ON ROBUST CONCEPTS AND SMALL NEURAL NETS

Amit Deshpande
Microsoft Research, Vigyan, 9 Lavelle Road, Bengaluru 560001, India
amitdesh@microsoft.com

Sushrut Karmalkar[*]
Department of Computer Science, The University of Texas at Austin,
2317 Speedway, Stop D9500 Austin, TX 78712, USA
sushrutk@cs.utexas.edu

ABSTRACT

The universal approximation theorem for neural networks says that any reasonable function is well-approximated by a two-layer neural network with sigmoid gates but it does not provide good bounds on the number of hidden-layer nodes or the weights. However, *robust* concepts often have *small* neural networks in practice. We show an efficient analog of the universal approximation theorem on the boolean hypercube in this context.

We prove that any *noise-stable* boolean function on n boolean-valued input variables can be well-approximated by a two-layer linear threshold circuit with a *small* number of hidden-layer nodes and *small* weights, that depend only on the noise-stability and approximation parameters, and are *independent of* n. We also give a polynomial time learning algorithm that outputs a *small* two-layer linear threshold circuit that approximates such a given function. We also show weaker generalizations of this to noise-stable polynomial threshold functions and noise-stable boolean functions in general.

1 INTRODUCTION

The universal approximation theorem of Hornik et al. (1989) and Cybenko (1992) provides a foundation to the mathematical theory of artificial neural networks. It states that any continuous function on a compact subset of the Euclidean space can be approximated arbitrarily well by a feed-forward artificial neural network with only one hidden layer containing finitely many neurons, under mild assumptions on the activation function. In such neural networks, each node applies an activation function to a weighted linear combination of its inputs, and the above theorem holds true for many different choices of activation functions as shown by Hornik (1991). However, the universal approximation theorem and its quantitative improvements by Barron (1993) and others have certain limitations, namely, they do not provide reasonable, practical bounds or efficient learning algorithms for the parameters of these neural networks, that is, the number of neurons in the hidden layer and the size of weights used in the linear combinations. For a detailed survey of these results in approximation theory, we point the reader to Pinkus (1999).

In practice, we notice that even moderate-sized neural networks can be trained to learn various natural concepts in computer vision tasks, and the typical rules of thumb followed for their model and size selection are usually guided by the domain knowledge, the learning algorithm, and the available computational resources more than any theoretical bounds; see Simard et al. (2003). The known theoretical bounds are either based on the Network Information Criterion (NIC) by Amari (1998), which is a generalization of Akaike Information Criterion (AIC) by Akaike (1974) used in statistical inference, or based on the Vapnik-Chervonenkis dimension; see Baum & Haussler (1989), Bartlett (1993), Maass (1995), Karpinski & Macintyre (1997). These bounds do not adequately explain the observed efficiency of learning many natural concepts in practice.

[*] This work was done during an internship at Microsoft Research India, when the author was a student at Chennai Mathematical Institute, H1, SIPCOT IT Park, Siruseri, Chennai 603103, India

Most natural concepts are often based on a small number of relevant attributes or features, and can be learnt efficiently once we implicitly map our input to the correct attribute space and focus on these relevant attributes or features. Moreover, most natural concepts are also robust, that is, their positive and negative examples are reasonably unambiguous and far from each other. Thus, an important theoretical question is to understand the underlying cognitive process, find a reasonably close and accurate model for it, and answer why certain models like artificial neural networks can mimic this cognitive process in practice.

The implicit mapping of our input coordinates to the space of attributes is formalized by the kernel method in machine learning; see Hofmann et al. (2008). Attribute-efficient learning proposed by Valiant (2000) and Littlestone (1988) captures the ease of learning via improved VC-dimension bounds that depend only a small number of relevant attributes. Robust concepts are often defined using large-margin classifiers studied in the context of Support Vector Machines; see Cortes & Vapnik (1995). We use a different notion of robustness suited to the boolean hypercube known as noise-stability. Due to known results from Fourier analysis over the boolean hypercube, noise-stability also implies closeness to a function that depends only on a small number of attributes.

Since the universal approximation theorem gives a depth-2 neural network with only one hidden layer, the effect of depth on the power of neural networks has attracted considerable interest in approximation theory as well as boolean circuit complexity; see de Villiers & Barnard (1993) and Siu et al. (1995). Note that on the boolean hypercube, depth-d circuits with sigmoid gates and linear threshold gates are essentially equivalent. An important result relevant to our paper is due to a long line of work including Goldmann et al. (1992), Goldmann & Karpinski (1998), and Hofmeister (1996) which proved that any depth-d linear threshold circuit with polynomially (in the number n of input variables) many nodes but arbitrary weights can be efficiently simulated by a depth-$(d+1)$ linear threshold circuit with polynomially many nodes and polynomially bounded integer weights.

