ELSEVIER

Theoretical Computer Science 284 (2002) 53–66

Theoretical
Computer Science

www.elsevier.com/locate/tcs

Hardness results for neural network approximation problems

Peter L. Bartlett[a,*], Shai Ben-David[b]

[a]Research School of Information Sciences and Engineering, Australian National University, Canberra ACT 0200, Australia
[b]Department of Computer Science, Technion, Haifa 32000, Israel

Abstract

We consider the problem of efficiently learning in two-layer neural networks. We investigate the computational complexity of agnostically learning with simple families of neural networks as the hypothesis classes. We show that it is NP-hard to find a linear threshold network of a fixed size that approximately minimizes the proportion of misclassified examples in a training set, even if there is a network that correctly classifies all of the training examples. In particular, for a training set that is correctly classified by some two-layer linear threshold network with k hidden units, it is NP-hard to find such a network that makes mistakes on a proportion smaller than c/k^2 of the examples, for some constant c. We prove a similar result for the problem of approximately minimizing the quadratic loss of a two-layer network with a sigmoid output unit.

1. Introduction

Previous negative results for learning two-layer neural network classifiers show that it is difficult to find a network that correctly classifies all examples in a training set. However, for learning to a particular accuracy it is only necessary to approximately solve this problem, that is, to find a network that correctly classifies *most* examples in a training set. In this paper, we show that this approximation problem is hard for several neural network classes.

The hardness of PAC style learning is a very natural question that has been addressed from a variety of viewpoints. The strongest non-learnability conclusions are those stating that no matter what type of algorithm a learner may use, as long as its computational resources are limited, it would not be able to predict a previously unseen label (with probability significantly better than that of a random guess). Such results have been derived by noticing that, in some precise sense, learning may be viewed

* Corresponding author.
E-mail addresses: peter.bartlett@anu.edu.au (P.L. Bartlett), shai@cs.technion.ac.il (S. Ben-David).

0304-3975/02/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0304-3975(01)00057-3

as breaking a cryptographic scheme. These strong hardness results are based upon assuming the security of certain cryptographic constructions (and in this respect are weaker than hardness results that are based on computational complexity assumptions like $\mathrm{P} \neq \mathrm{NP}$ or even $\mathrm{RP} \neq \mathrm{NP}$). The weak side of these results is that they apply only to classes that are rich enough to encode a cryptographic mechanism. For example, under cryptographic assumptions, Goldreich et al. [6] show that it is difficult to learn boolean circuits over n inputs with at most $p(n)$ gates, for some polynomial p. Kearns and Valiant [12] improve this result to circuits of polynomially many linear threshold gates and some constant (but unknown) depth. Thus, these techniques have not been so useful for analyzing neural networks as they have for understanding the hardness of learning classes of boolean circuits.

Another line of research considers agnostic learning using natural hypothesis classes. In such a learning setting, no assumptions are made about the rule used to label the examples, and the learner is required to find a hypothesis in the class that minimizes the labeling errors over the training sample. If such a hypothesis class is relatively small (say, in terms of its VC-dimension), then it can be shown that such a hypothesis will have a good prediction ability (that is, its test error will be close to its training error).

There are several hardness results in this framework. The first type are results showing hardness of finding a member of the hypothesis class that indeed minimizes the number of misclassifications over a given labeled sample. Blum and Rivest [3] prove that it is NP-hard to decide if there is a two-layer linear threshold network with only two hidden units that correctly classifies *all* examples in a training sample. (Our main reduction uses an extension of the technique used by Blum and Rivest.) They also show that finding a conjunction of k linear threshold functions that correctly classifies all positive examples and some constant proportion of negative examples is as hard as coloring an n-vertex k-colorable graph with $\mathrm{O}(k \log n)$ colors. DasGupta et al. [4] extend Blum and Rivest's results to two-layer networks with piecewise linear hidden units. Megiddo [15] shows that it is NP-hard to decide if any boolean function of two linear threshold functions can correctly classify a training sample.

The weakness of such results is that, for the purpose of learning, one can settle for *approximating* the best hypothesis in the class, while the hardness results apply only to *exactly* meeting the best possible error rate.

Related results show the hardness of 'robust learning'. A robust learner should be able to find, for any given labeled sample, and for *every* $\varepsilon > 0$, a hypothesis with training error rate within ε of the best possible within the class, in time polynomial in the sample size and in $1/\varepsilon$. Höffgen and Simon [8] show that, assuming $\mathrm{RP} \neq \mathrm{NP}$, no such learner exists for some subclasses of the class of half-spaces. Judd [10] shows NP-hardness results for an approximate sample error minimization problem for certain linear threshold networks with many outputs.

One may argue, that, for all practical purposes, a learner may be considered successful once it finds a hypothesis that approximates within ε the target (or the best hypothesis in a given class) for some *fixed* small ε. Such learning is not ruled out by ruling out robust learning.

We are therefore led to the next level of hardness-of-learning results, showing hardness of approximating the best fitting hypothesis in the class to within some *fixed* error rate. Arora et al. [1] show that, for any constant, it is NP-hard to find a linear threshold function that has the ratio of the number of misclassifications to the optimum number below that constant. Höffgen and Simon [8] show a similar result. We extend this type of result to richer classes of neural networks.

