signal:

Theorem 7 *Suppose that*

- *$\psi : \mathbb{R} \to [-1, +1]$ is a periodic function of period 1, which has bounded variation on every finite interval.*
- *φ^2 is a density function on $\mathbb{R}^d$, which can be written as a (possibly infinite) mixture $\varphi^2 = \sum_i \alpha_i \varphi_i^2$, where each φ_i^2 is $\epsilon(r)$ Fourier-concentrated.*
- *At some fixed $\mathbf{v}$, $\mathbb{E}_{\mathbf{x} \sim \varphi^2} \left\| \frac{\partial}{\partial \mathbf{v}} f(\mathbf{v}, \mathbf{x}) \right\|^2 \leq G_{\mathbf{v}}$ for some $G_{\mathbf{v}}$.*

5. More generally, our analysis is applicable to any separable Hilbert space.

Then for some universal positive constants c_1, c_2, c_3, if $d \geq c_1$, and if $\mathbf{w}^\star \in \mathbb{R}^d$ is a vector of norm $2r$ chosen uniformly at random, then

$$Var_{\mathbf{w}^\star}\left(\nabla F_{\mathbf{w}^\star}(\mathbf{v})\right) := \mathbb{E}_{\mathbf{w}^\star}\left\|\nabla F_{\mathbf{w}^\star}(\mathbf{v}) - \mathbb{E}_{\mathbf{w}^\star}[\nabla F_{\mathbf{w}^\star}(\mathbf{v})]\right\|^2 \leq \; c_2 G_{\mathbf{v}}\left(\exp(-c_3 d) + \sum_{n=1}^{\infty}\epsilon(nr)\right).$$

Note that bounded variation is weaker than, say, Lipschitz continuity. Also, we note that the mixture assumption is used in a rather limited sense: most of the proof is devoted to establishing the bound w.r.t. Fourier-concentrated densities, and the result is then extended to mixtures by Jensen's inequality (see the proof for details).

Assuming $\epsilon(r)$ decays rapidly with r—say, exponentially in r^2 as is the case for a mixture of Gaussians—we get that the bound in the theorem is on the order of $\exp(-\Omega(\min\{d, r^2\}))$. Overall, the theorem implies that if r, d are moderately large, the gradient of $F_{\mathbf{w}^\star}$ at any point $\mathbf{v}$ is extremely concentrated around a fixed value, independent of $\mathbf{w}^\star$. This implies that gradient-based methods, which attempt to optimize $F_{\mathbf{w}^\star}$ via gradient information, are unlikely to succeed.

One way to formalize this argument is to consider any iterative algorithm (possibly randomized), which relies on an *ε-approximate* gradient oracle to optimize $F_{\mathbf{w}^\star}$: At every iteration t, the algorithm chooses a point $\mathbf{v}_t \in \mathcal{V}$, and receives a vector $\mathbf{g}_t$ such that $|\nabla F_{\mathbf{w}^\star}(\mathbf{v}_t) - \mathbf{g}_t| \leq \varepsilon$. More formally, we can define such oracles and algorithms as follows:

Definition 8 (Approximate Gradient Oracle) *$O_{F,\epsilon}$ is an ε-approximate gradient oracle w.r.t. the function F on a domain $\mathcal{V}$, if given any input $\mathbf{v} \in \mathcal{V}$, it returns a fixed vector $\mathbf{g}$ such that $|\nabla F(\mathbf{v}) - \mathbf{g}| \leq \varepsilon$.*

Definition 9 (Approximate Gradient-Based Method) *Given a domain $\mathcal{V}$, we say that an algorithm A is a T-iterations, ε-approximate gradient-based method, if there exists some (deterministic or randomized) $\mathbf{v}_1 \in \mathcal{V}$ and (deterministic or randomized) functions $\{f_i : \mathcal{V}^i \to \mathcal{V}\}_{i=1}^{T-1}$, such that for any function F on $\mathcal{V}$, the output $\mathbf{v}_T \in \mathcal{V}$ of A given F can be written recursively as $\mathbf{v}_{t+1} = f_t(O_{F,\epsilon}(\mathbf{v}_1), \ldots, O_{F,\epsilon}(\mathbf{v}_t))$ for some approximate gradient oracle $O_{F,\epsilon}$.*

In our case, we will be interested in ε such that ε^3 is on the order of the bound in Thm. 7. Since the bound is extremely small for moderate d, r (say, smaller than machine precision), our definition of approximate gradient-based methods are a realistic model of gradient-based methods on finite-precision machines, even if one attempts to compute the gradients accurately. The following theorem implies that if the number of iterations is not extremely large (on the order of $1/\varepsilon$, e.g. $\exp(\Omega(\min\{d, r^2\})$ iterations for Gaussian mixtures), then with high probability, an approximate gradient-based method will return the same predictor independent of $\mathbf{w}^\star$. However, since the objective function $F_{\mathbf{w}^\star}$ is generally highly sensitive to the choice of $\mathbf{w}^\star$, this means that no such method can train a reasonable predictor.

Theorem 10 *Assume the conditions of Thm. 7, and let*

$$\varepsilon = \sqrt[3]{c_2(\sup_{\mathbf{v}\in\mathcal{V}} G_{\mathbf{v}})\left(\exp(-c_3 d) + \sum_{n=1}^{\infty}\epsilon(nr)\right)}$$

be the cube root of the bound specified there (uniformly over all $\mathbf{v} \in \mathcal{V}$). Then there exist a choice of approximate gradient oracles $\{O_{F_{\mathbf{w}^*},\varepsilon} : \|\mathbf{w}^*\| = 2r\}$, such that the following holds for any $p \in (0,1)$ and any $\lfloor p/\varepsilon \rfloor$-iterations, ε-approximate gradient-based method A: Conditioned on an event which holds with probability $1-p$ over the random choice of $\mathbf{w}^\star$, the distribution of the output of A given $F_{\mathbf{w}^*}$ is fixed independent of $\mathbf{w}^*$.

5. Proofs

5.1 Proof of Thm. 4

Let P_M denote the whitening matrix employed if we transform the instances X by some invertible $d \times d$ matrix M (that is, X becomes MX), and P the whitening matrix employed for the original data.

Using the same notation as in the theorem, it is easily verified that $PX = V^\top$, and $P_M MX = V_M^\top$, where $U_M D_M V_M^\top$ is an SVD decomposition of the matrix MX. Since both $V^\top$ and $V_M^\top$ are $\mathrm{Rank}(X) \times m$ matrices with rows consisting of orthonormal vectors, they are related by an orthogonal transformation (i.e. there is an orthogonal matrix R_M such that $R_M V^\top = V_M^\top$). Therefore, $R_M PX = P_M MX$. Since the data is fed to an orthogonally-invariant algorithm, its output W_M satisfies $W_M^\top P_M MX = W^\top PX$. This in turn implies $W_M^\top R_M PX = W^\top PX$, and hence $W_M^\top R_M V^\top = W^\top V^\top$. Multiplying both sides on the right by V and taking a transpose, we get that $R_M^\top W_M = W$, and hence $W_M = R_M W$. In words, W and W_M are the same up to an orthogonal transformation R_M depending on M. Therefore,

$$(P_M^\top W_M)^\top MX = W_M^\top P_M MX = W^\top R_M^\top R_M PX = W^\top PX = (P^\top W)^\top X,$$

so we see that the returned predictor makes the same predictions over the data set, independent of the transformation matrix M.

5.2 Proof of Thm. 5

We start with the following auxiliary theorem, which reduces the hardness result of Daniely and Shalev-Shwartz (2016) to one about neural networks of the type we discuss here:

Theorem 11 *Under the assumption stated in Daniely and Shalev-Shwartz (2016), the following holds for any $n_d = \omega(\log(d))$ (as $d \to \infty$):*

There is no algorithm running in time $\mathrm{poly}(d, 1/\epsilon)$, which for any distribution $\mathcal{D}$ on $\{0,1\}^d$, and any $h(W^\top \mathbf{x}) = [\sum_{i=1}^{n_d} [\langle \mathbf{w}_i, \mathbf{x}\rangle]_+]_{[0,1]}$ (where $W = [\mathbf{w}_1, \mathbf{w}_2 \ldots \mathbf{w}_{n_d}]$ and $\max_i \|\mathbf{w}_i\|$ is $\mathcal{O}(d)$), given only access to samples $(\mathbf{x}, h(W^\top \mathbf{x}))$ where $\mathbf{x} \sim \mathcal{D}$, returns with high probability a function f such that

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\Big(f(\mathbf{x}) - h(W^\top \mathbf{x})\Big)^2\right] \leq \epsilon.$$

Proof Suppose for the sake of contradiction that there exists an algorithm $\mathcal{A}$ which for any distribution and $h(W^\top \mathbf{x})$ as described in the theorem, returns a function f such that $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[(f(\mathbf{x}) - h(W^\top \mathbf{x}))^2] \leq \epsilon$ with high probability.

