# On Approximate Learning by Multi-layered Feedforward Circuits

Bhaskar DasGupta*[1] and Barbara Hammer[2]

[1] Department of Computer Science,
Rutgers University, Camden, NJ 08102, U.S.A.
Email: bhaskar@crab.rutgers.edu.
[2] Department of Mathematics/Computer Science,
University of Osnabrück, D-49069 Osnabrück, Germany.
Email: hammer@informatik.uni-osnabrueck.de

**Abstract.** We consider the problem of efficient *approximate* learning by multilayered feedforward circuits subject to two objective functions.
First, we consider the objective to *maximize* the ratio of correctly classified points compared to the training set size (e.g., see [3, 5]). We show that for single hidden layer threshold circuits with $n$ hidden nodes and varying input dimension, approximation of this ratio within a relative error $c/n^3$, for some positive constant $c$, is NP-hard *even if* the number of examples is *limited* with respect to $n$. For architectures with two hidden nodes (e.g., as in [6]), approximating the objective within some fixed factor is NP-hard *even if any* sigmoid-like activation function in the hidden layer and $\varepsilon$-separation of the output [19] is considered, or if the semilinear activation function substitutes the threshold function.
Next, we consider the objective to *minimize* the *failure ratio* [2]. We show that it is NP-hard to approximate the failure ratio within every *constant larger than* 1 for a multilayered threshold circuit provided the input biases are zero. Furthermore, even *weak* approximation of this objective is *almost* NP-hard.

## 1 Introduction

Feedforward circuits are a well established learning mechanism which offer a simple and successful method of learning an unknown hypothesis given some examples. However, the inherent complexity of training the circuits is till now an open problem for most practically relevant situations. Starting with the work of Judd [15, 16] it turned out that training is NP-hard in general. However, most work in this area deals either with only very restricted architectures, activation functions not used in practice, or a training problem which is too strict compared to practical problems. In this paper we want to consider situations which are closer to the training problems as they occur in practice.

A *feedforward circuit* consists of nodes which are connected in a directed acyclic graph. The overall behavior of the circuit is determined by the *architecture* $\mathcal{A}$ and the circuit *parameters* $\boldsymbol{w}$. Given a *pattern* or *example* set $P$ consisting of points $(\boldsymbol{x}_i; y_i)$, we want to learn the regularity with a feedforward circuit. Frequently, this is performed by

* Research supported by NSF grant CCR-9800086.

first chosing an architecture $\mathcal{A}$ which computes a function $\beta_{\mathcal{A}}(\boldsymbol{w}, \boldsymbol{x})$ and then chosing the parameters $\boldsymbol{w}$ such that $\beta_{\mathcal{A}}(\boldsymbol{w}, \boldsymbol{x}_i) = y_i$ holds for every pattern $(\boldsymbol{x}_i; y_i)$. The *loading problem* (or the *training problem*) is the problem to find weights $\boldsymbol{w}$ such that these equalities hold. The *decision version* of the loading problem is to decide (rather than to find the weights) whether such weights exist that load $M$ onto $\mathcal{A}$.

Some previous results consider specific situations. For example, for every fixed architecture with threshold activation function or architectures with appropriately restricted connection graph loading is polynomial [8, 10, 15, 20]. For some strange activation functions or a setting where the number of examples coincides with the number of hidden nodes loadability becomes trivial [25]. However, Blum and Rivest [6] show that a varying input dimension yields the NP-hardness of training threshold circuits with only two hidden nodes. Hammer [10] generalizes this result to multilayered threshold circuits. References [8, 11, 12, 14, 23, 27] constitute generalizations to circuits with the sigmoidal activation function or other continuous activations. Hence finding an optimum weight setting in a concrete learning task may require a large amount of time.

Naturally, the constraint that all the examples must be correctly classified is too strict. In a practical situation, one would be satisfied if a large fraction (but not necessarily all) of the examples can be satisfied. Moreover, it may be possible that there are no choices for the weights which load a given set of examples. From these motivations, researchers have considered an approximate version of the learning problem where the number of correctly classified points is to be maximized. References [1, 2, 13] consider the complexity of training single threshold nodes with some error bounds. Bartlett and Ben-David [3] mostly deal with threshold architectures, whereas Ben-David et. al. [5] deals with other concept classes such as monomials, axis-aligned hyper-rectangles, monotone monomials and closed balls. We obtain NP-hardness results for the task of approximately minimizing the relative error of the success ratio for a correlated architecture and training set size, various more realistic activation functions, and training sets without multiple points. Another objective function is to approximately minimize the failure ratio. The work in [1, 2] considers inapproximability of minimizing the failure ratio for a single threshold gate. We show that approximating this failure ratio for multilayered threshold circuits within every constant is NP-hard and even weak approximation of this objective function is almost NP-hard. Several proofs are omitted due to space limitations. They can be found in the long version of this paper.