2 Our Results

We work with linear threshold circuits with boolean inputs and outputs, which are discrete analogs of the neural networks with real-valued inputs and continuous activation functions. They are also known as multi-layer perceptrons as in Minsky & Papert (1987), which are simply feed-forward neural networks where each node computes a weighted linear combination of its inputs and applies a threshold function for activation. As mentioned above, the notion of robustness we use is noise-stability or low noise-sensitivity. The noise sensitivity of a boolean function is simply the fraction of inputs whose output changes, if we change each coordinate of the input independently with a small probability, say some $\epsilon > 0$.

As a warm-up, we show that if a boolean function defined on the boolean hypercube $\{-1,1\}^n$ is noise-stable, that is, if it has low noise-sensitivity, then it can be approximated by a depth-2 linear threshold circuit (that is, with one hidden layer), that depends only on constantly many variables in the input, and its number of hidden nodes and the weights are also constants, all independent of n. Here we quantify approximation or closeness based on the fraction of inputs where two functions differ. This result may be folklore although we are not aware of any reference.

Theorem 1. *Any $f : \{-1,1\}^n \to \{-1,1\}$ that has small noise-sensitivity for ϵ-perturbations, that is, $\mathrm{NS}_\epsilon(f) = O\left(\delta\sqrt{\epsilon}\right)$, is δ-close to a depth-2 linear threshold circuit that depends only on $O(1)$ variables of the input with $O(1)$ hidden nodes and $O(1)$ weights, where the constants $O(1)$ depend on ϵ and δ but are independent of n.*

When the given function is actually a linear threshold function, that is, when it represents a halfspace, we can improve the above theorem with constants $O(1)$ that are polynomial in $1/\epsilon$ and $1/\delta$, and thus, give an efficient analog of the universal approximation theorem for neural networks over the boolean hypercube. Note that this is consistent with the intuition that better noise-stable concepts can be approximated by smaller neural networks. It also shows that a given concept may be linearly separable in a high n-dimensional kernel space but its approximation by neural networks only depends on an inherent parameter like robustness or noise-sensitivity, independent of n.

Theorem 2. *Any linear threshold function $f : \{-1,1\}^n \to \{-1,1\}$ that has small noise-sensitivity for ϵ-perturbations, that is, $\mathrm{NS}_\epsilon(f) = O\left(\delta^3\sqrt{\epsilon}\right)$, is δ-close to a depth-2 linear threshold circuit* *that depends only on $O(1)$ variables of the input with $O(1)$ hidden nodes and $O(1)$ integer weights, where the constants are poly$(1/\epsilon, 1/\delta)$ but independent of n.*

Equipped with this, we show the following implication for learning. Given oracle access to such a linear threshold function f of low noise-sensitivity, we can learn a depth-2 linear threshold circuit that approximates f well, in polynomial time.

Theorem 3. *Let $f : \{-1,1\}^n \to \{-1,1\}$ be any linear threshold function with small noise-sensitivity for ϵ-perturbations, that is, $\mathrm{NS}_\epsilon(f) = O\left(\delta^3\sqrt{\epsilon}\right)$. Then we can learn a depth-2 linear threshold circuit on k variables that is $\exp\left(-\Omega(\sqrt[3]{\log(1/\delta)})\right)$-close to f with probability $1-\gamma$, in time $n^k \cdot p\left(1/\epsilon, 1/\delta, 1/\gamma\right)$, where p is polynomial in $1/\epsilon$, exponential in polylog $(1/\delta)$, and logarithmic in $1/\gamma$, and $k = O\left(1/\epsilon^2 \cdot \log(1/\epsilon) \cdot \log(1/\delta)\right)$. Moreover, the size and integer weights of the depth-2 linear threshold circuits are polynomially bounded in $1/\epsilon$ and $1/\delta$.*

We would also like to note that it is possible to extend our result for halfspaces to polynomial threshold functions. This uses the facts that any degree-d polynomial threshold function ϵ-close to a J-junta, is close to junta that is a polynomial threshold function of degree at most d, and that the machinery from De et al. (2014) extends to small weight polynomial threshold functions as well.

In a recent paper, Feldman & Vondrak (2013) have shown that sub-modular functions are ϵ close to $O\left(1/\epsilon^2 \cdot \log\left(1/\epsilon\right)\right)$-juntas. Note that this tells us that we can ϵ-approximate submodular functions by polynomials of degree $O\left(1/\epsilon^2 \cdot \log\left(1/\epsilon\right)\right)$. This means we can approximate submodular functions by depth-3 neural networks with linear threshold gates everywhere except for the top gate.