In particular, let us focus on distributions $\mathcal{D}$ supported on $\{0,1\}^{d-1}\times\{1\}$. For these distributions, we argue that any intersection of halfspaces on $\mathbb{R}^{d-1}$ specified by $\mathbf{w}_1,\ldots,\mathbf{w}_{n_d}\in\mathbb{R}^{d-1}$ with integer coordinates, and integer $b_1,\ldots,b_n$, can be specified as $\mathbf{x}\mapsto 1-h(W^\top\mathbf{x})$ for some function h as described in the theorem statement. To see this, note that for any $\mathbf{w}_i, b_i$ and $\mathbf{x}=(\mathbf{x}',1)$ in the support of $\mathcal{D}$, $[\langle(-\mathbf{w}_i,b_i),\mathbf{x}\rangle]_+ = [-\langle\mathbf{w}_i,\mathbf{x}'\rangle+b_i]_+$ is a non-negative integer, hence

$$\left[\sum_{i=1}^{n_d}[\langle(-\mathbf{w}_i,b_i),\mathbf{x}\rangle]_+\right]_{[0,1]} = \left[\sum_{i=1}^{n_d}[-\langle\mathbf{w}_i,\mathbf{x}'\rangle+b_i]_+\right]_{[0,1]} = \bigvee_{i=1}^{n_d}\left(-\langle\mathbf{w}_i,\mathbf{x}'\rangle+b_i>0\right)$$
$$= \bigvee_{i=1}^{n_d}\left(\langle\mathbf{w}_i,\mathbf{x}'\rangle<b_i\right) = \neg\left(\bigwedge_{i=1}^{n_d}\left(\langle\mathbf{w}_i,\mathbf{x}'\rangle\geq b_i\right)\right).$$

Therefore, for any distribution over examples labelled by an intersection of halfspaces $\mathbf{x}\mapsto 1-h(W^\top\mathbf{x})$ (with integer-valued coordinates and bounded norms), by feeding $\mathcal{A}$ with $\{\mathbf{x}_i,1-y_i\}_{i=1}^m$, the algorithm returns a function f, such that with high probability, $\mathbb{E}_{\mathbf{x}}\left(f(\mathbf{x})-h(W^\top\mathbf{x})\right)^2\leq\epsilon$, and therefore

$$\mathbb{E}_{\mathbf{x}}\left((1-f(\mathbf{x}))-(1-h(W^\top\mathbf{x}))\right)^2\leq\epsilon.$$

In particular, if we consider the Boolean function $\tilde{f}(\mathbf{x})=1-\mathrm{rnd}(f(\mathbf{x}))$, where $\mathrm{rnd}(z)=0$ if $z\leq 1/2$ and $\mathrm{rnd}(z)=1$ if $z>1/2$, we argue that $\Pr_{\mathbf{x}}(\tilde{f}(\mathbf{x})\neq 1-h(W^\top\mathbf{x}))\leq 8\epsilon$. Since ϵ is arbitrary, and $1-h(W^\top\mathbf{x})$ specifies an intersection of halfspaces, this would contradict the hardness result of Daniely and Shalev-Shwartz (2016), and therefore prove the theorem. This argument follows from the following chain of inequalities, where $\mathbf{1}$ denotes the indicator function:

$$\begin{aligned}\Pr\left(\tilde{f}(\mathbf{x}))\neq g(\mathbf{x})\right) &= \Pr(f(\mathbf{x})>1/2\ \wedge\ g(\mathbf{x})=1)+\Pr(f(\mathbf{x})\leq 1/2\ \wedge\ g(\mathbf{x})=0)\\ &= \mathbb{E}\left[\mathbf{1}\left(f(\mathbf{x})>1/2\ \wedge\ g(\mathbf{x})=1\right)\right]+\mathbb{E}\left[\mathbf{1}\left(f(\mathbf{x})\leq 1/2\ \wedge\ g(\mathbf{x})=0\right)\right]\\ &\leq \mathbb{E}\left[4\left((1-f(\mathbf{x}))-g(\mathbf{x})\right)^2\right]+\mathbb{E}\left[4\left((1-f(\mathbf{x}))-g(\mathbf{x})\right)^2\right]\\ &\leq 8\cdot\mathbb{E}\left[\left((1-f(\mathbf{x}))-g(\mathbf{x})\right)^2\right]\ \leq\ 8\epsilon.\end{aligned}$$

■

Proposition 12 *Thm. 11 holds even if we restrict $\mathbf{w}_1,\ldots,\mathbf{w}_{n_d}$ to be linearly independent, with $s_{\min}(W)\geq 1$.*

Proof Suppose for the sake of contradiction that there exists an algorithm $\mathcal{A}$ which succeeds for any W as stated above. We will describe how to use $\mathcal{A}$ to get an algorithm which succeeds for any W as described in Thm. 11, hence reaching a contradiction.

Specifically, suppose we have access to samples $(\mathbf{x},h(W^\top\mathbf{x}))$, where $\mathbf{x}$ is supported on $\{0,1\}^d$, and where W is any matrix as described in Thm. 11. We do the following: We

map every $\mathbf{x}$ to $\tilde{\mathbf{x}} \in \{0,1\}^{d+n_d}$ by $\tilde{\mathbf{x}} = (\mathbf{x}, 0, \ldots, 0)$, run $\mathcal{A}$ on the transformed samples $(\tilde{\mathbf{x}}, h(W^\top \mathbf{x}))$ to get some predictor $\tilde{f} : \{0,1\}^{d+n_d} \mapsto \mathbb{R}$, and return the predictor $f(\mathbf{x}) = \tilde{f}((\mathbf{x}, 0, \ldots, 0))$.

To see why this reduction works, we note that the mapping $\mathbf{x} \mapsto \tilde{\mathbf{x}}$ we have defined, where $\mathbf{x}$ is distributed according to $\mathbf{x}$, induces a distribution $\tilde{\mathcal{D}}$ on $\{0,1\}^{d+n_d}$. Let $\tilde{W}$ be the $(d+n_d)\times n_d$ matrix $[W; I_{n_d}]$ (that is, we add another $n_d \times n_d$ unit matrix below W). We have $\tilde{W}^\top \tilde{W} = W^\top W + I_{n_d}$, so the minimal eigenvalue of $\tilde{W}^\top \tilde{W}$ is at least 1, hence $s_{\min}(\tilde{W}) \geq 1$, so $\tilde{W}$ satisfies the conditions in the proposition. Moreover, the norm of each column of $\tilde{W}$ is larger than the norm of the corresponding column in W by at most 1, so the norm constraint in Thm. 11 still holds. Finally, $\tilde{W}\tilde{\mathbf{x}} = W\mathbf{x}$ for all $\mathbf{x}$, and therefore $h_{\tilde{W}}(\tilde{\mathbf{x}}) = h_W(\mathbf{x})$. Thus, the distribution of $(\tilde{\mathbf{x}}, h(W^\top \mathbf{x})) = (\tilde{\mathbf{x}}, h(\tilde{W}^\top \mathbf{x}))$ (which is used to feed the algorithm $\mathcal{A}$) is a valid distribution corresponding to the conditions of the proposition and Thm. 11 (only in dimension $d + n_d \leq 2d$ instead of d), so $\mathcal{A}$ returns with high probability a predictor $\tilde{f}$ such that

$$\mathbb{E}_{\tilde{\mathbf{x}}\sim\tilde{\mathcal{D}}}\left[\left(\tilde{f}(\tilde{\mathbf{x}}) - h(\tilde{W}^\top \tilde{\mathbf{x}})\right)^2\right] \leq \epsilon.$$

However, $\tilde{f}(\tilde{\mathbf{x}}) = f(\mathbf{x})$, $h(\tilde{W}^\top \tilde{\mathbf{x}}) = h(W^\top \mathbf{x})$, so the returned predictor f satisfies

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\left(f(\mathbf{x}) - h(W^\top \mathbf{x})\right)^2\right] \leq \epsilon.$$

This contradicts Thm. 11, which states that no efficient algorithm can return such a predictor for *any* sufficiently large dimension d and norm bound $\mathcal{O}(d)$. ■

In the definitions of orthogonal invariance and linear invariance, we only required the invariance to hold with respect to instances $\mathbf{x}_i$ in the data set. A stronger condition is that the invariance is satisfied for any $\mathbf{x} \in \mathbb{R}^d$. However, the following lemma shows that invariance w.r.t. a data set sampled i.i.d. from some distribution is sufficient to imply invariance w.r.t. "nearly all" $\mathbf{x}$ (under the same distribution):

Lemma 13 *Suppose the data set $\{\mathbf{x}_i, y_i\}_{i=1}^m$ is sampled i.i.d. from some distribution (where $\mathbf{x}_i \in \mathbb{R}^d$), then the following holds with probability at least $1-\delta$ for any $\delta \in (0,1)$: For any invertible M and linearly-invariant algorithm (or orthogonal M and orthogonally-invariant algorithm), conditioned on the algorithm's internal randomness, the returned matrices W and W_M (with respect to the original data and the data transformed by M respectively) satisfy*

$$\Pr_{\mathbf{x}}(W_M^\top M\mathbf{x} \neq W^\top \mathbf{x}) \leq \frac{d}{\delta(m+1)}.$$

Proof It is enough to prove that with probability at least $1-\delta$ over the sampling of $\mathbf{x}_1, \ldots, \mathbf{x}_m$,

$$\Pr_{\mathbf{x}}(\mathbf{x} \notin \operatorname{span}(\mathbf{x}_1, \ldots, \mathbf{x}_m)) \leq \frac{d}{\delta(m+1)}. \tag{5}$$

This is because the event $W_M^\top M\mathbf{x}_i = W^\top \mathbf{x}_i$ for all i means that $W_M^\top M\mathbf{x} = W^\top \mathbf{x}$ for any $\mathbf{x}$ in the span of $\mathbf{x}_1, \ldots, \mathbf{x}_m$.