## 2 The Basic Model and Notations

The architecture of a feedforward circuit $\mathcal{C}$ is described by a directed interconnection graph and the activation functions of $\mathcal{C}$. A node $v$ of $\mathcal{C}$ computes a function

$$\gamma_v \left( \sum_{i=1}^{k} w_{v_i,v} u_{v_i} + b_v \right)$$

of its inputs $u_{v_1}, \ldots, u_{v_k}$. $\sum_{i=1}^{k} w_{v_i,v} u_{v_i} + b_v$ is called the *activation* of the node $v$. The inputs are either external, representing the input data, or internal, representing the outputs of the immediate predecessors of $v$. The coefficients $w_{v_i,v}$ (resp. $b_v$) are the *weights* (resp. *threshold*) of node $v$, and $\gamma_v$ is the *activation function* of $v$. No cycles are allowed in the interconnection graph of $\mathcal{C}$ and the output of a designated node provides the output of the circuit. An *architecture* specifies the interconnection structure and the $\gamma_v$'s, but not the actual numerical values of the weights or thresholds. The *depth* of a feedforward circuit is the length of the longest path of the interconnection graph. A *layered* feedforward circuit is one in which nodes at depth $d$ are connected only to nodes at depth $d+1$, and all inputs are provided to nodes at depth 1 only. A layered $(n_0, n_1, \ldots, n_h)$ circuit is a layered circuit with $n_i$ nodes at depth $i \geq 1$ where $n_0$ is the number of inputs. We assume $n_h = 1$. Nodes at depth $j$, for $1 \leq j < h$, are called *hidden nodes*, and all nodes at depth $j$, for a particular $j$, constitute the $j$th *hidden layer*.

A $\Gamma$-circuit $\mathcal{C}$ is a feedforward circuit in which only functions in some set $\Gamma$ are assigned to nodes. Hence each architecture $\mathcal{A}$ of a $\Gamma$-circuit defines a behavior function $\beta_{\mathcal{A}}$ that maps from the $r$ real weights and the $n$ inputs into an output value. We denote such a behavior as the function $\beta_{\mathcal{A}} \ : \ \mathbb{R}^{r+n} \mapsto \mathbb{R}$ . Some popular choices of the activation functions are the perceptron activation function $H(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$ and the standard sigmoid sgd$(x) = 1/(1 + \mathrm{e}^{-x})$.

The *loading problem* $L$ is defined as follows (e.g., see [6, 8]): Given an architecture $\mathcal{A}$ and a set of examples $P = \{(\boldsymbol{x}; y) \mid x \in \mathbb{R}^n, y \in \mathbb{R}\}$, find weights $\boldsymbol{w}$ so that for all $(\boldsymbol{x}; y) \in M$: $\beta_{\mathcal{A}}(\boldsymbol{w}, \boldsymbol{x}) = y$ . In this paper we will deal with those classification tasks where $y \in \{0, 1\}$. Clearly, the hardness results obtained with this restriction will be valid in the unrestricted case also. An example $(\boldsymbol{x}; y)$ is a *positive example* if $y = 1$, otherwise it is a *negative example*. An example is *misclassified* by the circuit if $\beta_{\mathcal{A}}(\boldsymbol{w}, \boldsymbol{x}) \neq y$, otherwise it is *classified correctly*.

An *optimization* problem $C$ is characterized by a non-negative objective function $m_C(x, y)$, where $x$ is an input instance of the problem, $y$ is a solution for $x$, and $m_C(x, y)$ is the cost of the solution $y$; the goal of the problem is to either maximize or minimize $m_C(x, y)$ for any particular $x$, depending on the problem. Denote by opt$_C(x)$ (or shortly opt$(x)$ if $C$ is clear from the context) the optimum value of $m_C(x, y)$. For maximization, (opt$_C(x) - m_C(x, y)$)/opt$_C(x)$ is the *relative error* of a solution $y$. The objective functions that are of relevance to this paper are as follows:

**Success ratio function:** $m_L(x, y) =\mid \{\boldsymbol{x} \mid \beta_{\mathcal{A}}(\boldsymbol{w}, \boldsymbol{x}) = y\} \mid / |P|$ is the fraction of the correctly classified examples compared to the training set size (e.g., see [3]).

**Failure ratio function:** $m_C(x, y) = \mid \{\boldsymbol{x} \mid \beta_{\mathcal{A}}(\boldsymbol{w}, \boldsymbol{x}) \neq y\} \mid$. If opt$_C(x) > 0$, $m_f(x, y) = m_C(x, y)/$opt$_C(x)$ is the ratio of the number of misclassified examples to the minimum possible number of misclassifications when at least one missclassification is unavoidable (e.g., see [2]).