2.1 Obstacles to improvements

We now discuss some obstacles to possible improvements of our results.

The n^k running time is needed to identify the specific set of $O\left(1/\epsilon^2 \cdot \log(1/\epsilon) \cdot \log(1/\delta)\right)$ relevant coordinates. This $n^{O(k)}$ factor is unavoidable while learning k-juntas, and a candidate hard case is presented in Blum et al. (1994). Only recently Valiant (2015) gave an improved algorithm to learn k-juntas with noise rate η that runs in time less than $n^{0.8k} \cdot \text{poly}\left(2^k, 1/(1-2\eta)\right)$.

Weak, proper, agnostic learning of halfspaces under non-uniform distributions is NP-hard as shown by Guruswami & Raghavendra (2006), and extended to improper learning by Daniely et al. (2013) and Daniely (2015). Daniely's result rules out efficient, constant factor approximation for even improper learning of halfspaces using any hypothesis class on the boolean hypercube under non-uniform distributions[1]. However, Daniely (2014) can get around this by giving a PTAS for improper learning of halfspaces on the unit sphere under uniform distribution. Our result can be seen as another way to circumvent the hardness results. We learn noise-stable halfspaces on the boolean hypercube under uniform distribution, by giving an efficient, agnostic-type learning algorithm where the output hypothesis is a depth-2 neural network. This is arguably more natural than other improper learning results for halfspaces via low-degree polynomials.

Not having an efficient version of Bourgain's theorem for arbitrary noise-stable boolean functions, where the number of junta variables is polynomial in the noise-sensitivity parameters is another obstacle to efficient generalizations of our result. Note that the proof of this for noise-stable halfspaces does not generalize to higher depth linear threshold circuits. Another approach is to approximate any noise-stable function first using a halfspace and then by a depth-2 linear threshold circuit, but this has been ruled out by Mossel & Neeman (2016) with an example of a noise-stable function that is far from any halfspace.

We now give a brief outline of the proofs of the above theorems. Bourgain (2002) proved that any function with small noise-sensitivity can be approximated by another function that is a junta, which means that it depends on very few coordinates. In Theorem 1, we show that such a function can also be represented by a small depth-2 linear threshold circuit with small size and small integer weights. Moreover, any linear threshold function that is close to a junta is actually close to a linear threshold

[1] Results in Daniely et al. (2013) are under certain assumptions that are refuted in Allen et al. (2015). However, Daniely (2015) recovers a slightly weaker but very similar result for halfspaces under different assumptions

function defined over those junta coordinates. Thus, we can approximate the given noise-stable function by a linear threshold function on a small number of inputs, however, its weights may be large. Therefore, we use the size-depth-weight trade-off from Goldmann et al. (1992) to simulate this linear threshold function by a depth-2 linear threshold circuit with small size as well as small weights in Theorem 2. We also use a recent improvement over Bourgain's theorem by Diakonikolas et al. (2014) to get bounds polynomial in the noise-stability parameters. Theorem 3 follows by combining a result of De et al. (2014) on agnostic-type learning by a linear threshold function with a constructive, efficient simulation of the Goldmann et al. (1992) result by Goldmann & Karpinski (1998).

3 Related Work

Motivated by the recent advances in neural networks, there have been various attempts to build a theory to understand why neural networks can efficiently simulate many natural concepts and why their models and parameters can be learnt efficiently, for example, Andoni et al. (2014) and Arora et al. (2014). Our objective is to show efficient analogs of the universal approximation theorem for neural networks, a question that has been studied in approximation theory as well as boolean circuit complexity. We combine the size-depth-weight trade-off results from about two decades ago such as Goldmann et al. (1992) and Goldmann & Karpinski (1998) with more recent work on the Fourier analysis of boolean functions and its corollaries in learning. Also note that There are known NP-hardness results for learning halfspaces by Guruswami & Raghavendra (2009) and for approximately learning depth-2 threshold circuits by Bartlett & Ben-David (2002). However, these are for arbitrary threshold circuits. As we will show, the noise-stability constraint allows us to get a polynomial time algorithm to learn a depth-2 threshold circuit approximating the original function.

The low effective-dimension of hyperparameters has been observed and exploited to learn using neural networks in practice by Bergstra & Bengio (2012). We propose noise-stability as an approach to study this theoretically.