The neural networks that we consider have two layers, with a fixed number of linear threshold units in the first layer and a variety of output units. For pattern classification, we consider output units that compute boolean functions, and for real prediction we consider sigmoidal output units. Both problems can be expressed in a probabilistic setting, in which the training data is generated by some probability distribution, and we attempt to find a function that has near-minimal expected loss with respect to this distribution (see, for example, [7]). For pattern classification, we use the discrete loss; for real estimation, we use the quadratic loss. In both cases, efficiently finding a network with expected loss nearly minimal is equivalent to efficiently finding a network that has the sample average of loss nearly minimal. In this paper, we give results that quantify the difficulty of these approximate sample error minimization problems. For the pattern classification problem, we show that it is NP-hard to find a network with k linear threshold units in the first layer and an output unit that computes a conjunction that has proportion of data correctly classified within c/k of optimal, for some constant c. We extend this result to two-layer linear threshold networks (that is, where the output unit is also a linear threshold unit). In this case, the problem is hard to approximate within c/k^2 for some constant c. Further extensions of these results apply to the class of two-layer neural nets with k linear threshold units in the first layer and an output unit from any class of boolean functions that contains the conjunction. In this case the approximation constant for which we can show hardness is of the form $c/2^k$. These results apply even when there is a network that correctly classifies all of the data.

The case of quadratic loss has also been studied recently. Jones [9] considers the problem of approximately minimizing the sample average of the quadratic loss over a class of two-layer networks with sigmoid units in the first layer and a linear output unit with constraints on the size of the output weights. He shows that this approximation problem is NP-hard, for approximation accuracies of order $1/m$, where m is the sample size. The weakness of these results is that the approximation accuracy is sufficiently small to ensure that every single training example has small quadratic loss, a requirement that exceeds the sufficiency conditions needed to ensure valid generalization. Vu [18] has used results on hardness of approximations to improve Jones' results. He shows that the problem of approximately minimizing the sample average of the quadratic loss of a two-layer network with k linear threshold hidden units and a linear output unit remains hard when the approximation error is as large as $ck^{-3/2}n^{-3/2}$, where c is a constant and n is the input dimension. The hard samples in Vu's result have size that grows polynomially with n, so once again, the approximation threshold is a decreasing function of m.

In this paper, we also study the problem of approximately minimizing quadratic loss. We consider the class of two-layer networks with linear threshold units in the first layer and a sigmoid output unit (and no constraints on the output weights). We show that it is NP-hard to find such a network that has the sample average of the quadratic loss within c/k^2 of its optimal value, for some constant c. This result is true even when the infimum over all networks of the error on the training data is zero. One should note that our results show hardness for an approximation value that is independent of input dimension and of the sample size.

All of the learning problems studied in this paper can be solved efficiently if we fix the input dimension and the number of hidden units k. In that case, the algorithm 'Splitting' described in [14] (see also [5] and [13]) efficiently enumerates all training set dichotomies computed by a linear threshold function.

2. Preliminary definitions and notation

2.1. Approximate optimization basics

A maximization problem A is defined as follows. Let m_A be a non-negative objective function. Given an input x, the goal is to find a solution y for which the objective function $m_A(x,y)$ is maximized. Define $\mathrm{opt}_A(x)$ as the maximum value of the objective function. (We assume that, for all $x, m_A(x,\cdot)$ is not identically zero, so that the maximum is positive.) The *relative error* of a solution y is defined as $(\mathrm{opt}_A(x) - m_A(x,y))/\mathrm{opt}_A(x)$.

Our proofs use *L-reductions* (see [16, 11]), which preserve approximability. An L-reduction from one optimization problem A to another B is a pair of functions $\mathscr{F}$ and $\mathscr{G}$ that are computable in polynomial time and satisfy the following conditions:

1. $\mathscr{F}$ maps from instances of A to instances of B.
2. There is a positive constant α such that, for all instances x of A, $\mathrm{opt}_B(\mathscr{F}(x)) \leqslant \alpha\mathrm{opt}_A(x)$.
3. $\mathscr{G}$ maps from instances of A and solutions of B to solutions of A.
4. There is a positive constant β such that, for instances x of A and all solutions y of $\mathscr{F}(x)$, we have

$$\mathrm{opt}_A(x) - m_A(x, \mathscr{G}(x,y)) \leqslant \beta(\mathrm{opt}_B(\mathscr{F}(x)) - m_B(\mathscr{F}(x), y)).$$

The following lemma is immediate from the definitions.

Lemma 1. *Let A and B be maximization problems. Suppose that it is* NP-*hard to approximate A with relative error less than δ, and that A L-reduces to B with constants α and β. Then it is* NP-*hard to approximate B with relative error less than $\delta/(\alpha\beta)$.*

Clearly, this lemma remains true if we relax condition (4) of the L-reduction, so that it applies only to solutions y of an instance $\mathscr{F}(x)$ that have relative error less than $\delta/(\alpha\beta)$.

For all of the problems studied in this paper, we define the objective function such that $\max_x \mathrm{opt}_A(x) = 1$. With this normalization condition, we say that an L-reduction *preserves maximality* if $\mathrm{opt}_A(x) = 1$ implies $\mathrm{opt}_B(\mathscr{F}(x)) = 1$. (This is a special case of Petrank's notion [17] of preserving the 'gap location' in reductions between optimization problems.) The following lemma is also trivial.