Let $\mathbf{x}_1, \ldots, \mathbf{x}_{m+1}$ be sampled i.i.d. according to $\mathcal{D}$. Considering probabilities over this sample, we have

$$\sum_{j=1}^{m+1} \Pr\left(\mathbf{x}_j \notin \text{span}(\mathbf{x}_1, \ldots, \mathbf{x}_{j-1})\right) \;=\; \mathbb{E}\left[\sum_{j=1}^{m+1} \mathbf{1}\left(\mathbf{x}_j \notin \text{span}(\mathbf{x}_1, \ldots, \mathbf{x}_{j-1})\right)\right] \leq d, \quad (6)$$

where the latter inequality is because each $\mathbf{x}_j$ is a d-dimensional vector, hence the number of times we can get a vector not in the span of the previous ones is at most d. Moreover, since the vectors are sampled i.i.d, we have

$$\Pr(\mathbf{x}_{j+1} \notin \text{span}(\mathbf{x}_1 \ldots \mathbf{x}_j)) \leq \Pr(\mathbf{x}_{j+1} \notin \text{span}(\mathbf{x}_1 \ldots \mathbf{x}_{j-1})) = \Pr(\mathbf{x}_j \notin \text{span}(\mathbf{x}_1 \ldots \mathbf{x}_{j-1})),$$

so the probabilities in Eq. (6) monotonically decrease with j. Thus, Eq. (6) implies

$$(m+1)\Pr\left(\mathbf{x}_{m+1} \notin \text{span}(\mathbf{x}_1, \ldots, \mathbf{x}_m)\right) \leq d \quad \Rightarrow \quad \Pr\left(\mathbf{x}_{m+1} \notin \text{span}(\mathbf{x}_1, \ldots, \mathbf{x}_m)\right) \leq \frac{d}{m+1}.$$

This is equivalent to

$$\mathbb{E}_{\mathbf{x}_1,\ldots,\mathbf{x}_m \sim \mathcal{D}}\left[\Pr_{\mathbf{x}_{m+1}}\left(\mathbf{x}_{m+1} \notin \text{span}(\mathbf{x}_1, \ldots, \mathbf{x}_m) | \mathbf{x}_1, \ldots, \mathbf{x}_m\right)\right] \leq \frac{d}{m+1},$$

so by Markov's inequality, with probability at least $1-\delta$ over the sampling of $\mathbf{x}_1, \ldots, \mathbf{x}_m$,

$$\Pr_{\mathbf{x}_{m+1}}\left(\mathbf{x}_{m+1} \notin \text{span}(\mathbf{x}_1, \ldots, \mathbf{x}_m)\right) \leq \frac{d}{\delta(m+1)}.$$

Since $\mathbf{x}_{m+1}$ is sampled independently, Eq. (5) and hence the lemma follows. ■

With these results in hand, we can finally turn to prove Thm. 5. Suppose for the sake of contradiction that there exists an efficient linearly-invariant algorithm $\mathcal{A}$, which for any distribution $\mathcal{D}$ supported on vectors of norm $\mathcal{O}\left(d\sqrt{2dn_d}\right) / \min\{1, s_{\min}(W_\star)\}$, returns w.h.p. a predictor $\mathbf{x} \mapsto f(\tilde{W}^\top \mathbf{x})$ such that

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}^\star}\left[\left(f(\tilde{W}^\top \mathbf{x}) - h(W_\star^\top \mathbf{x})\right)^2\right] \leq \epsilon.$$

We will show that the very same algorithm, if given $\text{poly}(d, 1/\epsilon)$ samples, can successfully learn w.r.t. *any* $d \times n_d$ matrix W and any distribution $\mathcal{D}$ satisfying Proposition 12 and Thm. 11, contradicting those results.

In what follows, we assume without loss of generality that f maps to $[0,1]$: If that is not the case, we can simply consider the predictor $\hat{f}(W^\top \mathbf{x})$, where $\hat{f}(z) = \max\{0, \min\{1, f(z)\}\}$, and note that since h returns values in $[0,1]$, then for any input $\mathbf{x}$,

$$((\hat{f}(W^\top \mathbf{x}) - h(W_\star^\top \mathbf{x}))^2 \;\leq\; (f(W^\top \mathbf{x}) - h(W_\star^\top \mathbf{x}))^2,$$

hence the expected squared loss of $\mathbf{x} \mapsto \hat{f}(W^\top \mathbf{x})$ is only smaller than $\mathbf{x} \mapsto f(W^\top \mathbf{x})$

Indeed, let W and $\mathcal{D}$ be an arbitrary matrix and distribution which satisfy the conditions of both Thm. 11 and 12 (namely, $\mathcal{D}$ is a distribution on $\{0,1\}^d$, and $W = [\mathbf{w}_1, \ldots, \mathbf{w}_{n_d}]$

satisfies $\max_i \|\mathbf{w}_i\| \leq \mathcal{O}(d)$ as well as $s_{\min}(W) \geq 1$). We first argue that there exists a $d \times d$ invertible matrix M such that

$$W = M^\top W_\star \quad , \quad \|M\| = \frac{\mathcal{O}(d\sqrt{2n_d})}{\min\{1, s_{\min}(W_\star)\}}. \tag{7}$$

To see this, note that W and $W_\star$ are of the same size and our conditions imply that both of them have full column rank. Thus, we can simply augment them to invertible $d \times d$ matrices $[W\ \hat{W}]$ and $[W_\star\ \hat{W}_\star]$, where the columns of $\hat{W}$ (respectively $\hat{W}_\star$) are an orthonormal basis for the subspace orthogonal to the column space of W (respectively $W_\star$), and choosing $M^\top = [W\ \hat{W}][W_\star\ \hat{W}_\star]^{-1}$. Thus,

$$\|M\| \leq \left\|[W\ \hat{W}]\right\| \cdot \left\|[W_\star\ \hat{W}_\star]^{-1}\right\|. \tag{8}$$

$\left\|[W\ \hat{W}]\right\|$ can be upper bounded by the Frobenius norm, which by the assumption on W from Thm. 11 and the fact that $\hat{W}$ has orthogonal columns, is $\sqrt{\mathcal{O}(d)^2 \cdot n_d + 1 \cdot (d - n_d)} = \mathcal{O}(\sqrt{d^2 n_d}) = \mathcal{O}(d\sqrt{n_d})$. Also, $\left\|[W_\star\ \hat{W}_\star]^{-1}\right\|$ can be upper bounded by the inverse square root of the smallest eigenvalue of

$$[W_\star\ \hat{W}_\star]^\top [W_\star\ \hat{W}_\star] \ = \ \begin{bmatrix} W_\star^\top W_\star & 0 \\ 0 & \hat{W}_\star^\top \hat{W}_\star \end{bmatrix} \ = \ \begin{bmatrix} W_\star^\top W_\star & 0 \\ 0 & I \end{bmatrix} \tag{9}$$

(where I is the unit matrix), which equals $1/\min\{1, s_{\min}(W_\star)\}$. Plugging these bounds into Eq. (8), we get Eq. (7).