## 3 Approximating the Success Ratio Function $m_L$

We want to show that in several situations it is difficult to approximate $m_L$ for a loading problem $L$. These results would extend the results of [3] to more complex situations. For this purpose, the L-reduction from the so-called MAX-$k$-cut problem to a loading problem which is constructed in [3] is generalized such that it can be applied to several further situations as well. Since approximating the MAX-$k$-cut problem is NP-hard, the NP-hardness of approximability of the latter problems follows.

**Definition 1.** *Given an undirected graph $G = (V, E)$ and $k \geq 2$ in $\mathbb{N}$, the MAX-$k$-cut problem is to find a function $\psi : V \mapsto \{1, 2, \ldots, k\}$, such that $|\{(u, v) \in E \mid \psi(u) \neq \psi(v)\}| \,/\, |E|$ is maximized. The set of nodes in $V$ which are mapped to $i$ in this setting is called the $i$th* cut. *The edges $(v_i, v_j)$ in the graph for which $v_i$ and $v_j$ are contained in the same cut are called monochromatic; all other edges are called bichromatic.*

**Theorem 1.** [17] *It is NP-hard to approximate the MAX-$k$-cut problem within relative error smaller than $1/(34(k-1))$ for $k \geq 2$, and within error smaller than $c/k^3$, $c$ being some constant, $k \geq 3$, even if solutions without monochromatic edges exist.*

The concept of an *$L$-reduction* was defined in [21]. The definition stated below is a slightly modified version of [21] that will be useful for our purposes.

**Definition 2.** *An $L$-reduction from a maximization problem $C_1$ to a maximization problem $C_2$ consists of two polynomial time computable functions $T_1$ and $T_2$, two constants $\alpha, \beta > 0$, and a parameter $0 \leq a \leq 1$ with the following properties:*

**(a)** *For each instance $I_1$ of $C_1$, algorithm $T_1$ produces an instance $I_2$ of $C_2$.*
**(b)** *The maxima of $I_1$ and $I_2$,* opt$(I_1)$ *resp.* opt$(I_2)$, *satisfy* opt$(I_2) \leq \alpha$ opt$(I_1)$.
**(c)** *Given any solution of the instance $I_2$ of $C_2$ with cost $c_2$ such that the* relative error *of $c_2$ is at most $a$, algorithm $T_2$ produces a solution $I_1$ of $C_1$ with cost $c_1$ satisfying* (opt$(I_1) - c_1$) $\leq$ $\beta$ (opt$(I_2) - c_2$).

*If $C_1$ is hard to approximate within relative error $a/(\alpha\beta)$ then $C_2$ is hard to approximate within relative error $a$.*

Consider an $L$-reduction from the MAX-$k$-cut problem to the loading problem $L$ with objective function $m_L$ where the reductions performed by $T_1$ and $T_2$ have the following additional properties. Given an instance $I_1 = (V, E)$ of the MAX-$k$-cut problem, assume that $T_1$ produces in polynomial time an instance $I_2$, a specific architecture and an example set in $\mathbb{R}^n \times \{0, 1\}$ of the loading problem $L$ with training set:

– $2|E|$ copies of each of some set of special points $P_0$ (e.g. the origin),
– for each node $v_i \in V$, $d_i$ copies of one point $\boldsymbol{e}_i$, where $d_i$ is the degree of $v_i$,
– for each edge $(v_i, v_j) \in E$, one point $\boldsymbol{e}_{ij}$.

Furthermore, assume that the following properties are satisfied:

**(i)** For an optimum solution for $I_1$ the algorithm $T_1$ finds an optimum solution of the instance $I_2$ of the corresponding loading problem $L$ in which all special points $P_0$ and all points $\boldsymbol{e}_i$ are correct classified and exactly those points $\boldsymbol{e}_{ij}$ are misclassified which correspond to a monochromatic edge $(v_i, v_j)$ in an optimal solution of $I_1$.
**(ii)** For any approximate solution of the instance $I_2$ of the loading problem $L$ which classifies all special points in $P_0$ correctly, $T_2$ computes an approximate solution of the instance $I_1$ of the MAX-$k$-cut problem such that for every monochromatic edge $(v_i, v_j)$ in this solution, either $\boldsymbol{e}_i$, $\boldsymbol{e}_j$, or $\boldsymbol{e}_{ij}$ is missclassified.