Arriaga & Vempala (2006) showed that robust or large-margin halfspaces in $\mathbb{R}^n$ can be learnt efficiently using random projections. Their learning algorithm outputs a depth-2 neural network with different activation functions in different layers. We define robustness using noise-stability instead, and show that better noise-stability reduces learning complexity. Our results also generalize to polynomial threshold functions, that is, a noise-stable polynomial threshold function (PTF) can be represented by a small, depth-2 neural network.

4 Preliminaries

Here we give a compilation of definitions and known results that we will use to prove Theorems 1, 2, and 3. Noise-stable boolean functions have low noise-sensitivity. Noise-sensitivity of a boolean function, with respect to ϵ-perturbations, is defined as the fraction of inputs whose output changes, when we change each bit of the input independently with a small probability ϵ.

Definition 1. *The noise sensitivity of a boolean function $f : \{-1,1\}^n \rightarrow \{-1,1\}$ at a given noise rate $\epsilon > 0$ is defined as*

$$\mathrm{NS}_\epsilon(f) = \mathit{Prob}_{x,y}\left(f(x) \neq f(y)\right),$$

where x is uniformly distributed in $\{-1,1\}^n$, and y is obtained from x by flipping each bit of x independently with probability ϵ.

A theorem of Bourgain (2002) states that boolean functions with small noise-sensitivity are close to juntas, which are boolean functions that depend on very few coordinates. Note that the number of these relevant coordinates is independent of n.

Lemma 1. *Any $f : \{-1,1\}^n \rightarrow \{-1,1\}$ that satisfies $\mathrm{NS}_\epsilon(f) = O\left(\delta\sqrt{\epsilon}\right)$ is δ-close to a k-junta, where*

$$k = \left(\frac{1}{\delta\epsilon}\right)^{O(1/\epsilon)}.$$

Here, δ-closeness means agreement on $1-\delta$ fraction of the inputs.

Note that the $\sqrt{\epsilon}$ in the bound has a special significance for linear threshold functions, as we explain below.

Definition 2. *A linear threshold function $f : \{-1, 1\}^n \to \{-1, 1\}$ is defined as*

$$f(x) = \operatorname{sgn}\left(\textstyle\sum_{i=1}^{n} w_i x_i - w_0\right),$$

for some weights $w_0, w_1, w_2, \ldots, w_n \in \mathbb{R}$.

A theorem of Peres (2004) states that the noise sensitivity of any linear threshold function at noise rate ϵ is at most $2\sqrt{\epsilon}$.

Lemma 2. *Any linear threshold function $f : \{-1, 1\}^n \to \{-1, 1\}$ satisfies $\mathrm{NS}_{\epsilon}(f) \leq 2\sqrt{\epsilon}$.*

The bounds in Proposition 1 can be improved when f is a linear threshold function as shown by the result of Diakonikolas et al. (2014) mentioned below. Thus, a noise-stable linear threshold function is close to a k-junta, where k is polynomial dependent on the noise and approximation parameters, but is independent of n.

Lemma 3. *Any linear threshold function $f : \{-1, 1\}^n \to \{-1, 1\}$ that satisfies $\mathrm{NS}_{\epsilon}(f) = O\left(\delta^{(2-\epsilon)/(1-\epsilon)}\sqrt{\epsilon}\right)$, for some $0 < \epsilon < 1/2$, is δ-close to a k-junta, where*

$$k = O\left(\frac{1}{\epsilon^2}\,\log\left(\frac{1}{\epsilon}\right)\,\log\left(\frac{1}{\delta}\right)\right).$$

Remark: *For convenience, we use $\mathrm{NS}_{\epsilon}(f) = O\left(\delta^3\sqrt{\epsilon}\right)$ in our assumption whenever using the above theorem.*

The following lemma from O'Donnell & Servedio (2011) ties it up nicely to say that if any linear threshold function is close to a junta, then it must be close to a linear threshold function defined over those junta coordinates.

Lemma 4. *If a linear threshold function $f : \{-1, 1\}^n \to \{-1, 1\}$ is δ-close to a junta over a subset $J \subseteq [n]$ of coordinates, then f is δ-close to a linear threshold function defined over that subset $J \subseteq [n]$ of coordinates.*

Linear threshold circuits where each gate computes a linear threshold function forms an important class in circuit complexity. We borrow the standard definitions and notation from Siu et al. (1995) and Goldmann et al. (1992).

Definition 3. *LT_d is defined as the class of linear threshold circuits of depth d on n inputs with the number of nodes polynomial in n but arbitrary weights inside the linear threshold functions. $\widehat{LT}_d$ is defined as the class of linear threshold circuit of depth d on n inputs with both the number of nodes and weights inside the linear threshold functions polynomially bounded in n.*

The size-depth-weight trade-offs for linear threshold circuits have been studied in circuit complexity with keen interest, and a long line of work culminated in the following result by Goldmann et al. (1992). Here, the weight bounds are bounds on the ratio of the maximum and the minimum weights, when all of them are integers.