Lemma 2. *Let A and B be maximization problems. Suppose that it is* NP-*hard to approximate A with relative error less than* δ, *even for instances with* $\mathrm{opt}_A(x) = 1$. *If A L-reduces to B with constants* α *and* β, *and the L-reduction preserves maximality, then it is* NP-*hard to approximate B with relative error less than* $\delta/(\alpha\beta)$, *even for instances with* $\mathrm{opt}_A(x) = 1$.

2.2. Families of boolean functions

We introduce some definitions and notations concerning functions that map $\{0,1\}^k$ to $\{0,1\}$ (for some k).

Definition.

- A function f is a *generalized conjunction* if $|f^{-1}(1)| = 1$ (so, in particular, the conjunction is such a function).
- A function f is *monotone* if there exists a boolean vector $a = (a_1, \ldots, a_n) \in \{0,1\}^k$ so that, for every $x \in f^{-1}(1)$ and every $1 \leqslant i \leqslant k$, if $x_i = a_i$ and $y \in \{0,1\}^k$ is identical to x except that its ith entry is flipped ($y_i \neq x_i$), then $f(y) = 1$. Note that every generalized conjunction is monotone.
- A class of boolean functions F is *monotone* if every function $g \in F$ is a monotone function.
- A class of boolean functions is *semi-monotone* if for every $g \in F$, if for some $x \in g^{-1}(1)$, for every y that is obtained by flipping exactly one bit of x, $g(y) = 0$, then g is a generalized conjunction.

Note that every linear threshold function is monotone. Note also every monotone family of functions is semi-monotone. It follows that every class of linear threshold functions is a semi-monotone class.

3. Results

In this section we describe our hardness results. The proofs of these results are deferred to the following section where we discuss the needed reductions. We first consider two-layer networks with k linear threshold units in the first layer and an output unit that computes a generalized conjunction. These networks compute functions of the form $f(x) = g(f_1(x), \ldots, f_k(x))$, where g is a generalized conjunction and each f_i is a linear threshold function of the form $f_i(x) = \mathrm{sgn}(w_i \cdot x - \theta_i)$ for some $w_i \in \mathbf{R}^n, \theta_i \in \mathbf{R}$. Here, $\mathrm{sgn}(\alpha)$ is 1 if $\alpha \geqslant 0$ and 0 otherwise. Let $N_n^{g,k}$ denote this class of functions.

Max k-And Consistency.
Given: A generalized conjunction function g.
Input: A sequence S of labeled examples, $(x_i, y_i) \in \{0,1\}^n \times \{0,1\}$.
Goal: Find a function f in $N_n^{g,k}$ that maximizes the proportion of consistent examples, $(1/m)|\{i\colon\ f(x_i) = y_i\}|$.

The condition $\mathrm{opt}_{\textsc{Max}\,k\text{-}\textsc{And Consistency}}(S) = 1$ in the following theorem corresponds to the case in which the training sample is consistent with some function in $N_n^{g,k}$.

Theorem 1. *Suppose $k \geqslant 3$. It is* NP-*hard to approximate* Max k-And Consistency *with relative error less than* $1/(136k)$. *Furthermore, there is a constant c such that even when* $\mathrm{opt}_{\textsc{Max}\,k\text{-}\textsc{And Consistency}}(S) = 1$ *it is* NP-*hard to approximate* Max k-And Consistency *with relative error less than* c/k^2.

Classes of the form $N_n^{g,k}$ are somewhat unnatural, since the output unit is constrained to compute some fixed generalized conjunction. Let F be a set of boolean functions on k inputs, and let $N_n^{F,k}$ denote the class of functions of the form $f(x) = g(f_1(x), \ldots, f_k(x))$, where $g \in F$ and $f_1, \ldots, f_k$ are linear threshold functions.

For arbitrary classes F, we do not know how to extend Theorem 1 to give corresponding hardness result for the class $N_n^{F,k}$ over binary-vector inputs. However, we can obtain results of this form if we allow rational inputs.

Max k-F Consistency.
Input: A sequence S of labeled examples, $(x_i, y_i) \in \mathbf{Q}^n \times \{0,1\}$.
Goal: Find a function f in $N_n^{F,k}$ that maximizes the proportion of consistent examples, $(1/m)|\{i\colon\ f(x_i) = y_i\}|$.

Theorem 2. (1) *There exists a constant c such that for any semi-monotone class F of boolean functions containing the conjunction, for any $k \geqslant 3$, is* NP-*hard to approximate* Max k-F Consistency *with relative error less than c/k^2, even for instances with* $\mathrm{opt}_{\textsc{Max}\,k\text{-}F\ \textsc{Consistency}}(S) = 1$.

(2) *There exists a constant c' such that for every class F of boolean functions containing the conjunction, for every $k \geqslant 3$, it is* NP-*hard to approximate Max k-F* Consistency *with relative error less than $c'/2^k$, even for instances with* $\mathrm{opt}_{\textsc{Max}\,k\text{-}F\ \textsc{Consistency}}(S) = 1$.