An analogous proof to [3] yields the following result:

**Theorem 2.** *Approximation of the above loading problem within relative error smaller than $((k-1)\epsilon)/(k(2|P_0|+3))$ is NP-hard since the above reduction is an $L$-reduction with $\alpha = k/(k-1)$, $\beta = 2|P_0|+3$, and $a = (k-1)/(k^2\ (2|P_0|+3))$.*

### 3.1 Application to Multi-layered Feedforward Circuits

First we consider $H$-circuits, $H(x)$ being the perceptron activation function. This type of architecture is common in theoretical study of neural networks (e.g., see [22, 24]) as well as in their practical applications (e.g., see [28]). Assume that the first layer contains the input nodes $1$, …, $n$, $h+1$ denotes the depth of the $H$-circuit, and $n_i$ denotes the number of nodes at depth $i$. An instance of the loading problem will be represented by a tuple $(n, n_1, n_2, \ldots, n_h, 1)$ and by an example set with rational numbers. The following fact is an immediate consequence of Theorem 2 in [3]:

For any $h \geq 1$, constant $n_1 \geq 2$ and any $n_2$, …, $n_h \in \mathbb{N}$, it is NP-hard to approximate the success ratio function $m_L$ with instances $(N, P)$, where $N$ is the architecture of a layered $\{(n, n_1, \ldots, n_h, 1) \mid n \in \mathbb{N}\}$ $H$-circuit and $P$ is a set of examples from $\mathbb{Q}^n \times \{0, 1\}$, with relative error at most $(68n_1 2^{n_1} + 136n_1^3 + 136n_1^2 + 170n_1)^{-1}$.

**Correlated Architecture and Training Set Size** The above training setting may be unrealistic in practical applications where one would allow larger architectures if a large amount of data is to be trained. One strategie would be to choose the size of the architecture such that valid generalization can be expected using well known bounds in the PAC setting [26]. Naturally the question arises about what happens to the complexity of training if one is restricted to situations where the number of examples is limited with respect to the number of hidden nodes. One extreme position would be to allow the number of training examples to be at most equal to the number of hidden nodes. Although this may not yield valid generalization, the decision version of the loading problem becomes trivial because of [25], or, more precisely:

If the number of hidden nodes in the first hidden layer is at least equal to the number of training examples and the threshold activation function, the standard sigmoidal function, or the semilinear activation function (or any function $\sigma$ such that the class of $\sigma$-circuits possesses the universal approximation capability as defined in *[25]*) is used then the error of an optimum solution of the loading problem is determined by the number of contradictory training examples (i.e. $(x; y_1)$ and $(x; y_2)$ with $y_1 \neq y_2$.)

However, the following theorem yields an inapproximability result even if we restrict to situations where the number of examples and hidden nodes are correlated.

**Theorem 3.** *Approximtion of the success ratio function $m_L$ with relative error smaller than $c/k^3$ ($c$ is a constant, $k$ is the number of hidden nodes) is NP-hard for the loading problem with instances $(\mathcal{A}, P)$ where $\mathcal{A}$ is a layered $(n, k, 1)$-$H$-architecture ($n$ and $k$ may vary) and $P \subset \mathbb{Q}^n \times \{0, 1\}$ is an example set with $k^{3.5} \leq |P| \leq k^4$ which can be loaded without errors.*

*Proof.* The proof is via L-reduction from the MAX-3-cut problem with $a$ and $\beta$ depending on $k$. The algorithms $T_1$ and $T_2$, respectively, will be defined in two steps: mapping an instance of the MAX-3-cut problem to an instance of the MAX-$k$-cut problem with appropriate $k$ and size of the problem and to an instance of the loading problem, afterwards, or mapping a solution for the loading problem to a solution of the MAX-$k$-cut problem and then to a solution of the MAX-3-cut problem afterwards, respectively.

We first define $T_1$: given a graph $(V, E)$ define $k = |V| \cdot |E|$ (w.l.o.g. $k \geq 3$) and $(V', E')$ with $V' = V \cup \{v_{|V|+1}, \dots, v_{|V|+k-3}\}$, $E' = E \cup \{(v_i, v_j) \mid i \in \{|V| + 1, \dots, |V| + k - 3\}, j \in \{1, \dots, |V| + k - 3\} \backslash \{i\}\}$ where the new edges in $E'$ have the multiplicity $2|E|$. Reduce $(V', E')$ to a loading problem for the architecture with $n = |V'| + 3$, $k$ as above, and examples

**(I)** $2|E'|$ copies of the origin $(0^n; 1)$,

**(II)** $d_i$ copies of the point $\boldsymbol{e}_i$, i.e. $(0, \dots, 0, 1, 0, \dots, 0; 0)$ (the $1$ is at the $i$th position from left) for each node $v_i \in V'$ where $d_i$ is the degree of $v_i$,

**(III)** a vector $\boldsymbol{e}_{ij}$ for each edge $(v_i, v_j) \in E'$: $(0, \dots, 0, 1, 0 \dots, 0, 1, 0, \dots, 0; 1)$ (the numbers $1$ are at the $i$th and $j$th positions from left),