Lemma 5. *$LT_d \subseteq \widehat{LT}_{d+1}$.*

This means that any depth-d linear threshold circuit of polynomial size but arbitrary weights can be simulated by a depth-$(d+1)$ linear threshold circuit whose size and weights are both polynomially bounded. While Goldmann et al. (1992) gives an existence result, Goldmann & Karpinski (1998) gives a constructive proof and it is easy to check that the underlying simulation is efficient and can be computed in polynomial time as well. Hofmeister (1996) has a simplified proof of Goldmann & Karpinski (1998) with improved explicit bounds.

Bourgain's theorem has also been extended to the case of boolean functions with inputs that come from constant biased distributions over $\{-1, 1\}^n$ in Kindler & Safra (2002). Our general result can be extended to these cases as well. For this we need to define the λ-noise-sensitivity of a boolean function with respect to μ_p, where μ_p is the distribution that picks -1 with probability p and 1 with probability $1-p$.

Definition 4. *The λ-noise-sensitivity of a Boolean funciton $f : \{-1,1\}^n \to \{-1,1\}$ with respect to μ_p is defined as*

$$\mathrm{NS}_{\lambda,p}\left(f\right) = \textit{Prob}_{x,y}\left(f(x) \neq f(y)\right)$$

where $x \sim \mu_p^n$ and y is constructed by first sampling coordinates I from $[n]$ according to μ_λ^n and then replacing those coordinates in x by coordinates independently sampled from μ_p^I.

Lemma 6. *For any parameter $\lambda > 0$, fix $k = log_{1-\lambda}(1/2)$. Then every Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ whose λ-noise-sensitivity with respect to μ_p^n is bounded by $(\epsilon/k)^2$, is a $\max[O(\epsilon \log(1/p)/p^2), J]$-junta, where*

$$J = O\left(\frac{k^3}{\epsilon^2 p^k}\right)$$

5 Proof of Theorem 1

Proof. (Proof of Theorem 1) The proof immediately follows from Proposition 1 and the following easy lemma.

Lemma 7. *Any $f : \{-1,1\}^n \to \{-1,1\}$ that is a k-junta can be represented by a depth-2 linear threshold circuit with the number of nodes and weights bounded by $2^{O(k)}$.*

Proof. Since f is a k-junta we can pretend that $f : \{-1,1\}^k \to \{-1,1\}$. Each positive example $x \in \{-1,1\}^k$ such that $f(x) = 1$ can be isolated by a single halfspace $h(y) = \mathrm{sgn}\left(\langle x, y\rangle - (k - 1/2)\right)$, which outputs positive value for $y \in \{-1,1\}^k$ iff $x = y$. We can build a depth-2 linear threshold circuit where all the hidden nodes correspond to $h(x)$, one for each positive examples of f. Thus, for a positive example of f, exactly one of the hidden layer node outputs 1. Otherwise, all hidden layer nodes output -1. Now we can have a linear threshold gate are the top with all weights 1 and threshold $1 - p$, where p is the number of positive examples of f. Note that all the hidden threshold gates have integer weights bounded by k and they are at most 2^k in number. The top gate has integer weights bounded by 2^k. Thus, f can be represented by an LT_2 or depth-2 linear threshold circuit where the size of the circuit and the integer weights used in it are bounded by $2^{O(k)}$. □

Therefore, combining this with Proposition 1, we get that any noise-stable f as required in Theorem 1 is δ-close to a depth-2 linear threshold circuit whose size and integer weights are bounded by $2^{O(k)}$, where

$$k = \left(\frac{1}{\delta\epsilon}\right)^{O(1/\epsilon)},$$

independent of n. □

6 Proof of Theorem 2

Since Bourgain's theorem can be improved for linear threshold functions with polynomial dependency in the noise and approximation parameters, we can approximate the given function using a junta where the number of junta variables is polynomially bounded. Due to Lemma 4, we can moreover, say that our function is not just close to a junta but close to a linear threshold function defined over these junta variables. The only caveat is that the weights used in this linear threshold function may be large. This is where we invoke size-depth-weight trade-off result such as Proposition 5 from circuit complexity to simulate this linear threshold function by a linear threshold circuit with an extra depth but polynomially bounded weights.