Now, consider the following: Suppose we run the algorithm $\mathcal{A}$ using the data points $(M\mathbf{x}_i, h(W_\star^\top(M\mathbf{x}_i)))$, $i = 1, 2, \ldots, m$, where $\mathbf{x}_i$ is sampled from $\mathcal{D}$. Since $\mathbf{x}_i \in \{0,1\}^d$, and $\|\mathbf{x}_i\| \leq \sqrt{d}$, it follows from Eq. (7) that $M\mathbf{x}_i$ is always of norm at most $\|M\|\,\|\mathbf{x}_i\| \leq \|M\|\sqrt{d} = \mathcal{O}(d\sqrt{dn_d})/\min\{1, s_{\min}(W_\star)\}$, and the outputs correspond to the network specified by $W_\star$. Therefore, by assumption, the algorithm $\mathcal{A}$ would return w.h.p. a matrix $\tilde{W}_M$ such that

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\left(f(\tilde{W}_M^\top(M\mathbf{x})) - h(W_\star^\top(M\mathbf{x}))\right)^2\right] \leq \epsilon.$$

By Eq. (7), $W_\star^\top M = (M^\top W_\star)^\top = W^\top$, so this is equivalent to

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\left(f((\tilde{W}_M^\top M\mathbf{x}) - h(W^\top\mathbf{x}))\right)^2\right] \leq \epsilon. \tag{10}$$

Let $\tilde{W}_I$ be the matrix returned by $\mathcal{A}$ if we had fed it with the samples $\{(\mathbf{x}_i, h(W^\top\mathbf{x}_i))\}_{i=1}^m$ (or equivalently, $\{(\mathbf{x}_i, h(W_\star^\top(M\mathbf{x}_i)))\}_{i=1}^m$)[6]. Let $E_{\mathbf{x}}$ be the event (conditioned on the samples used by the algorithm) that a freshly sampled $\mathbf{x} \sim \mathcal{D}$ satisfies $\tilde{W}_M^\top M\mathbf{x} = \tilde{W}_I^\top\mathbf{x}$. By Lemma 13, w.h.p. over the samples fed to the algorithm, $\Pr_{\mathbf{x}}(E_{\mathbf{x}}) = 1 - \mathcal{O}(d/m)$. Therefore,

6. Note that if the algorithm is stochastic, both $\tilde{W}_M$ and $\tilde{W}_I$ are not fixed given the data, but also depend on the algorithm's internal randomness. However, the proof will still follow by conditioning on any possible realization of this randomness.

w.h.p. over the samples $\mathbf{x}_1, \ldots, \mathbf{x}_m$,

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\left(f(\tilde{W}_I^\top \mathbf{x}) - h(W^\top \mathbf{x})\right)^2\right] \\
&= \Pr_{\mathbf{x}}(E_{\mathbf{x}}) \cdot \mathbb{E}_{\mathbf{x}}\left[\left(f(\tilde{W}_I^\top \mathbf{x}) - h(W^\top \mathbf{x})\right)^2 \middle| E_{\mathbf{x}}\right] + \Pr_{\mathbf{x}}(\neg E_{\mathbf{x}}) \cdot \mathbb{E}_{\mathbf{x}}\left[\left(f(\tilde{W}_I^\top \mathbf{x}) - h(W^\top \mathbf{x})\right)^2 \middle| \neg E_{\mathbf{x}}\right] \\
&\leq \Pr_{\mathbf{x}}(E_{\mathbf{x}}) \cdot \mathbb{E}_{\mathbf{x}}\left[\left(f(\tilde{W}_M^\top M\mathbf{x}) - h(W^\top \mathbf{x})\right)^2 \middle| E_{\mathbf{x}}\right] + \Pr_{\mathbf{x}}(\neg E_{\mathbf{x}}) \cdot 1 \\
&= \mathbb{E}_{\mathbf{x}}\left[\left(f(\tilde{W}_M^\top M\mathbf{x}) - h(W^\top \mathbf{x})\right)^2 \mathbf{1}(E_{\mathbf{x}})\right] + \mathcal{O}\left(\frac{d}{m}\right) \\
&\leq \mathbb{E}_{\mathbf{x}}\left[\left(f(\tilde{W}_M^\top M\mathbf{x}) - h(W^\top \mathbf{x})\right)^2\right] + \mathcal{O}\left(\frac{d}{m}\right) \quad \leq \quad \epsilon + \mathcal{O}\left(\frac{d}{m}\right),
\end{aligned}$$

where we used the facts that both f and h map to $[0,1]$, a union bound and Eq. (10). Now, recall that $\tilde{W}_I$ refers to the output of the algorithm, given samples $\{(\mathbf{x}_i, h(W^\top \mathbf{x}_i))\}_{i=1}^m$ where $m = \mathrm{poly}(d, 1/\epsilon)$. Thus, we have shown that w.h.p., as long as the algorithm is fed with $m \geq d/\epsilon$ samples[7], the algorithm returns $\tilde{W}_I$ which satisfies

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\left(f(\tilde{W}_I^\top \mathbf{x}) - h(W^\top \mathbf{x})\right)^2\right] \;=\; \mathcal{O}(\epsilon).$$

This means that the algorithm succesfully learns the hypothesis $\mathbf{x} \mapsto h(W^\top \mathbf{x})$ with respect to the distribution $\mathcal{D}$. Since ϵ is arbitrarily small and $W, \mathcal{D}$ were chosen arbitrarily, the result follows.

5.3 Proof of Thm. 7

To prove the theorem, we will require some tools from Fourier analysis on Euclidean space. We will consider functions from $\mathbb{R}^d$ to the reals $\mathbb{R}$ or complex numbers $\mathbb{C}$, and view them as elements in the Hilbert space $L^2(\mathbb{R}^d)$ of square integrable functions, equipped with the inner product

$$\langle f, g\rangle = \int_{\mathbf{x}} f(\mathbf{x}) \cdot \overline{g(\mathbf{x})} d\mathbf{x}$$

and the norm $\|f\| = \sqrt{\langle f, f\rangle}$. We use fg or $f \cdot g$ as shorthand for the function $\mathbf{x} \mapsto f(\mathbf{x})g(\mathbf{x})$. Any function $f \in L^2(\mathbb{R}^d)$ has a Fourier transform $\hat{f} \in L^2(\mathbb{R}^d)$, which for absolutely integrable functions can be defined as

$$\hat{f}(\mathbf{w}) = \int \exp\left(-2\pi i \mathbf{x}^\top \mathbf{w}\right) f(\mathbf{x}) d\mathbf{x}, \tag{11}$$

where $\exp(iz) = \cos(z) + i \cdot \sin(z)$, i being the imaginary unit. In the proofs, we will use the following well-known properties of the Fourier transform:

- Linearity: For scalars a, b and functions f, g, $\widehat{af + bg} = a\hat{f} + b\hat{g}$.

7. Even if the algorithm does not require that many samples, we can still artificially add more samples—these are merely used to ensure that its linear invariance is with respect to a sufficiently large data set.

- Isometry: $\langle f, g\rangle = \langle \hat{f}, \hat{g}\rangle$ and $\|f\| = \|\hat{f}\|$.
- Convolution: $\widehat{fg} = \hat{f} * \hat{g}$, where $*$ denotes the convolution operation: $(f * g)(\mathbf{w}) = \int f(\mathbf{z}) \cdot g(\mathbf{w} - \mathbf{z})\, d\mathbf{z}$.

We now turn to prove the theorem. First, we note that for any function q, $\mathbb{E}_{\mathbf{x}\sim\varphi^2}[q(\mathbf{x})] = \sum_i \alpha_i \cdot \mathbb{E}_{\mathbf{x}\sim\varphi_i^2}[q(\mathbf{x})]$. Thus, it is enough to prove the bound in the theorem when φ^2 consists of a single element whose square root is Fourier-concentrated. For a mixture $\varphi^2 = \sum_i \alpha_i \varphi_i^2$, the result follows by applying the bound for each φ_i^2 individually, and using Jensen's inequality.

To simplify notation a bit, we let $\psi_{\mathbf{w}}(\cdot)$ stand for the function $\psi(\langle \mathbf{w}, \cdot\rangle)$, and let $h(\cdot - \mathbf{v})$ (where $\mathbf{v}$ is some vector and h is a function on $\mathbb{R}^d$) stand for the function $\mathbf{x} \mapsto h(\mathbf{x} - \mathbf{v})$. Also, we will use several times the fact that for any two $L^2(\mathbb{R}^d)$ functions h_1, h_2,

$$\langle h_1(\cdot - \mathbf{v}), h_2(\cdot - \mathbf{v})\rangle \;=\; \int h_1(\mathbf{x} - \mathbf{v})\overline{h_2(\mathbf{x} - \mathbf{v})} d\mathbf{x} \;=\; \int h_1(\mathbf{x})\overline{h_2(\mathbf{x})} d\mathbf{x} = \langle h_1, h_2\rangle\,.$$

In other words, inner products (and hence also norms) in $L^2(\mathbb{R}^d)$ are invariant to a shift in coordinates.

The proof is a combination of a few lemmas, presented below.