**(IV)** $2|E'|$ copies of each of the points $(0^{|V'|}, \boldsymbol{p}^{ij}, 1; 1)$, $(0^{|V'|}, \boldsymbol{n}^{ij}, 1; 0)$, where $\boldsymbol{p}^{ij}$ and $\boldsymbol{n}^{ij}$ are constructed as follows: define the points $\boldsymbol{x}^{ij} = (4(i-1)+j, j(i-1)+4((i-2)+\dots+1))$ for $i \in \{1, \dots, k\}$, $j \in \{1, 2, 3\}$. These $3k$ points have the property that if three of them lie on one line then we can find an $i$ such that the three points coincide with $\boldsymbol{x}^{i1}$, $\boldsymbol{x}^{i2}$, and $\boldsymbol{x}^{i3}$. Now we divide each point into a pair $\boldsymbol{p}^{ij}$ and $\boldsymbol{n}^{ij}$ of points which are obtained by a slight shift of $\boldsymbol{x}^{ij}$ in a direction that is orthogonal to the line $[\boldsymbol{x}^{i1}, \boldsymbol{x}^{i3}]$. Formally, $\boldsymbol{p}^{ij} = \boldsymbol{x}^{ij} + \epsilon \boldsymbol{N}_i$ and $\boldsymbol{n}^{ij} = \boldsymbol{x}^{ij} - \epsilon \boldsymbol{N}_i$, where $\boldsymbol{N}_i$ is a normal vector of the line $[\boldsymbol{x}^{i1}, \boldsymbol{x}^{i3}]$ with a positive second coefficient and $\epsilon$ is a small positive value. $\epsilon$ can be chosen such that the following holds:

> Assume one line separates three pairs $(\boldsymbol{n}^{i_1 j_1}, \boldsymbol{p}^{i_1 j_1})$, $(\boldsymbol{n}^{i_2 j_2}, \boldsymbol{p}^{i_2 j_2})$, and $(\boldsymbol{n}^{i_3 j_3}, \boldsymbol{p}^{i_3 j_3})$, then necessarily $i_1 = i_2 = i_3$.

This property is fulfilled for $\epsilon \leq 1/(24 \cdot k(k-1) + 6)$ due to Proposition 6 of [20], $N$ being a vector of length 1. Consequently, the representation of the points $\boldsymbol{n}^{ij}$ and $\boldsymbol{p}^{ij}$ is polynomial in $n$ and $k$.

Note that the number of points is $k^{3.5} \leq 5|E'| + 12k|E'| \leq k^4$ for large $|V|$. An optimum solution of the instance of the MAX-3-cut problem gives rise to a solution of the instance of the MAX-$k$-cut problem with the same number of monochromatic edges via mapping the nodes in $V \cap V'$ to the same three cuts as before and defining the $i$th cut by $\{v_{|V|+i}\}$ for $i \in \{1, \dots, k-3\}$. This solution can be used to define a solution of the instance of the loading problem as follows: The $j$th weight of node $i$ in the hidden layer is chosen as $\begin{cases} -1 & \text{if } v_j \text{ is in the } i\text{th cut} \\ 2 & \text{otherwise,} \end{cases}$ and the bias is chosen as $0.5$. The weights $(|V'|+1, |V'|+2, |V'|+3)$ of the $i$th node are chosen as $(-i+1, 1, -0.5+2 \cdot i(i-1))$ which corresponds to the line through the points $\boldsymbol{x}^{i1}$, $\boldsymbol{x}^{i2}$, and $\boldsymbol{x}^{i3}$. The output unit has the bias $-k+0.5$ and weights $1$, i.e. it computes an AND. With this choice of weights one can compute that all examples except the points $\boldsymbol{e}_{ij}$ corresponding to monochromatic edges are mapped correctly.

Conversely, an optimum solution of the loading problem classifies all points in **(I)**, **(II)**, and **(IV)** and all points $\boldsymbol{e}_{ij}$ corresponding to edges in $E' \backslash E$ correct because of the multiplicities of the respective points. We can assume that the activations of the nodes do not exactly coincide with $0$ when the outputs on $P$ are computed. Consider the restriction of the circuit mapping to the plane $\{(0,\ldots,0,x_{n+1},x_{n+2},1)\mid x_{n+1},x_{n+2}\in\mathbb{R}\}$. The points $\boldsymbol{p}^{ij}$ and $\boldsymbol{n}^{ij}$ are contained in this plane. Because of the different outputs each pair $(\boldsymbol{p}^{ij},\boldsymbol{n}^{ij})$ is to be separated by at least one line defined by the hidden nodes. A number $3k$ of such pairs exists. Therefore, each of the lines defined by the hidden nodes necessarily separates three pairs $(\boldsymbol{p}^{ij},\boldsymbol{n}^{ij})$ with $j\in\{1,2,3\}$ and nearly coincides with the line defined by $[\boldsymbol{x}^{i1},\boldsymbol{x}^{i3}]$. Denote the output weights of the circuit by $w_1,\ldots,w_k$ and the output bias by $\theta$. We can assume that the $i$th node nearly coincides with the $i$th line and that the points $\boldsymbol{p}^{ij}$ are mapped by the node to the value $0$. Otherwise we change all signs of the weights and the bias in node $i$, we change the sign of the weight $w_i$, and increase $\theta$ by $w_i$. But then the points $\boldsymbol{p}^{i2}$ are mapped to $0$ by all hidden nodes, the points $\boldsymbol{n}^{i2}$ are mapped to $0$ by all but one hidden node. This means that $\theta>0$, $\theta+w_i<0$ for all $i$ and therefore $\theta+w_{i_1}+\ldots+w_{i_l}<0$ for all $i_1,\ldots,i_l\in\{1,\ldots,k\}$ with $l\geq 1$. This means that the output unit computes the function NAND : $(x_1,\ldots,x_n)\mapsto\neg x_1\wedge\ldots\wedge\neg x_n$ on binary values.