Proof. (Proof of Theorem 2) From Proposition 3, we see that any linear threshold function f with low noise-sensitivity $\mathrm{NS}_\epsilon\left(f\right) = O\left(\delta^3\sqrt{\epsilon}\right)$ is δ-close to an $O\left(1/\epsilon^2\ \log\left(1/\epsilon\right)\ \log\left(1/\delta\right)\right)$-junta. From Lemma 4, moreover, it must be δ-close a linear threshold function over these junta variables.

Thus, f is δ-close to an LT_1 function over these junta variables but the weights could be large. However, Proposition 5 shows that this can be simulated by an LT_2 function over these junta variables with weights polynomially bounded in the number of junta variables. Therefore, f is δ-close to an LT_2 function over $O\left(1/\epsilon^2 \ \log\left(1/\epsilon\right) \ \log\left(1/\delta\right)\right)$ variables with the size of the circuits and the weights at the threshold gates polynomially bounded in $1/\epsilon$ and $1/\delta$, but independent of n. This concludes the proof of Theorem 2. □

7 Proof of Theorem 3

Proof. (Proof of Theorem 3) Looking at Theorem 2, the broad outline of the algorithm is as follows. As seen in the proof of Theorem 2, we know that the given linear threshold function of low noise-sensitivity is close to another linear threshold function that depends only on a small, constant number of input variables. We can go over each small subset by brute force. Now over each small subset, we can try to learn a linear threshold function over them that is closest to the given function. Here we use a result from De et al. (2014) (see Theorem 36 of De et al. (2014)) on agnostic-type learning halfspaces via reconstructing the Chow parameters of a linear threshold function; Chow parameters are the level-0 and level-1 Fourier coefficients which are known to completely determine a linear threshold function.

Lemma 8. *Let $f : \{-1,1\}^n \to \{-1,1\}$ and let opt be the minimum disagreement (in fraction of the inputs) of f with its closest linear threshold function. Then given any $0 < \epsilon, \gamma < 1/2$ and access to independent uniform samples $(x, f(x))$, we can output a linear threshold function g (given by its weights) such that, with probability $1 - \gamma$,*

$$d(f,g) \leq 2^{-\Omega(\sqrt[3]{\log(1/\mathit{opt})})} + \epsilon,$$

where the algorithm runs in time

$$\tilde{O}(n^2) \cdot \left(\frac{1}{\epsilon}\right)^{O(\log^2(1/\epsilon))} \cdot \log\left(\frac{1}{\gamma}\right).$$

An immediate corollary that is useful to us is

Corollary 1. *Let $f : \{-1,1\}^n \to \{-1,1\}$ be a boolean function that is δ-close to a linear threshold function in a given subset $S \subseteq [n]$ of k input variables. Then, for $0 < \delta, \gamma < 1/2$, and given access to independent uniform examples $(x, f(x))$, we can output a linear threshold function g (given by its weights) such that, with probability $1 - \gamma$,*

$$d(f,g) \leq 2^{-\Omega(\sqrt[3]{\log(1/\delta)})} + \delta,$$

where the algorithm runs in time

$$\tilde{O}(k^2) \ \left(\frac{1}{\delta}\right)^{O(\log^2(1/\delta))} \ \log\left(\frac{1}{\gamma}\right).$$

Thus, we go over all subsets of size $O\left(1/\epsilon^2 \cdot \log(1/\epsilon) \cdot \log(1/\delta)\right)$ and run the agnostic-type learning of linear threshold functions by De et al. (2014). We take the best of these and convert the corresponding output, which is a linear threshold function with weights possibly exponential in $1/\epsilon$ and $1/\delta$, and apply Goldmann & Karpinski (1998) to convert it into a depth-2 linear threshold circuit whose size and weights both are polynomially bounded in $1/\epsilon$ and $1/\delta$. □

8 Conclusion and Future work

We show an efficient analog of the universal approximation theorem for neural networks in the case of noise-sensitive halfspaces of boolean hypercube, and gave efficient learning algorithms for the same. We do this via an interplay of techniques from Fourier analysis over the boolean hypercube and size-weight-depth trade-off results on linear threshold circuits from circuit complexity.

One might be able to extend these result to continuous domains where the input is sampled uniformly from $[-1,1]^n$ by using the ANOVA (analysis of variance) decomposition of a function. However, to do this one will have to prove a Bourgain-type theorem for these settings.

References

H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, Dec 1974. ISSN 0018-9286. doi: 10.1109/TAC.1974.1100705.

Sarah R. Allen, Ryan O'Donnell, and David Witmer. How to refute a random CSP. *CoRR*, abs/1505.04383, 2015. URL http://arxiv.org/abs/1505.04383.