Next, we consider the class of two-layer networks with linear threshold units in the first layer and a sigmoid output unit. That is, we consider the class $N_n^{\sigma,k}$ of real-valued functions of the form

$$f(x) = \sigma\left(\sum_{i=1}^{k} v_i f_i(x) + v_0\right),$$

where $v_i \in \mathbf{R}, f_1, \ldots, f_k$ are linear threshold functions, and $\sigma : \mathbf{R} \to \mathbf{R}$ is a fixed function. We require that the fixed function σ maps to the interval [0,1], is monotonically non-decreasing, and satisfies

$$\lim_{\alpha \to -\infty} \sigma(\alpha) = 0, \qquad \lim_{\alpha \to \infty} \sigma(\alpha) = 1.$$

(The limits 0 and 1 here can be replaced by any two distinct numbers.)

Max k-σ Consistency.
Input: A sequence S of labeled examples, $(x_i, y_i) \in \mathbf{Q}^n \times ([0,1] \cap \mathbf{Q})$.
Goal: Find a function f in $N_n^{\sigma,k}$ that maximizes
$1 - (1/m)\sum_{i=1}^{m}(y_i - f(x_i))^2$.

Theorem 3. *For $k \geqslant 3$, there is a constant c such that it is* NP-*hard to approximate* Max k-σ Consistency *with relative error less than c/k^2, even for samples with* $\mathrm{opt}_{\text{Max}\,k\text{-}\sigma\ \text{Consistency}}(S) = 1$.

4. Reductions

4.1. *Learning with a generalized conjunction output unit:* Max k-And Consistency

We give an L-reduction to Max k-Cut.

Max k-Cut.
Input: A graph $G = (V, E)$.
Goal: Find a color assignment $c : V \to [k]$ that maximizes the proportion of multicolored edges, $(1/|E|)|\{(v_1, v_2) \in E\colon\ c(v_1) \neq c(v_2)\}|$.

We use the following result, due to Kann et al. [11], to prove the first part of Theorem 1.

Theorem 4 (Kann et al. [11]). *For $k \geqslant 2$, it is* NP-*hard to approximate* Max k-Cut *with relative error less than* $1/(34(k-1))$.

For the second part of the theorem, we need a similar hardness result for k-colorable graphs. The following result is essentially due to Petrank [17]; Theorem 3.3 in [17] gives the hardness result without calculating the dependence of the gap on k. Using the reduction due to Papadimitriou and Yannakakis [16] that Petrank uses in the final step of his proof, one gets that this dependence is of the form c/k^2.

Theorem 5 (Petrank [17]). *For $k \geqslant 3$, there is a constant c such that it is* NP-*hard to approximate* Max k-Cut *with relative error less than c/k^2, even for k-colorable graphs.*

Given a graph $G = (V, E)$, we construct a sample $S = \mathscr{F}(G)$ for a Max k-And Consistency problem using a technique similar to that used by Blum and Rivest

[3]. The key difference is that we use multiple copies of certain points in the training sample, in order to preserve approximability.

Suppose $|V|=n$, and relabel $V=\{v_1,\dots,v_n\}\subset\{0,1\}^n$, where v_i is the unit vector with a 1 in position i and 0s elsewhere. For every edge $e=(v_i,v_j)\in E$ let $F(e)$ be the labeled sample consisting of

— $(0^n,1)$ (where 0^n is the all-0 vector in $\{0,1\}^n$),
— $(v_i,0)$, $(v_j,0)$, and
— $(v_i+v_j,1)$.

Let $\mathscr{F}(G)$ be the concatenation of the samples $F(e)$ for all $e\in E$. Clearly, for $S=\mathscr{F}(G)$, $|S|=4|E|$.

The proof of Theorem 1 relies on the following two lemmas.

Lemma 3. *For* $k\geqslant 2, \mathrm{opt}_{\text{MAX}\,k\text{-AND CONSISTENCY}}(\mathscr{F}(G))\geqslant(3+\mathrm{opt}_{\text{MAX}\,k\text{-CUT}}(G))/4$. (*Consequently, if* $\mathrm{opt}_{\text{MAX}\,k\text{-CUT}}(G)=1$ *then* $\mathrm{opt}_{\text{MAX}\,k\text{-AND CONSISTENCY}}(\mathscr{F}(G))=1$).

Proof. For concreteness, let us assume that g is the conjunction, $\bigwedge_{i=1}^{k}x_i$. Let c be the optimal coloring of V. Define hidden unit i as $f_i(x)=\mathrm{sgn}(w_i\cdot x-\theta_i)$, where $\theta_i=-\frac{1}{2}$ and $w_i=(w_{i,1},\dots,w_{i,n})\in\mathbf{R}^n$ satisfies $w_{i,j}$ takes value -1 if $c(v_j)=i$ and 1 otherwise. Clearly, the $|E|$ copies of $(0^n,1)$ are correctly classified. It is easy to verify that each $(v_i,0)$ is correctly classified. Finally, every labeled example $(v_i+v_j,1)$ corresponding to an edge $(v_i,v_j)\in E$ has

$$f_l(v_i+v_j)=\begin{cases}0 & \text{if } c(v_i)=c(v_j)=l,\\ 1 & \text{otherwise}\end{cases}$$

for $l=1,\dots,k$. Hence, for $S=\mathscr{F}(G)$,

$$\mathrm{opt}_{\text{MAX}\,k\text{-AND CONSISTENCY}}(S)\geqslant\frac{3|E|+|E|\mathrm{opt}_{\text{MAX}\,k\text{-CUT}}(G)}{4|E|}.\qquad\square \tag{1}$$

Notation. For a sample S and a solution f for the MAX k-AND CONSISTENCY problem for it, let c_f denote the profit of this solution, namely, $c_f=m_{\text{MAX}\,k\text{-AND CONSISTENCY}}\ (S,f)$. Abusing notation, if G is an input graph for MAX k-CUT and g is a solution for it, we shall also denote $m_{\text{MAX}\,k\text{-CUT}}(G,g)$ by c_g.