Define a solution of the instance of the MAX-$k$-cut problem by setting the $i$th cut $c_i$ as $\{v_j\mid \text{the } i\text{ th hidden node maps } \boldsymbol{e}_j \text{ to } 1\}\backslash(c_1\cup\ldots\cup c_{i-1})$. Assume some edge $(v_i,v_j)$ is monochromatic. Then $\boldsymbol{e}_i$ and $\boldsymbol{e}_j$ are mapped to 1 by the same hidden node. Therefore $\boldsymbol{e}_{ij}$ is classified wrong. Note that all $\boldsymbol{e}_{ij}$ corresponding to edges in $E\backslash E'$ are correct, hence the nodes $v_{|V|+1},\ldots,v_{|V|+k-3}$ each form one cut and the remaining nodes are contained in the remaining three cuts. Hence these three cuts define a solution of the instance of the MAX-3-cut problem such that almost edges corresponding to misclassified $\boldsymbol{e}_{ij}$ are monochromatic.

Denote by $\text{opt}_1$ the value of an optimum solution of the MAX-3-cut problem and by $\text{opt}_2$ the optimum value of the loading problem. We have shown that

$$\text{opt}_2=\frac{|E|\text{opt}_1+(|E'|-|E|)+4|E'|+12|E'|k}{5|E'|+12|E'|k}\leq\frac{3}{2}\,\text{opt}_1\,.$$

Next we construct $T_2$. Assume that a solution of the loading problem with relative error smaller than $c/k^3$ is given. Then the points **(I)** and **(IV)** are correct due to their multiplicities. Otherwise the relative error of the problem would be at least $|E'|/(5|E'|+12|E'|k)\geq c/k^3$ for appropriately small $c$ and large $k$. As before we can assume that the output node computes the function $\boldsymbol{x}\mapsto\neg x_1\wedge\ldots\wedge\neg x_k$. Define $\text{opt}_2$ to be the value of an optimum solution of the loading problem and $I_2$ the value of the given solution. Assume some point $\boldsymbol{e}_{ij}$ corresponding to an edge in $E'\backslash E$ is misclassified. Then $T_2$ yields an arbitrary solution of the MAX-3-cut problem. For the quality $I_1$ of this solution compared to an optimum $\text{opt}_1$ we can compute

$$\text{opt}_1-I_1\leq 1\leq\frac{5|E'|+12|E'|k}{|E|}(\text{opt}_2-I_2)\,.$$

This holds because an optimum solution of the loading problem classifies at least a number of $|E|$ points more correct than in the solution considered here.

If all $\boldsymbol{e}_{ij}$ corresponding to edges in $E'\backslash E$ are correct then we define a solution of the MAX-3-cut problem via the activation of the hidden nodes as above. Remaining nodes become members of the first cut. An argument as above shows that each monochromatic edge comes from a misclassification of either $\boldsymbol{e}_i$, $\boldsymbol{e}_j$, or $\boldsymbol{e}_{ij}$. Hence

$$\mathrm{opt}_1 - I_1 \leq \frac{5|E'| + 12|E'|k}{|E|}(\mathrm{opt}_2 - I_2)\,.$$

Setting $\alpha = 3/2, \beta = \tilde{c} \cdot k^3 \geq (5|E'| + 12|E'|k)/|E|$ for some constant $\tilde{c}$ and using Theorem 1 yields the result as stated above. □

**The $(n, 2, 1)$-$\{$sgd, $H_\epsilon\}$-net** The above result deals with realistic circuit structures. However, usually a continuous and differentiable activation function is used in practice. A very common activation function is the standard sigmoid activation $\mathrm{sgd}(x) = 1/(1 + \mathrm{e}^{-x})$. Here we consider the loading problem with a feedforward architecture of the form $(n, 2, 1)$ where the input dimension $n$ is allowed to vary. The sigmoidal activation function is used in the two hidden nodes. The output is the function

$$H_\epsilon(x) = \begin{cases} 0 & \text{if } x < -\epsilon\,, \\ \text{undefined} & \text{if } -\epsilon \leq x \leq \epsilon\,, \\ 1 & \text{otherwise}\,. \end{cases}$$