Shun-ichi Amari. The handbook of brain theory and neural networks. chapter Learning and Statistical Inference, pp. 522–526. MIT Press, Cambridge, MA, USA, 1998. ISBN 0-262-51102-9. URL http://dl.acm.org/citation.cfm?id=303568.303829.

Alexandr Andoni, Rina Panigrahy, Gregory Valiant, and Li Zhang. Learning polynomials with neural networks. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 1908–1916, 2014. URL http://jmlr.org/proceedings/papers/v32/andoni14.html.

Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 584–592, 2014. URL http://jmlr.org/proceedings/papers/v32/arora14.html.

Rosa I. Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. *Machine Learning*, 63(2):161–182, 2006. ISSN 1573-0565. doi: 10.1007/s10994-006-6265-7. URL http://dx.doi.org/10.1007/s10994-006-6265-7.

Andrew R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993. doi: 10.1109/18.256500. URL http://dx.doi.org/10.1109/18.256500.

Peter L. Bartlett. Vapnik-chervonenkis dimension bounds for two- and three-layer networks. *Neural Computation*, 5(3):371–373, 1993. doi: 10.1162/neco.1993.5.3.371. URL http://dx.doi.org/10.1162/neco.1993.5.3.371.

Peter L. Bartlett and Shai Ben-David. Hardness results for neural network approximation problems. *Theoretical Computer Science*, 284(1):53 – 66, 2002. ISSN 0304-3975. doi: http://dx.doi.org/10.1016/S0304-3975(01)00057-3. URL http://www.sciencedirect.com/science/article/pii/S0304397501000573. Computing Leariníng Theory.

Eric B. Baum and David Haussler. What size net gives valid generalization? *Neural Comput.*, 1(1):151–160, March 1989. ISSN 0899-7667. doi: 10.1162/neco.1989.1.1.151. URL http://dx.doi.org/10.1162/neco.1989.1.1.151.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13:281–305, February 2012. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=2188385.2188395.

Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In *Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '93, pp. 278–291, London, UK, UK, 1994. Springer-Verlag. ISBN 3-540-57766-1. URL http://dl.acm.org/citation.cfm?id=646758.759585.

J. Bourgain. On the distribution of the fourier spectrum of Boolean functions. *Israel Journal of Mathematics*, 131:269–276, 2002. doi: 10.1007/BF02785861.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, September 1995. ISSN 0885-6125. doi: 10.1023/A:1022627411411. URL `http://dx.doi.org/10.1023/A:1022627411411`.

George Cybenko. Approximation by superpositions of a sigmoidal function. *MCSS*, 5(4):455, 1992. doi: 10.1007/BF02134016. URL `http://dx.doi.org/10.1007/BF02134016`.

Amit Daniely. A PTAS for agnostically learning halfspaces. *CoRR*, abs/1410.7050, 2014. URL `http://arxiv.org/abs/1410.7050`.

Amit Daniely. Complexity theoretic limitations on learning halfspaces. *CoRR*, abs/1505.05800, 2015. URL `http://arxiv.org/abs/1505.05800`.

Amit Daniely, Nati Linial, and Shai Shalev-Shwartz. From average case complexity to improper learning complexity. *CoRR*, abs/1311.2272, 2013. URL `http://arxiv.org/abs/1311.2272`.

Anindya De, Ilias Diakonikolas, Vitaly Feldman, and Rocco A. Servedio. Nearly optimal solutions for the chow parameters problem and low-weight approximation of halfspaces. *J. ACM*, 61(2):11:1–11:36, 2014. doi: 10.1145/2590772. URL `http://doi.acm.org/10.1145/2590772`.

J. de Villiers and E. Barnard. Backpropagation neural nets with one and two hidden layers. *Neural Networks, IEEE Transactions on*, 4(1):136–141, Jan 1993. ISSN 1045-9227. doi: 10.1109/72.182704.

I. Diakonikolas, R. Jaiswal, R. A. Servedio, L.-Y. Tan, and A. Wan. Noise Stable Halfspaces are Close to Very Small Juntas. November 2014.

Vitaly Feldman and Jan Vondrak. Optimal bounds on approximation of submodular and xos functions by juntas. In *Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, FOCS '13, pp. 227–236, Washington, DC, USA, 2013. IEEE Computer Society. ISBN 978-0-7695-5135-7. doi: 10.1109/FOCS.2013.32. URL `http://dx.doi.org/10.1109/FOCS.2013.32`.

Mikael Goldmann and Marek Karpinski. Simulating threshold circuits by majority circuits. *SIAM Journal on Computing*, 27(1):230–246, 1998. doi: 10.1137/S0097539794274519. URL `http://dx.doi.org/10.1137/S0097539794274519`.