Lemma 4. *There exists a polynomial time algorithm that, given a graph G and a* MAX k-AND CONSISTENCY *solution f for* $\mathscr{F}(G)$, *finds a* MAX k-CUT *solution g for G such that* $c_g\geqslant 4(c_f-3/4)$.

Proof. Given a MAX k-AND CONSISTENCY solution for the sample $\mathscr{F}(G)$, $f=\bigwedge_{i=1}^{k}f_i$, define a coloring g of the graph $G=(V,E)$ as follows: If $f(v_i)=1$, set $g(v_i)=1$, otherwise set $g(v_i)=\min\{j:f_j(v_i)=0\}$.

Claim. *For every edge $e\in E$, if f is consistent with $F(e)$ then the coloring g assigns different colors to the vertices of e.*

Proof (*of the claim*). Let $e=(v_i,v_j)$. If $g(v_i)=g(v_j)$, then $f(v_i)=f(v_j)=0$ implies $f(v_i+v_j)=0$. To see this, suppose that $f(v_i)=0$ and $f(v_j)=0$. Then $g(v_i)=g(v_j)$ implies some l has $f_l(v_i)=f_l(v_j)=0$. But since we also have $f(0)=1$, we must have $f_l(0)=1$ and, since f_l is a linear threshold function, this implies $f_l(v_i+v_j)=0$. It follows that $f(v_i+v_j)=0$, contradicting the assumption that f is consistent with the labels of $F(e)$.

As each sample $F(e)$ consists of 4 examples,

$$|\{e\in E: f \text{ is consistent with } F(e)\}| \geqslant |\{(x,y)\in\mathscr{F}(G): f(x)=y\}| - \tfrac{3}{4}|\mathscr{F}(G)|.$$

The lemma is now established by recalling that $|\mathscr{F}(G)|=4|E|$, noting that

$$c_f=\frac{|\{(x,y)\in\mathscr{F}(G): f(x)=y\}|}{4|E|}$$

and

$$c_g=\frac{|\{e\in E: g \text{ assigns different colors to its vertices}\}|}{|E|}$$

and applying the above claim. □

Finally, we can reduce the problem of approximating Max k-Cut to that of approximating Max k-And Consistency.

Lemma 5. *There is an L-reduction from* Max k-Cut *to* Max k-And Consistency, *with parameters* $\alpha=k/(k-1)$ *and* $\beta=4$.

Proof. On input graph G construct the sample $\mathscr{F}(G)$ and apply the Max k-And Consistency approximation algorithm to it. Let f be the resulting solution and let g be the graph coloring that f induces by the transformation described in the proof of Lemma 4.

By Lemmas 3 and 4,

$$\begin{aligned}\mathrm{opt}_{\text{Max}\,k\text{-And Consistency}}(\mathscr{F}(G)) - c_f &\geqslant \tfrac{1}{4}(3+\mathrm{opt}_{\text{Max}\,k\text{-Cut}}(G)) - c_f\\ &\geqslant \tfrac{1}{4}(3+\mathrm{opt}_{\text{Max}\,k\text{-Cut}}(G)) - \tfrac{1}{4}(c_g+3)\\ &= \tfrac{1}{4}(\mathrm{opt}_{\text{Max}\,k\text{-Cut}}(G) - c_g).\end{aligned}$$

Note that, for any graph $G, \mathrm{opt}_{\text{Max}\,k\text{-Cut}}(G) \geqslant 1-1/k$. This, together with the fact that $\mathrm{opt}_{\text{Max}\,k\text{-And Consistency}}(\mathscr{F}(G)) \leqslant 1$, implies that

$$\mathrm{opt}_{\text{Max}\,k\text{-And Consistency}}(\mathscr{F}(G)) \leqslant \frac{k}{k-1}\,\mathrm{opt}_{\text{Max}\,k\text{-Cut}}(G).$$

Together with Theorems 4 and 5, this implies Theorem 1. □

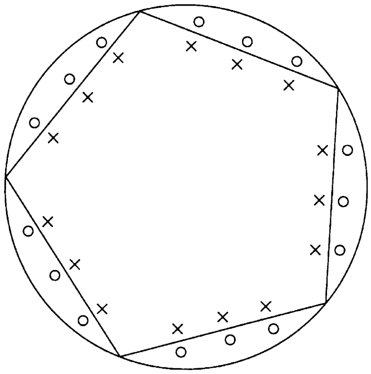

Fig. 1. The sets S_{in} and S_{out} used in the proof of Theorem 2, for the case $k=5$. The points in S_{in} are marked as crosses; those in S_{out} are marked as circles.