Lemma 14 *For any* $\mathbf{w}$*, it holds that* $\psi_{\mathbf{w}}\varphi \in L^2(\mathbb{R}^d)$*, and satisfies*

$$\widehat{\psi_{\mathbf{w}}\varphi}(\mathbf{x}) \;=\; \sum_{z\in\mathbb{Z}} a_z \cdot \hat{\varphi}(\mathbf{x} - z\mathbf{w})$$

for any $\mathbf{x}$*, where* $\mathbb{Z}$ *is the set of integers and* a_z *are complex-valued coefficients (corresponding to the Fourier series expansion of* ψ*, hence depending only on* ψ*) which satisfy* $\sum_{z\in\mathbb{Z}} |a_z|^2 \leq 1$*.*

Proof First, we note that $\psi_{\mathbf{w}}\varphi \in L^2(\mathbb{R}^d)$, since both $\psi_{\mathbf{w}}$ and φ are locally integrable[8] by the theorem's conditions, and satisfy

$$\|\psi_{\mathbf{w}}\varphi\|^2 = \int \psi_{\mathbf{w}}^2(\mathbf{x})\varphi^2(\mathbf{x})d\mathbf{x} = \int \psi^2(\mathbf{w}^\top\mathbf{x})\varphi^2(\mathbf{x})d\mathbf{x} \leq \int \varphi^2(\mathbf{x})d\mathbf{x} = 1 < \infty.$$

As a result, $\widehat{\psi_{\mathbf{w}}\varphi}$ exists as a function in $L^2(\mathbb{R}^d)$. Since ψ is a function of bounded variation, it is equal everywhere to its Fourier series expansion:

$$\psi(x) = \sum_{z\in\mathbb{Z}} a_z \exp\left(2\pi i z x\right),$$

where i is the imaginary unit (note that since ψ is real-valued, the imaginary components eventually cancel out, but it will be more convenient for us to represent the Fourier series in this compact form). By Parseval's identity, $\sum_z |a_z|^2 = \int_{-1/2}^{1/2} \psi^2(x)dx$, which is at most 1 (since $\psi(x) \in [-1, +1]$).

8. Namely, integrable on any compact subset of their domain.

Based on this equation, we have

$$\psi_{\mathbf{w}}(\mathbf{x}) = \psi(\mathbf{w}^\top \mathbf{x}) = \sum_{z\in\mathbb{Z}} a_z \exp\left(2\pi i z \mathbf{w}^\top \mathbf{x}\right).$$

We now wish to compute the Fourier transform of the above[9]. First, we note that the Fourier transform of $\exp(2\pi i \langle \mathbf{v}, \cdot \rangle)$ is given by $\delta(\cdot - \mathbf{v})$, where δ is the Dirac delta function (a so-called generalized function which satisfies $\delta(\mathbf{x}) = 0$ for all $\mathbf{x} \neq \mathbf{0}$, and $\int \delta(\mathbf{x}) d\mathbf{x} = 1$). Based on this and the linearity of the Fourier transform, we have that

$$\hat{\psi}_{\mathbf{w}}(\mathbf{x}) \;=\; \sum_{z\in\mathbb{Z}} a_z \cdot \delta\left(\mathbf{x} - z\mathbf{w}\right),$$

and therefore, by the convolution property of the Fourier transform, we have

$$\begin{aligned}
\widehat{\psi_{\mathbf{w}}\varphi}(\mathbf{x}) = \left(\hat{\psi}_{\mathbf{w}} * \hat{\varphi}\right)(\mathbf{x}) &= \int \hat{\psi}_{\mathbf{w}}(\mathbf{z}) \cdot \hat{\varphi}(\mathbf{x} - \mathbf{z}) d\mathbf{z} \\
&= \sum_{z\in\mathbb{Z}} a_z \cdot \int \delta(\mathbf{z} - z\mathbf{w}) \cdot \hat{\varphi}(\mathbf{x} - \mathbf{z}) d\mathbf{z} \\
&= \sum_{z\in\mathbb{Z}} a_z \cdot \hat{\varphi}(\mathbf{x} - z\mathbf{w})
\end{aligned}$$

as required. ■

Lemma 15 *For any distinct integers $z_1 \neq z_2$ and any $\mathbf{w}$ such that $\|\mathbf{w}\| = 2r$, it holds that*

$$\left\langle \; |\hat{\varphi}(\cdot - z_1\mathbf{w})| \;,\; |\hat{\varphi}(\cdot - z_2\mathbf{w})| \; \right\rangle \;\leq\; 2 \cdot \epsilon(|z_1 - z_2| r).$$

Proof Let $\Delta = |z_2 - z_1| r$, and $\mathbf{v} = (z_2 - z_1)\mathbf{w}$, so $\mathbf{v}$ is a vector of norm 2Δ. Since the inner product is invariant to shifting the coordinates, we can assume without loss of generality that $z_1 = 0$, and our goal is to bound $\langle |\hat{\varphi}|, |\hat{\varphi}(\cdot - \mathbf{v})| \rangle$.

Using the convention that $\mathbf{1}_{\leq\Delta}$ is the indicator of $\{\mathbf{x} : \|\mathbf{x}\| \leq \Delta\}$, and $\mathbf{1}_{>\Delta}$ is the indicator for its complement, we have

$$\begin{aligned}
\left\langle |\hat{\varphi}|, |\hat{\varphi}(\cdot - \mathbf{v})| \right\rangle &= \left\langle |\hat{\varphi}| \;,\; |\hat{\varphi}(\cdot - \mathbf{v})| \, \mathbf{1}_{\leq\Delta} \right\rangle + \left\langle |\hat{\varphi}| \;,\; |\hat{\varphi}(\cdot - \mathbf{v})| \, \mathbf{1}_{>\Delta} \right\rangle \\
&= \left\langle |\hat{\varphi}| \;,\; |\hat{\varphi}(\cdot - \mathbf{v})| \, \mathbf{1}_{\leq\Delta} \right\rangle + \left\langle |\hat{\varphi}| \, \mathbf{1}_{>\Delta} \;,\; |\hat{\varphi}(\cdot - \mathbf{v})| \right\rangle \\
&\leq \|\hat{\varphi}\| \, \|\hat{\varphi}(\cdot - \mathbf{v}) \mathbf{1}_{\leq\Delta}\| \;+\; \|\hat{\varphi}\mathbf{1}_{>\Delta}\| \, \|\hat{\varphi}(\cdot - \mathbf{v})\|,
\end{aligned}$$

where in the last step we used Cauchy-Schwartz. Using the fact that norms and inner products are invariant to coordinate shifting, the above is at most

$$\begin{aligned}
&\|\hat{\varphi}\| \, \|\hat{\varphi}\mathbf{1}_{\leq\Delta}(\cdot + \mathbf{v})\| + \|\hat{\varphi}\mathbf{1}_{>\Delta}\| \, \|\hat{\varphi}\| \\
&\qquad = \|\hat{\varphi}\| \left(\sqrt{\int |\hat{\varphi}(\mathbf{x})|^2 \mathbf{1}_{\|\mathbf{x}+\mathbf{v}\|\leq\Delta} d\mathbf{x}} + \sqrt{\int |\hat{\varphi}(\mathbf{x})|^2 \mathbf{1}_{>\Delta}(\mathbf{x}) d\mathbf{x}} \right).
\end{aligned}$$

9. Strictly speaking, this function does not have a Fourier transform in the sense of Eq. (11), since the associated integrals do not converge. However, the function still has a well-defined Fourier transform in the more general sense of a generalized function or distribution (see Hunter and Nachtergaele 2001 for a survey). In the derivation below, we will simply rely on some standard formulas from the Fourier analysis literature, and refer to Hunter and Nachtergaele (2001) for their formal justifications.

By the triangle inequality and the assumption $\|\mathbf{v}\| = 2\Delta$, the event $\|\mathbf{x}+\mathbf{v}\| \le \Delta$ implies $\|\mathbf{x}\| \ge \Delta$. Therefore, the above can be upper bounded by

$$2\left\|\hat{\varphi}\right\| \sqrt{\int |\hat{\varphi}(\mathbf{x})|^2 \mathbf{1}_{\ge\Delta}(\mathbf{x}) d\mathbf{x}} \;=\; 2\left\|\hat{\varphi}\right\| \cdot \left\|\hat{\varphi}\cdot \mathbf{1}_{\ge\Delta}\right\|.$$

Since φ is Fourier-concentrated, this is at most $2\epsilon(\Delta)\left\|\hat{\varphi}\right\|^2 = 2\epsilon(\Delta)\left\|\varphi\right\|^2 = 2\epsilon(\Delta)$, where we use the isometry of the Fourier transform and the assumption that $\left\|\varphi\right\|^2 = \int \varphi^2(\mathbf{x})d\mathbf{x} = 1$. Plugging back the definition of Δ, the result follows. ■

Lemma 16 *It holds that*

$$\sum_{z_1 \ne z_2 \in \mathbb{Z}} |a_{z_1}| \cdot |a_{z_2}| \cdot \epsilon(r|z_1 - z_2|) \;\le\; 2\sum_{n=1}^{\infty} \epsilon(nr)$$