The purpose of this definition is to enforce that any classification is performed with a minimum separation accuracy $\epsilon$. Furthermore, we restrict to solutions with output weights whose absolute values are bounded by some positive constant $B$. This setting is captured by the notion of so-called $\epsilon$-separation (for example, see [19]). Formally, the circuit computes the function $\beta_{\mathcal{A}}(\boldsymbol{w}, \boldsymbol{x}) = H_\epsilon(\alpha\,\mathrm{sgd}(\boldsymbol{a}^t\boldsymbol{x} + a_0) + \beta\,\mathrm{sgd}(\boldsymbol{b}^t\boldsymbol{x} + b_0) + \gamma)$ where $\boldsymbol{w} = (\alpha, \beta, \gamma, \boldsymbol{a}, a_0, \boldsymbol{b}, b_0)$ are the weights and thresholds, respectively, of the output node and the two hidden nodes and $|\alpha|, |\beta| < B$ for some positive constant $B$.

**Theorem 4.** *It is NP-hard to approximate the $m_L$ with relative error smaller than $1/2244$ for the architecture of a $\{(n, 2, 1) \mid n \in \mathbb{N}\}$-circuit with sigmoidal activation function for the hidden nodes, output activation function $H_\epsilon$ with $0 < \epsilon < 0.5$, weight restriction $B \geq 2$ of the output weights, and examples from $\mathbb{Q}^n \times \{0, 1\}$.*

The proof consists in an application of Theorem 2 and a careful examination of the geometric form of the classification boundary defined by those types of networks. It turns out that some argumentation can be transferred from the standard perceptron case since some geometrical situations merely correspond to the respective cases for perceptron networks. However, additional geometric situations may take place which are excluded in our setting with appropriate points in the set of special points $P_0$ in near optimum solutions. Due to the situation of $\epsilon$-separation it turns out that the result transfers to more general activation functions:

**Definition 3.** *Two functions $f, g : \mathbb{R} \to \mathbb{R}$ are $\epsilon$-approximates of each other if $|f(x) - g(x)| \leq \epsilon$ holds for all $x \in \mathbb{R}$.*

**Corollary 1.** *It is NP-hard to approximate the success ratio function $m_L$ with relative error smaller than $1/2244$ for $\{(n, 2, 1) \mid n \in \mathbb{N}\}$-circuit architectures with activation function $\sigma$ in the hidden layer and $H_\epsilon$ in the output, $\epsilon < 1/3$, weight restriction $B \geq 2$, and examples from $\mathbb{Q}^n \times \{0, 1\}$, provided $\sigma(x)$ is $\epsilon/(4B)$-approximate to* $\mathrm{sgd}(x)$.

**The $(n, 2, 1)$-$\{$lin, $H\}$-net** In this section, we prove the NP-hardness of the approximability of the success ratio function with the semilinear activation function commonly used in the neural net literature [7, 8]:

$$\text{lin}(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } 0 < x \leq 1 \\ 1 & \text{otherwise} \end{cases} .$$

This function captures the linearity of the sigmoidal activation at $0$ as well as the asymptotic behaviour. Note that the following result does not require $\epsilon$-separation.

**Theorem 5.** *It is NP-hard to approximate $m_L$ with relative error smaller than $1/2380$ for the architecture of $\{(n, 2, 1) \mid n \in \mathbb{N}\}$-circuit with the semilinear activation function in the hidden layer and the threshold activation function in the output.*

Again the proof consists in an application of Theorem 2 and an investigation of the geometrical form of the classification boundaries which enables us to define appropriate algorithms $T_1$ and $T_2$.

**Avoiding Multiplicities** In the reductions of previous sections, examples with multiplicities were contained in the training sets. In the practical relevant case of neural network training, patterns are often subject to noise. Hence the points do not come from a probability distribution with singletons, i.e. points with nonzero probability. As a consequence the question arises as to whether training sets where each point is contained at most once yield NP-hardness results for approximate training as well.

The reduction of the MAX-$k$-cut problem to a loading problem can be modified as follows: $T_1$ yields the *mutually different* points:

- a set $P_0$ of points $p_i^j$, $j = 1, \ldots, 3|E|$ for each $i$,
- for each node $v_i$, points $\boldsymbol{e}_i^j$, $j = 1, \ldots, 2d_i$, where $d_i$ is the degree of $v_i$,
- for each edge $(v_i, v_j)$, two points $\boldsymbol{e}_{ij}$ and $\boldsymbol{o}_{ij}$.