Proposition 1. *The hardness of the* Max k-And Consistency *problem is already manifest on sample inputs S for which* $\text{opt}_{\text{Max}\,k\text{-And Consistency}}(S) \geqslant 1 - 1/(4k)$.

Proof. Since for every graph $G, \text{opt}_{\text{Max}\,k\text{-Cut}}(G) \geqslant 1 - 1/k$, Lemma 3 implies that for every sample S of the form $\mathscr{F}(G)$ in the reduction above,

$$\text{opt}_{\text{Max}\,k\text{-And Consistency}}(S) \geqslant 1 - 1/(4k). \qquad \square$$

4.2. *Learning with an arbitrary output unit:* Max *k-F* Consistency

We apply two constructions, one for the case of semi-monotone classes of functions and one for the general case. Both constructions are L-reductions from Max k-And Consistency.

The case of semi-monotone F: For the proof of the claim for a family F of monotone functions we map each input S of Max k-And Consistency to a new sample $\mathscr{G}(S)$ by augmenting the input with two extra, rational, components, which we use to force the output unit ot compute a conjunction. For a labeled sample $S \subseteq \{0,1\}^n \times \{0,1\}$, we let $\mathscr{G}(S)$ consist of the following labeled points from $\mathbf{Q}^2 \times \{0,1\}^n \times \{0,1\}$:
— $3k$ copies of $((0,0),s)$, for each labeled point $s \in S$,
— $|S|$ copies of $(x, 0^n, 1)$ for $x \in S_{\text{in}} \subset \mathbf{Q}^2$, and
— $|S|$ copies of $(x, 0^n, 0)$ for $x \in S_{\text{out}} \subset \mathbf{Q}^2$,
where the sets S_{in} and S_{out} are defined as follows:

The sets S_{in} and S_{out} both have cardinality $3k$. Each point in S_{in} is paired with a point in S_{out}, and this pair straddles some edge of a regular k-sided polygon $\mathbf{R}^2$ that has vertices on the unit circle centered at the origin, as shown in Fig. 1. (We call this pair of points a 'straddling pair'.) The midpoint of each pair lies on some edge of the polygon, and the line passing through the pair is perpendicular to that edge. The set of

$3k$ midpoints (one for each pair) and the k vertices of the polygon are equally spaced around the polygon.

Clearly, $|\mathscr{G}(S)| = 9|S|k$.

Lemma 6. *For every sample* $S \subseteq \{0,1\}^n \times \{0,1\}$,

$$\mathrm{opt}_{\text{MAX } k\text{-}F \text{ CONSISTENCY}}(\mathscr{G}(S)) \geqslant \tfrac{1}{3}(\mathrm{opt}_{\text{MAX } k\text{-AND CONSISTENCY}}(S) + 2).$$

Proof. Given a solution f to the MAX k-AND CONSISTENCY problem on the input S, we extend it to a solution of MAX k-F CONSISTENCY on the input $\mathscr{G}(S)$ by augmenting each halfspace f with appropriate weights for the two additional inputs. We choose the output unit as a conjunction and arrange the new hidden unit weights so that the intersection of the hidden unit decision boundaries with the plane of the two additional inputs coincide with the k sides of the polygon. □

The resulting neural net classifies correctly all the points in $S_{\text{in}} \cup S_{\text{out}}$ as well as all the images of the points of S that are classified correctly by f. The lemma now follows by a straightforward calculation.

Lemma 7. *There exists a polynomial time algorithm that, given a sample* $S \subseteq \{0,1\}^n \times \{0,1\}$ *and a* MAX k-F CONSISTENCY *solution* g *for* $\mathscr{G}(S)$ *for which* $c_g > 1 - 1/(9k)$, *the algorithm finds a* MAX k-AND CONSISTENCY *solution* f *for* S, *such that* $c_f \geqslant 3(c_g - 2/3)$, *where* c_f *is the profit of the solution* f *for* S, *and* c_g *is the profit of the solution* g *for* $\mathscr{G}(S)$.

Proof. First note that, as we assume that $c_g > 1 - 1/(9k)$, g classifies correctly all the points in $S_{\text{in}} \cup S_{\text{out}}$. Let α denote the distance between a point in $S_{\text{in}} \cup S_{\text{out}}$ and the associated edge of the polygon. Clearly, since the points in $\{(x, 0^n) : x \in S_{\text{in}}\}$ are labeled 1 and those in $\{(x, 0^n) : x \in S_{\text{out}}\}$ are labeled 0, for every straddling pair described above, any function in $N^{F,k}_{n+2}$ that is consistent with these points has some hidden unit whose decision boundary separates the pair. It is easy to show using elementary trigonometry that there is a constant c such that, if $\alpha < c/k$, no line in $\mathbf{R}^2$ can pass between more than three of these pairs, and no line can pass between three unless they all straddle the same edge of the polygon. Let g be any function in $N^{F,k}_{n+2}$ that classifies correctly the points in $S_{\text{in}} \cup S_{\text{out}}$, and suppose that g is of the form $g = g_0(g_1, \dots g_k)$ for hidden units $g_1, \dots, g_k$. Since k lines must separate $3k$ straddling pairs, the decision boundaries of $g_1, \dots, g_k$ must be hyperplanes whose projections to the two rational coordinates of S are lines, each separating three straddling pairs. Thus, $(g_1(x, 0^n), \dots, g_k(x, 0^n))$ is a constant vector (which we denote h) for any $x \in S_{\text{in}}$, and it satisfies $g_0(h) = 1$. Furthermore, the points in S_{out} force the output to 0 for every vector that differs from the vector h at exactly one entry. Therefore, as F is semi-monotone, the output gate g_0 is a generalized conjunction. Without loss of generality, let $g = \bigwedge_{i=1}^{k} g_i$, and for each linear threshold function g_i let f_i be its composition with the projection to the coordinates of $\{0,1\}^n$. Let f be $\bigwedge_{i=1}^k f_i$. Note that for every point of the form $((0,0),s)$ in $\mathscr{G}(S)$, if g classifies it correctly, then f classifies s correctly. Since only $3k|S|$ of the $9k|S|$ points in $\mathscr{G}(S)$ are of this form, the number of such points classified correctly by g, counting multiple copies, is at least $9k|S|c_g - 6k|S|$, and so $|S|c_f \geqslant (9k|S|c_g - 6k|S|)/(3k)$, which implies the result.