Proof For simplicity, define $\epsilon'(x) = \epsilon(x)$ for all $x > 0$, and $\epsilon(0) = 0$. Then the expression in the lemma equals

$$\begin{aligned}
&\sum_{z_1,z_2\in\mathbb{Z}} |a_{z_1}| \cdot |a_{z_2}| \cdot \epsilon'(|z_1 - z_2|r) \\
&= \sum_{z_1,z_2\in\mathbb{Z}} \left(|a_{z_1}|\sqrt{\epsilon'(|z_1 - z_2|r)}\right)\left(|a_{z_2}|\sqrt{\epsilon'(|z_1 - z_2|r)}\right) \\
&\le \sqrt{\sum_{z_1,z_2\in\mathbb{Z}} |a_{z_1}|^2 \epsilon'(|z_1 - z_2|r)} \sqrt{\sum_{z_1,z_2\in\mathbb{Z}} |a_{z_2}|^2 \epsilon'(|z_1 - z_2|r)} \\
&= \sum_{z_1,z_2\in\mathbb{Z}} |a_{z_1}|^2 \epsilon'(|z_1 - z_2|r)
\end{aligned}$$

where in the last step we used the fact that the two inner square roots are the same up to a different indexing. Recalling the definition of ϵ' and that $\sum_z |a_z|^2 \le 1$, the above is at most

$$\begin{aligned}
&\sum_{z_1\in\mathbb{Z}} |a_{z_1}|^2 \sum_{z_2\in\mathbb{Z}} \epsilon'(|z_1 - z_2|r) \;\le\; \sup_{z_1\in\mathbb{Z}} \sum_{z_2\in\mathbb{Z}} \epsilon'(|z_1 - z_2|r) \\
&= \left(\epsilon'(0) + 2\sum_{n=1}^{\infty} \epsilon'(nr)\right) \;=\; 2\sum_{n=1}^{\infty} \epsilon(nr)\,.
\end{aligned}$$

■

Lemma 17 *For any $g \in L^2(\mathbb{R}^d)$, if $d \ge 40$, and we sample $\mathbf{w}$ uniformly at random from $\{\mathbf{w} : \|\mathbf{w}\| = 2r\}$, it holds that*

$$\mathbb{E}\left[\left(\left\langle g, \widehat{\psi_{\mathbf{w}}\varphi}\right\rangle - a_0 \left\langle g, \hat{\varphi}\right\rangle\right)^2\right] \;\le\; 10\left\|g\right\|^2 \left(\exp(-d/20) + \sum_{n=1}^{\infty} \epsilon(nr)\right)$$

where a_0 is the coefficient from Lemma 14.

Proof By symmetry, given any function f of $\mathbf{w}$, the expectation $\mathbb{E}_{\mathbf{w}}[f(\mathbf{w})]$ (where $\mathbf{w}$ is uniform on a sphere) can be equivalently written as $\mathbb{E}_{\mathbf{w}\in\mathcal{W}}\mathbb{E}_U[f(U\mathbf{w})] = \mathbb{E}_U\mathbb{E}_{\mathbf{w}\in\mathcal{W}}[f(U\mathbf{w})]$, where U is a rotation matrix chosen uniformly at random (so that for any $\mathbf{w}$, $U\mathbf{w}$ is uniformly distributed on the sphere of radius $\|\mathbf{w}\|$), and $\mathbb{E}_{\mathbf{w}\in\mathcal{W}}$ refers to a uniform distribution of $\mathbf{w}$ over some finite set $\mathcal{W}$ of vectors of norm $2r$. In particular, we will choose $\mathcal{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_{\lceil \exp(d/20)\rceil}\}$ which satisfies the following:

$$\forall i \;\; \|\mathbf{w}_i\| = 2r \quad , \quad \forall i \neq j \;\; |\mathbf{w}_i^\top \mathbf{w}_j| < 2r^2. \tag{12}$$

The existence of such a set follows from standard concentration of measure arguments[10].

Thus, our goal is to bound $\mathbb{E}_U\mathbb{E}_{\mathbf{w}\in\mathcal{W}}\mathbb{E}\left[\left(\left\langle g, \widehat{\psi_{\mathbf{w}}\varphi}\right\rangle - a_0\left\langle g, \hat{\varphi}\right\rangle\right)^2\right]$. In fact, we will prove the bound stated in the lemma for any U, and will focus on $U = I$ without loss of generality (the argument for other U is exactly the same). First, by applying Lemma 14, we have

$$\begin{aligned}\mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left(\left\langle g, \widehat{\psi_{\mathbf{w}}\varphi}\right\rangle - a_0\left\langle g, \hat{\varphi}\right\rangle\right)^2\right] &= \mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left(\left\langle g, \sum_{z\in\mathbb{Z}} a_z\hat{\varphi}(\cdot - z\mathbf{w})\right\rangle - a_0\left\langle g, \hat{\varphi}\right\rangle\right)^2\right] \\ &= \mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left\langle g, \sum_{z\in\mathbb{Z}\setminus\{0\}} a_z\hat{\varphi}(\cdot - z\mathbf{w})\right\rangle^2\right]. \end{aligned}\tag{13}$$

For any $\mathbf{w}\in\mathcal{W}$, let

$$A_{\mathbf{w}} = \{\mathbf{x}\in\mathbb{R}^d \;:\; \exists z\in\mathbb{Z}\setminus\{0\} \; s.t. \; \|\mathbf{x} - z\mathbf{w}\| < r\}.$$

In words, each $A_{\mathbf{w}}$ corresponds to the union of open balls of radius r around $\pm\mathbf{w}, \pm 2\mathbf{w}, \pm 3\mathbf{w}\ldots$. An important property of these sets is that they are disjoint: $A_{\mathbf{w}}\cap A_{\mathbf{w}'} = \emptyset$ for any distinct $\mathbf{w}, \mathbf{w}'\in\mathcal{W}$. To see why, note that if there was some $\mathbf{x}$ in both of them, it would imply $\|\mathbf{x} - z_1\mathbf{w}\| < r$ and $\|\mathbf{x} - z_2\mathbf{w}'\| < r$ for some non-zero $z_1, z_2\in\mathbb{Z}$, hence $\|z_1\mathbf{w} - z_2\mathbf{w}'\| < 2r$ by the triangle inequality. Squaring both sides and performing some simple manipulations (using the facts that $\|\mathbf{w}\| = \|\mathbf{w}'\| = 2r$ and $|z_1|, |z_2| \geq 1$), we would get

$$2|z_1z_2|\cdot\left|\mathbf{w}^\top\mathbf{w}'\right| > 4r^2(z_1^2 + z_2^2 - 1) \geq 2r^2(z_1^2 + z_2^2) \quad\Rightarrow\quad \left|\mathbf{w}^\top\mathbf{w}'\right| \geq r^2\left(\left|\frac{z_1}{z_2}\right| + \left|\frac{z_2}{z_1}\right|\right) \geq 2r^2,$$

where we used the fact that $x + 1/x \geq 2$ for all $x > 0$. This contradicts the assumption on $\mathcal{W}$ (see Eq. 12), and establishes that $\{A_{\mathbf{w}}\}_{\mathbf{w}\in\mathcal{W}}$ are indeed disjoint sets.

We now continue by analyzing Eq. (13). Letting $\mathbf{1}_{A_{\mathbf{w}}}$ be the indicator function to the set $A_{\mathbf{w}}$, and $\mathbf{1}_{A_{\mathbf{w}}^C}$ be the indicator of its complement, and recalling that $(a+b)^2 \leq 2(a^2+b^2)$, we can upper bound Eq. (13) by

$$2\cdot\mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left\langle g, \mathbf{1}_{A_{\mathbf{w}}}\sum_{z\in\mathbb{Z}\setminus\{0\}} a_z\hat{\varphi}(\cdot - z\mathbf{w})\right\rangle^2\right] + 2\cdot\mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left\langle g, \mathbf{1}_{A_{\mathbf{w}}^C}\sum_{z\in\mathbb{Z}\setminus\{0\}} a_z\hat{\varphi}(\cdot - z\mathbf{w})\right\rangle^2\right]. \tag{14}$$

10. For any two vectors $\mathbf{w}, \mathbf{w}'$ picked uniformly at random from a ball of radius $2r$, $\Pr(|\mathbf{w}^\top\mathbf{w}'| \geq 2r^2) = 2\Pr\left(\left(\frac{1}{2r}\mathbf{w}\right)^\top\left(\frac{1}{2r}\mathbf{w}'\right) > \frac{1}{2}\right) \leq 2\exp\left(-\frac{d}{8}\right)$ (see Boucheron et al. 2013, Section 7.2), so by a union bound, the probability that Eq. (12) is not satisfied is at most $2\lceil\exp(d/20)\rceil^2\exp(-d/8)$. This can be verified to be strictly less than 1 if $d \geq 40$, hence such a set exists under the lemma's conditions.