Mikael Goldmann, Johan Håstad, and Alexander Razborov. Majority gates vs. general weighted threshold gates. *computational complexity*, 2(4):277–300, 1992. ISSN 1420-8954. doi: 10.1007/BF01200426. URL `http://dx.doi.org/10.1007/BF01200426`.

V. Guruswami and P. Raghavendra. Hardness of learning halfspaces with noise. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pp. 543–552, Oct 2006. doi: 10.1109/FOCS.2006.33.

Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. *SIAM Journal on Computing*, 39(2):742–765, 2009. doi: 10.1137/070685798. URL `http://dx.doi.org/10.1137/070685798`.

Thomas Hofmann, Bernhard Schlkopf, and Alexander J. Smola. Kernel methods in machine learning. *Ann. Statist.*, 36(3):1171–1220, 06 2008. doi: 10.1214/009053607000000677. URL `http://dx.doi.org/10.1214/009053607000000677`.

Thomas Hofmeister. *Computing and Combinatorics: Second Annual International Conference, COCOON '96 Hong Kong, June 17–19, 1996 Proceedings*, chapter A note on the simulation of exponential threshold weights, pp. 136–141. Springer Berlin Heidelberg, Berlin, Heidelberg, 1996. ISBN 978-3-540-68461-9. doi: 10.1007/3-540-61332-3_146. URL `http://dx.doi.org/10.1007/3-540-61332-3_146`.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4 (2):251–257, 1991. doi: 10.1016/0893-6080(91)90009-T. URL `http://dx.doi.org/10.1016/0893-6080(91)90009-T`.

Kurt Hornik, Maxwell B. Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. doi: 10.1016/0893-6080(89) 90020-8. URL `http://dx.doi.org/10.1016/0893-6080(89)90020-8`.

Marek Karpinski and Angus Macintyre. Polynomial bounds for $\{\mathrm{VC}\}$ dimension of sigmoidal and general pfaffian neural networks. *Journal of Computer and System Sciences*, 54(1):169 – 176, 1997. ISSN 0022-0000. doi: http://dx.doi.org/10.1006/jcss.1997.1477. URL `http://www.sciencedirect.com/science/article/pii/S002200009791477X`.

Guy Kindler and Shmuel Safra. Noise-resistant boolean functions are juntas. *preprint*, 2002.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Mach. Learn.*, 2(4):285–318, April 1988. ISSN 0885-6125. doi: 10.1023/A:1022869011914. URL `http://dx.doi.org/10.1023/A:1022869011914`.

Wolfgang Maass. Vapnik-chervonenkis dimension of neural nets. In Michael A. Arbib (ed.), *Handbook of Brain Theory and Neural Networks*, pp. 1000–1003. MIT Press, 1995.

Marvin Minsky and Seymour Papert. *Perceptrons - an introduction to computational geometry*. MIT Press, 1987. ISBN 978-0-262-63111-2.

E. Mossel and J. Neeman. Noise Stability and Correlation with Half Spaces. *ArXiv e-prints*, March 2016.

Ryan O'Donnell and Rocco A. Servedio. The chow parameters problem. *SIAM J. Comput.*, 40(1):165–199, 2011. doi: 10.1137/090756466. URL `http://dx.doi.org/10.1137/090756466`.

Yuval Peres. Noise stability of weighted majority. 2004. URL `http://arxiv.org/abs/math/0412377`.

Allan Pinkus. Approximation theory of the mlp model in neural networks. *Acta Numerica*, 8:143–195, 1 1999. ISSN 1474-0508. doi: 10.1017/S0962492900002919. URL `http://journals.cambridge.org/article_S0962492900002919`.

Patrice Y. Simard, Dave Steinkraus, and John C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 2*, ICDAR '03, pp. 958–, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1960-1. URL `http://dl.acm.org/citation.cfm?id=938980.939477`.

Kai-Yeung Siu, Vwani Roychowdhury, and Thomas Kailath. *Discrete Neural Computation: A Theoretical Foundation*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995. ISBN 0-13-300708-1.

Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and the closest pair problem. *J. ACM*, 62(2):13:1–13:45, May 2015. ISSN 0004-5411. doi: 10.1145/2728167. URL `http://doi.acm.org/10.1145/2728167`.

Leslie G. Valiant. A neuroidal architecture for cognitive computation. *J. ACM*, 47(5):854–882, September 2000. ISSN 0004-5411. doi: 10.1145/355483.355486. URL `http://doi.acm.org/10.1145/355483.355486`.