The hardness of the approximation problem for Max k-F Consistency will be established once we reduce it to the problem Max k-And Consistency. The following lemma presents this reduction, for sample inputs S for which $\mathrm{opt}_{\text{Max } k\text{-And Consistency}}(S) \geqslant 1 - 1/(4k)$. By Proposition 1, this is sufficient.

Lemma 8. *There is an L-reduction from* Max k-And Consistency, *restricted to sample inputs for which* $\mathrm{opt}_{\text{Max } k\text{-And Consistency}}(S) \geqslant 1-1/(4k)$, *to* Max k-F Consistency, *with parameters* $\alpha = 4k/(4k-1)$ *and* $\beta = 3$.

Proof. The proof is similar to the proof of Lemma 5 using Lemmas 6 and 7 instead of Lemmas 3 and 4.

Combining this with Theorem 1 we get a proof for the first part of Theorem 2.

The case of unrestricted family F: To obtain the hardness results for an arbitrary family of functions in the output gate, we repeat the idea of the previous construction. However, we have to modify it because, without the assumption that F is semi-monotone, forcing the ouput gate to output 1 on one vector and output 0 on all its immediate neighbors does not yet force the output gate to compute a conjunction. To handle this difficulty, we replace the $\mathbf{Q}^2$ coordinates of the previous construction by $\mathbf{Q}^k$. We let $H_1, \ldots H_k$ be k faces of a k-dimensional regular simplex in $\mathbf{R}^k$ that contains the origin (that is, each H_i is a $(k-1)$-dimensional hyper-plane). Now $S_{\text{in}} \cup S_{\text{out}}$ consists of $k(k+1)$ pairs of points straddling these k hyperplanes ($(k+1)$ many pairs for each H_i). Furthermore, we place one member of S in each of the 2^k many cells defined by $H_1, \ldots H_k$ and label all of these points by 0 except for the point that shares the cell with the points of S_{in}—the cell to which the origin belongs.

Once a function $g(f_1 \ldots, f_k)$ classifies all these points correctly, it must be a generalized conjunction. Repeating the calculation above yields part 2 of Theorem 2.

4.3. Learning with a sigmoid output unit: Max k-σ Consistency

We give an L-reduction from Max k-F Consistency to Max k-σ Consistency, where F is the class of linear threshold functions. Given a sample S for a Max k-F Consistency problem, we use the same sample for the Max k-σ Consistency problem. Trivially,[1] if $\mathrm{opt}_{\text{Max } k\text{-}F \text{ Consistency}}(S) = 1$ then $\mathrm{opt}_{\text{Max } k\text{-}\sigma \text{ Consistency}}(S) = 1$. Furthermore, we have the following lemma.

[1] In this problem, the maximum might not exist since the restriction of the function class to the set of training examples is infinite, so we consider the problem of approximating the supremum.

Lemma 9. *For a solution f to* Max k-σ Consistency *with cost c_f, we can find a solution h for* Max k-F Consistency *with cost c_h, and*

$$1 - c_h \leqslant \tfrac{1}{4}(1 - c_f).$$

Proof. Suppose that

$$f(x) = \sigma\left(\sum_{i=1}^{k} v_i f_i(x) + v_0\right).$$

Without loss of generality, assume that $\sigma(0) = 1/2$. (In any case, adjusting v_0 gives a function $\tilde{\sigma}$ that satisfies $\inf\{\alpha\colon\ \tilde{\sigma}(\alpha) > 1/2\} = 0$, which suffices for the proof.) Now, if we replace $\sigma(\cdot)$ by $\operatorname{sgn}(\cdot)$, we obtain a function h for which $h(x_i) \neq y_i$ implies $(f(x_i) - y_i)^2 \geqslant 1/4$. It follows that $1 - c_h \leqslant (1 - c_f)/4$, as required. □

Thus, for the case $\mathrm{opt}_{\text{Max } k\text{-}F \text{ Consistency}}(S) = 1$, we have an L-reduction from Max k-F Consistency to Max k-σ Consistency, with parameters $\alpha = 1$ and $\beta = \frac{1}{4}$, and this L-reduction preserves maximality. Theorem 3 follows from Theorem 2.