We consider each expectation separately. Starting with the first one, we have

$$\begin{aligned}\mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left\langle g, \mathbf{1}_{A_{\mathbf{w}}}\sum_{z\in\mathbb{Z}\setminus\{0\}} a_z\hat{\varphi}(\cdot - z\mathbf{w})\right\rangle^2\right] &= \mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left\langle g, \mathbf{1}_{A_{\mathbf{w}}}\left(\widehat{\psi_{\mathbf{w}}\varphi} - a_0\hat{\varphi}\right)\right\rangle^2\right] \\ &= \mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left\langle \mathbf{1}_{A_{\mathbf{w}}}g, \widehat{\psi_{\mathbf{w}}\varphi} - a_0\hat{\varphi}\right\rangle^2\right] \le \mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\|\mathbf{1}_{A_{\mathbf{w}}}g\|^2\left\|\widehat{\psi_{\mathbf{w}}\varphi} - a_0\hat{\varphi}\right\|^2\right] \\ &\le 2\cdot\mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\|\mathbf{1}_{A_{\mathbf{w}}}g\|^2\left(\left\|\widehat{\psi_{\mathbf{w}}\varphi}\right\|^2 + \|a_0\hat{\varphi}\|^2\right)\right] \\ &= 2\cdot\mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\|\mathbf{1}_{A_{\mathbf{w}}}g\|^2\left(\|\psi_{\mathbf{w}}\varphi\|^2 + |a_0|^2\cdot\|\hat{\varphi}\|^2\right)\right].\end{aligned}$$

Since we have $\|\hat{\varphi}\| = \|\varphi\| = 1$, $|a_0|^2 \le \sum_z |a_z|^2 \le 1$, and $\|\psi_{\mathbf{w}}\varphi\|^2 = \int \psi_{\mathbf{w}}^2(\mathbf{x})\varphi^2(\mathbf{x})d\mathbf{x} \le \int \varphi^2(\mathbf{x})d\mathbf{x} = 1$, the above is at most

$$\begin{aligned}4\cdot\mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\|\mathbf{1}_{A_{\mathbf{w}}}g\|^2\right] &\le \frac{4}{|\mathcal{W}|}\sum_{\mathbf{w}\in\mathcal{W}}\int 1_{A_{\mathbf{w}}}(\mathbf{x})|g(\mathbf{x})|^2 d\mathbf{x} \\ &= \frac{4}{|\mathcal{W}|}\int\left(\sum_{\mathbf{w}\in\mathcal{W}} 1_{A_{\mathbf{w}}}(\mathbf{x})\right)|g(\mathbf{x})|^2 d\mathbf{x}.\end{aligned}$$

Since $A_{\mathbf{w}}$ are disjoint sets, $\sum_{\mathbf{w}\in\mathcal{W}}\mathbf{1}_{A_{\mathbf{w}}}(\mathbf{x}) \le 1$ for any $\mathbf{x}$, so the above is at most

$$\frac{4}{|\mathcal{W}|}\int |g(\mathbf{x})|^2 d\mathbf{x} \le 4\exp(-d/20)\|g\|^2. \tag{15}$$

We now turn to analyze the second expectation in Eq. (14), namely

$$\mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left\langle g, \mathbf{1}_{A_{\mathbf{w}}^C}\sum_{z\in\mathbb{Z}\setminus\{0\}} a_z\hat{\varphi}(\cdot - z\mathbf{w})\right\rangle^2\right].$$

We will upper bound the expression deterministically for any $\mathbf{w}$, so we may drop the expectation. Applying Cauchy-Schwartz, it is at most

$$\|g\|^2\cdot\left\|\mathbf{1}_{A_{\mathbf{w}}^C}\sum_{z\in\mathbb{Z}\setminus\{0\}} a_z\hat{\varphi}(\cdot - z\mathbf{w})\right\|^2 = \|g\|^2\left(\sum_{z_1,z_2\in\mathbb{Z}\setminus\{0\}} a_{z_1}\overline{a_{z_2}}\left\langle \mathbf{1}_{A_{\mathbf{w}}^C}\hat{\varphi}(\cdot - z_1\mathbf{w}), \hat{\varphi}(\cdot - z_2\mathbf{w})\right\rangle\right). \tag{16}$$

We now divide the terms in the sum above to two cases:

- If $z_1 = z_2$, then $\left\langle \mathbf{1}_{A_{\mathbf{w}}^C}\hat{\varphi}(\cdot - z_1\mathbf{w}), \hat{\varphi}(\cdot - z_2\mathbf{w})\right\rangle$ equals

  $$\int \mathbf{1}_{A_{\mathbf{w}}^C}(\mathbf{x})|\hat{\varphi}(\mathbf{x} - z_1\mathbf{w})|^2 d\mathbf{x}. = \int \mathbf{1}_{A_{\mathbf{w}}^C}(\mathbf{x} + z_1\mathbf{w})|\hat{\varphi}(\mathbf{x})|^2 d\mathbf{x},$$

  and by definition of $A_{\mathbf{w}}^C$ and the assumption $z_1 \ne 0$, we have $\mathbf{1}_{A_{\mathbf{w}}^C}(\mathbf{x} + z_1\mathbf{w}) = 1$ only if $\|\mathbf{x}\| \ge r$. Therefore, as φ is Fourier-concentrated, the above is at most

  $$\int_{\mathbf{x}:\|\mathbf{x}\|\ge r} |\hat{\varphi}(\mathbf{x})|^2 d\mathbf{x} \le \epsilon^2(r)\cdot\|\hat{\varphi}\|^2 = \epsilon^2(r)\cdot\|\varphi\|^2 = \epsilon^2(r).$$

- If $z_1 \neq z_2$, then by Lemma 15,

$$\left\langle \mathbf{1}_{A^C_{\mathbf{w}}} \hat{\varphi}(\cdot - z_1\mathbf{w}), \hat{\varphi}(\cdot - z_2\mathbf{w}) \right\rangle \;\leq\; \langle |\hat{\varphi}(\cdot - z_1\mathbf{w})|, |\hat{\varphi}(\cdot - z_2\mathbf{w})| \rangle \;\leq\; 2\epsilon(|z_1 - z_2|r).$$

Plugging these two cases back into Eq. (16), we get the upper bound

$$\|g\|^2 \left(\sum_{z \in \mathbb{Z}\setminus\{0\}} |a_z|^2 \epsilon^2(r) + 2 \sum_{z_1 \neq z_2 \in \mathbb{Z}} |a_{z_1}| \cdot |a_{z_2}| \cdot \epsilon(|z_1 - z_2|r) \right).$$

Noting that $\sum_z |a_z|^2 \leq 1$, and applying Lemma 16, the above is at most

$$\|g\|^2 \left(\epsilon^2(r) + 4 \sum_{n=1}^{\infty} \epsilon(nr) \right) \;\leq\; 5 \|g\|^2 \sum_{n=1}^{\infty} \epsilon(nr),$$

where we used the fact that $\epsilon^2(r) \leq \epsilon(r) \leq \sum_{n=1}^{\infty} \epsilon(nr)$. Recalling this is an upper bound on the second expectation in Eq. (14), and that the first expectation is upper bounded as in Eq. (15), we get that Eq. (14) (and hence the expression in the lemma statement) is at most

$$10 \|g\|^2 \left(\exp(-d/20) + \sum_{n=1}^{\infty} \epsilon(nr) \right)$$

as required. ■

With these lemmas in hand, we can now turn to prove the theorem. We have that

$$\mathrm{Var}_{\mathbf{w}^\star} \left[\nabla F_{\mathbf{w}^\star}(\mathbf{v}) \right] \;=\; \mathbb{E}_{\mathbf{w}^\star} \left\| \nabla F_{\mathbf{w}^\star}(\mathbf{v}) - \mathbb{E}_{\mathbf{w}^\star}[\nabla F_{\mathbf{w}^\star}(\mathbf{v})] \right\|^2 \;\leq\; \mathbb{E}_{\mathbf{w}^\star} \left\| \nabla F_{\mathbf{w}^\star}(\mathbf{v}) - \mathbf{p} \right\|^2$$

for any vector $\mathbf{p}$ which is not dependent of $\mathbf{w}^\star$ (this $\mathbf{p}$ will be determined later). Recalling the definition of the objective function F, and letting $\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots) = \frac{\partial}{\partial \mathbf{v}} f(\mathbf{v}, \mathbf{x})$, the above equals

$$\begin{aligned}
&\mathbb{E}_{\mathbf{w}^\star} \left\| \mathbb{E}_{\mathbf{x} \sim \varphi^2} \left[\left(f(\mathbf{v}, \mathbf{x}) - \psi(\mathbf{w}^{\star\top} \mathbf{x}) \right) \mathbf{g}(\mathbf{x}) \right] - \mathbf{p} \right\|^2 \\
&\quad = \sum_i \mathbb{E}_{\mathbf{w}^\star} \left(\mathbb{E}_{\mathbf{x} \sim \varphi^2} \left[f(\mathbf{v}, \mathbf{x}) g_i(\mathbf{x}) - \psi(\mathbf{w}^{\star\top} \mathbf{x}) g_i(\mathbf{x}) \right] - p_i \right)^2 \\
&\quad = \sum_i \mathbb{E}_{\mathbf{w}^\star} \left(\langle \varphi g_i, \varphi f(\mathbf{v}, \cdot) \rangle - \langle \varphi g_i, \varphi \psi_{\mathbf{w}^\star} \rangle - p_i \right)^2
\end{aligned}$$

Let us now choose $\mathbf{p}$ so that $p_i = \langle \varphi g_i, \varphi f(\mathbf{v}, \cdot) \rangle - \langle \varphi g_i, a_0 \varphi \rangle$ (note that this choice is indeed independent of $\mathbf{w}^\star$). Plugging back and applying Lemma 17 (using the L^2 function $\widehat{\varphi g_i}$ for each i), we get

$$\begin{aligned}
\sum_i \mathbb{E}_{\mathbf{w}^\star} \left(\langle \varphi g_i, \varphi \psi_{\mathbf{w}^\star} \rangle - \langle \varphi g_i, a_0 \varphi \rangle \right)^2 \;&=\; \sum_i \mathbb{E}_{\mathbf{w}^\star} \left(\left\langle \widehat{\varphi g_i}, \widehat{\varphi \psi_{\mathbf{w}^\star}} \right\rangle - \langle \widehat{\varphi g_i}, a_0 \widehat{\varphi} \rangle \right)^2 \\
&\leq\; 10 \sum_i \|\widehat{\varphi g_i}\|^2 \left(\exp(-d/20) + \sum_{n=1}^{\infty} \epsilon(nr) \right),
\end{aligned}$$

and since

$$\sum_{i=1}^{d} \|\varphi g_i\|^2 \;=\; \sum_{i=1}^{d} \int g_i^2(\mathbf{x})\varphi^2(\mathbf{x})d\mathbf{x} \;=\; \int \|\mathbf{g}(\mathbf{x})\|^2 \varphi^2(\mathbf{x}) \;=\; \mathbb{E}_{\mathbf{x}\sim\varphi^2} \|\mathbf{g}(\mathbf{x})\|^2 \;\leq\; G_{\mathbf{v}}^2,$$

the theorem follows.

5.4 Proof of Thm. 10

For any $\mathbf{w}^*$, we define the oracle $O_{F_{\mathbf{w}^*},\epsilon}$ as follows:

$$O_{F_{\mathbf{w}^*},\epsilon}(\mathbf{v}) = \begin{cases} \mathbb{E}_{\mathbf{w}^\star}[\nabla F_{\mathbf{w}^\star}(\mathbf{v})] & \text{if } |\nabla F_{\mathbf{w}^\star}(\mathbf{v}) - \mathbb{E}_{\mathbf{w}^\star}[\nabla F_{\mathbf{w}^\star}(\mathbf{v})]| \leq \varepsilon \\ \nabla F_{\mathbf{w}^\star}(\mathbf{v}) & \text{otherwise} \end{cases},$$

where the expectation is with respect to a uniform choice of $\mathbf{w}^*$ over vectors of norm $2r$. This is an approximate gradient oracle by definition.

Below, we will prove the theorem statement assuming that the algorithm A is deterministic: Namely, that there is some event holding with probability at least $1-p$ (over the choice of $\mathbf{w}^*$), such that the algorithm's output is some fixed vector $\bar{\mathbf{v}}$ independent of $\mathbf{w}^*$. This can be extended to randomized A, by considering any possible realization of A's random coin flips: Formally, let A_c be the deterministic algorithm, derived from A by fixing its random coin flips to some fixed sequence c. The deterministic proof implies that there is some event E_c and a fixed $\bar{\mathbf{v}}_c$ (independent of $\mathbf{w}^*$) such that $\Pr_{\mathbf{w}^*}(E_c) \geq 1-p$ and $\Pr_{\mathbf{w}^*}(A_C(F_{\mathbf{w}^*}) = \bar{\mathbf{v}}_c | E_c) = 1$ (where we use $A_c(F)$ as shorthand for A_C given the function F). But now, consider the joint probability space over $\mathbf{w}^*$ and random coin flips C, and define the event E as

$$\bigvee_c (C = c \;\wedge\; E_c)$$

(namely, that the event E_c occured for the corresponding realization c of the coin flips). Since this is a disjunction of disjoint events, we have that

$$\Pr(E) = \sum_c \Pr(C = c \wedge E_c) = \sum_c \Pr(C=c)\Pr(E_c|C=c) \geq \sum_c \Pr(C=c)\cdot(1-p) = 1-p,$$

and moreover, if E occurs, then the distribution of $A_C(F_{\mathbf{w}^*})$ is identical to $\bar{\mathbf{v}}_C$ (depending only on the random coin flips C, but not on $\mathbf{w}^*$), which is exactly what the theorem states.

We now return to the proof, assuming A is deterministic. It is enough to show that with probability at least $1-p$, the oracle will only return responses of the form $\mathbb{E}_{\mathbf{w}^\star}[\nabla F_{\mathbf{w}^\star}(\mathbf{v})]$, which is clearly independent of $\mathbf{w}^\star$. Since the algorithm's output can depend on $\mathbf{w}^\star$ only through the oracle responses, this will prove the required result.

The (deterministic) algorithm's first point $\mathbf{v}_1$ is fixed before receiving any information from the oracle, and is therefore independent of $\mathbf{w}^\star$. By Thm. 7, we have that $\mathrm{Var}_{\mathbf{w}^\star}(\nabla F_{\mathbf{w}^\star}(\mathbf{v}_1)) \leq \varepsilon^3$, which by Chebyshev's inequality, implies that

$$\Pr\left(|\nabla F_{\mathbf{w}^\star}(\mathbf{v}_1) - \mathbb{E}_{\mathbf{w}^\star}[\nabla F_{\mathbf{w}^\star}(\mathbf{v}_1)]| > \varepsilon\right) \;\leq\; \varepsilon,$$

where the probability is over the choice of $\mathbf{w}^\star$. Assuming the event above does not occur, the oracle returns $\mathbb{E}_{\mathbf{w}^\star}[\nabla F_{\mathbf{w}^\star}(\mathbf{v})]$, which does not depend on the actual choice of $\mathbf{w}^\star$. This

means that the next point $\mathbf{v}_2$ chosen by the algorithm is again sampled from some fixed distribution $\mathcal{D}_2$, which might depend on the algorithm's internal randomness, but not on $\mathbf{w}^\star$. Again by Thm. 7 and Chebyshev's inequality,

$$\Pr\left(|\nabla F_{\mathbf{w}^\star}(\mathbf{v}_2) - \mathbb{E}_{\mathbf{w}^\star}[\nabla F_{\mathbf{w}^\star}(\mathbf{v}_2)]| > \varepsilon\right) \;\leq\; \varepsilon.$$

Repeating this argument and applying a union bound, it follows that as long as the number of iterations T satisfies $T\varepsilon \leq p$ (or equivalently $T \leq p/\varepsilon$), the oracle reveals no information whatsoever on the choice of $\mathbf{w}^\star$ all point chosen by the algorithm (and hence also its output) are independent of $\mathbf{w}^\star$ as required.

Acknowledgments

This research is supported in part by an FP7 Marie Curie CIG grant, Israel Science Foundation grant 425/13, and the Intel ICRI-CI Institute.

References

Alexandr Andoni, Rina Panigrahy, Gregory Valiant, and Li Zhang. Learning polynomials with neural networks. In *ICML*, 2014.

Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. In *ICML*, 2014.

Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning dnf and characterizing statistical query learning using fourier analysis. In *STOC*, 1994.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *AISTATS*, 2015.

Amit Daniely and Shai Shalev-Shwartz. Complexity theoretic limitations on learning dnf's. In *COLT*, 2016.

Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. *arXiv preprint arXiv:1602.05897*, 2016.

David Donoho and Iain Johnstone. Projection-based approximation and a duality with kernel methods. *The Annals of Statistics*, pages 58–106, 1989.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul): 2121–2159, 2011.

Vitaly Feldman, Cristobal Guzman, and Santosh Vempala. Statistical query algorithms for stochastic convex optimization. *arXiv preprint arXiv:1512.09170*, 2015.

Elad Hazan, Kfir Levy, and Shai Shalev-Shwartz. Beyond convexity: Stochastic quasi-convex optimization. In *NIPS*, 2015.

John K. Hunter and Bruno Nachtergaele. *Applied analysis*. World Scientific Publishing, 2001.

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

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.

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

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

Adam R. Klivans and Pravesh Kothari. Embedding hard learning problems into gaussian space. In *APPROX/RANDOM*, 2014.

Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. In *NIPS*, 2014.

Peter McCullagh and John Nelder. *Generalized linear models*. CRC press, 1989.

Yurii Nesterov. Minimization methods for nonsmooth convex and quasiconvex functions. *Matekon*, 29:519–531, 1984.

Itay Safran and Ohad Shamir. On the quality of the initial basin in overspecified neural networks. In *ICML*, 2016.

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

Daniel Soudry and Yair Carmon. No bad local minima: Data independent training error guarantees for multilayer neural networks. *arXiv preprint arXiv:1605.08361*, 2016.

Yuchen Zhang, Jason Lee, Martin Wainwright, and Michael Jordan. Learning halfspaces and neural networks with random initialization. *arXiv preprint arXiv:1511.07948*, 2015.