5. Extensions and future work

It would be interesting to extend the hardness result for networks with real outputs to the case of a linear output unit with a constraint on the size of the output weights. We conjecture that a similar result can be obtained, with a relative error bound that—unlike Vu's result for this case [18]—does not decrease as the input dimension increases.

Recently, Ben-David et al. [2] obtained similar hardness of approximation results for a variety of concept classes including axis-aligned hyper-rectangles, closed balls and the classes of monomials and monotone monomials over the boolean cube.

From the point of view of learning, an algorithm can achieve good generalization by approximating the best hypothesis in some class $\mathscr{H}$ by a hypothesis from another class $\mathscr{H}'$, as long as $\mathscr{H}'$ has a small VC dimension. It would be interesting to know if hardness results similar to ours hold for that extended framework as well. There is some related work in this direction. Theorem 7 in [3] shows that finding a conjunction of k' linear threshold functions that correctly classifies a set that can be correctly classified by a conjunction of k linear threshold functions is as hard as coloring a k-colorable graph with n vertices using k' colors. Note, however, that this result holds only when the learning algorithm is required to output a hypothesis that has zero error. Recently, Ben-David, Eiron and Long obtained a corresponding hardness result for *approximating* the best hypothesis having k hidden units by a hypothesis having k' hidden units, as long as $k' < (49/48)k$. The cryptographic results mentioned in Section 1 do not have such strong restrictions on the hypothesis class. They can therefore be viewed as an answer to the above question, however, they apply only to classes that have the number of hidden units grow (polynomially) with the size of the training data. One should recall that the generalization ability of a hypothesis class deteriorates as the class grows. No such result is known for learning with fixed-size neural networks, which are the focus of investigation of this paper.

Acknowledgements

We wish to thank Phil Long for detecting an error in a previous version of the proof of Theorem 1 and Shirley Halevi and Dror Rawitz for helping to fix that error. This research was done while Shai Ben-David was visiting the Australian National University. It was supported in part by the Australian Research Council.

References

[1] S. Arora, L. Babai, J. Stern, Z. Sweedyk, Hardness of approximate optima in lattices, codes, and linear systems, J. Comput. System Sci. 54 (2) (1997) 317–331.

[2] S. Ben-David, N. Eiron, P. Long, On the difficulty of approximately maximizing agreements, Proc. 13th Ann. Conference on Computational Learning Theory, 2000, pp. 266–274.

[3] A.L. Blum, R.L. Rivest, Training a 3-node neural network is NP-complete, Neural Networks 5 (1) (1992) 117–127.

[4] B. DasGupta, H.T. Siegelmann, E.D. Sontag, On the complexity of training neural networks with continuous activation functions, IEEE Trans. Neural Networks 6 (6) (1995) 1490–1504.

[5] A. Faragó G. Lugosi, Strong universal consistency of neural network classifiers, IEEE Trans. Inform. Theory 39 (4) (1993) 1146–1151.

[6] O. Goldreich, S. Goldwasser, S. Micali, How to construct random functions, J. ACM 33 (1986) 792–807.

[7] D. Haussler, Decision theoretic generalizations of the PAC model for neural net and other learning applications, Inform. and Comput. 100 (1) (1992) 78–150.

[8] K.U. Höffgen, H.U. Simon, K.S. Van Horn, Robust trainability of single neurons, J. Comput. Systems Sci. 50 (1) (1995) 114–125.

[9] L.K. Jones, The computational intractability of training sigmoidal neural networks, IEEE Trans. Inform. Theory 43 (1) (1997) 167–713.

[10] J.S. Judd, Neural Network Design and the Complexity of Learning, MIT Press, Cambridge, MA, 1990.

[11] V. Kann, S. Khanna, J. Lagergren, A. Panconesi, On the hardness of approximating max-k-cut and its dual, Tech. Report CJTCS-1997-2, Chicago J. Theoret. Comput. Sci., 1997.

[12] M. Kearns, L.G. Valiant, Cryptographic limitations on learning Boolean formulae and finite automata, Proc. 21st Ann. ACM Symp. on Theory of Computing, 1989, pp. 433–444.

[13] P. Koiran, Efficient learning of continuous neural networks, Proc. 7th Ann. ACM Workshop on Computational Learning Theory, 1994, pp. 348–355.

[14] W.S. Lee, P.L. Bartlett, R.C. Williamson, Efficient agnostic learning of neural networks with bounded fan-in, IEEE Trans. Inform. Theory 42 (6) (1996) 2118–2132.

[15] N. Megiddo, On the complexity of polyhedral separability, Discrete Comput. Geom. 3 (1988) 325–337.

[16] C.H. Papadimitriou, M. Yannakakis, Optimization, approximation, and complexity classes, J. Comput. System Sci. 43 (1991) 425–440.

[17] E. Petrank, The hardness of approximation: Gap location, Computat. Complexity 4 (2) (1994) 133–157.

[18] V.H. Vu, On the infeasibility of training neural networks with small squared errors, in: M.I. Jordan, M.J. Kearns, S.A. Solla (Eds.), Advances in Neural Information Processing Systems, Vol. 10, The MIT Press, Cambridge, MA, 1998, pp. 371–377.