Assume, $T_1$ and $T_2$ satisfy the following properties:

**(i')** For an optimum solution of the MAX-$k$-cut problem one can find an optimum solution of the instance of the corresponding loading problem $L$ in which the special points $P_0$ and all $\boldsymbol{e}_i^j$ points are correctly classified and exactly the monochromatic edges $(v_i, v_j)$ lead to misclassified points $\boldsymbol{e}_{ij}$ or $\boldsymbol{o}_{ij}$.

**(ii')** If for each $i$ at least one $p_l^j$ is correct, $T_2$ computes in polynomial time an approximate solution where, for each monochromatic edge $(v_i, v_j)$, one of the points $\boldsymbol{e}_{ij}$ or $\boldsymbol{o}_{ij}$ or all points $\boldsymbol{e}_i^l$ $(l = 1, \ldots, 3|E|)$ or all points $\boldsymbol{e}_j^l$ $(l = 1, \ldots, 3|E|)$ are misclassified.

An analogous proof to [3] shows the following:

**Theorem 6.** *Under the assumptions stated above, an L-reduction with constants $\alpha = k/(k-1)$, $\beta = 3|P_0| + 6$, and $a = (k-1)/(k^2(3|P_0| + 6))$ arises.*

**Corollary 2.** *The reductions for general perceptron circuits and in Theorems 4 and 5 can be modified such that* **(i')** *and* **(ii')** *hold. Hence minimizing the relative error within some constant is NP-hard even for training sets without multiple points in these situations.*

## 4 Approximating the Failure Ratio Function $m_f$

Given an instance $x$ of the loading problem, denote by $m_C(x, y)$ the number of examples in the training set missclassified by the circuit represented by $y$. Given $c$, we want to find weights such that $\text{opt}_C(x) \leq m_C(x, y) \leq c \cdot \text{opt}_C(x)$. The interesting case is *with errors*, i.e. $\text{opt}_C(x) > 0$. Hence we restrict to the case with errors and investigate if the failure ratio $m_f = m_C(x, y)/\text{opt}_C(x)$ can be bounded from above by a constant. We term this problem as *approximating the minimum failure ratio* within $c$ while learning in the presence of errors [2]. It turns out that the approximation is NP-hard within a bound which is *independent* of the circuit architecture. For this purpose we use a reduction from the set-covering problem.

**Definition 4 (Set Covering Problem [9]).** *Given a set of points $S = \{s_1, \ldots, s_p\}$ and a set of subsets $C = \{C_1, \ldots, C_m\}$, find indices $I \subset \{1, \ldots, m\}$ such that $\bigcup_{i \in I} C_i = S$. In this case the sets $C_i, i \in I$, are called a cover of $S$. A cover is called exact if the sets in a cover are mutually disjoint.*

For the set-covering problem the following result holds, showing that it is hard to approximate within every factor $c > 1$:

**Theorem 7.** [4] *For every $c > 1$ there is a polynomial time reduction that, given an instance $\varphi$ of SAT, produces an instance of the set-covering problem and a number $K \in \mathbb{N}$ with the properties: if $\varphi$ is satisfiable then there exists an exact cover of size $K$, if $\varphi$ is not satisfiable then every cover has size at least $c \cdot K$.*

Using Theorem 7 Arora et.al. [2] show that approximating the minimum failure ratio function within a factor of $c$ (for any constant $c > 1$) is NP-hard for a single threshold node if all the input thresholds are set to zero. We obtain the following result.

**Theorem 8.** *Assume that we are given a layered $H$-circuit where the thresholds of the nodes in the first hidden layer are fixed to $0$ and let $c > 1$ be any given constant. Then the problem of approximating minimum failure ratio $m_f$ while learning in the presence of errors within a factor of $c$ is NP-hard.*

*Proof.* Without loss of generality, assume that the circuit contains at least one hidden layer. Assume that we are given a formula $\varphi$. Transform this formula with the given constant $c$ to an instance $(S = \{s_1, \ldots, s_p\}, C = \{C_1, \ldots, C_m\})$ of the set-covering problem and a constant $K$ such that the properties in Theorem 7 hold. Transform this instance of the set-covering problem to an instance of the loading problem for the given architecture with input dimension $n = |C| + 2 + n_1 + 1$ where $n_1$ denotes the number of hidden nodes in the first hidden layer and the following examples from $\mathbb{Q}^n \times \{0, 1\}$:

**(I)** $(\mathbf{e}_i, 0, 1, 0^{n_1+1}; 1)$, $(-\mathbf{e}_i, 0, 1, 0^{n_1+1}; 1)$, where $\mathbf{e}_i$ is the $i$th unit vector in $\mathbb{R}^{|C|}$,

**(II)** $c \cdot K$ copies of each of the points $(\mathbf{e}_{s_i}, -1, 1, 0^{n_1+1}; 1)$, $(-\mathbf{e}_{s_i}, 1, 1, 0^{n_1+1}; 1)$, where $\mathbf{e}_{s_i} \in \{0, 1\}^{|C|}$ is the vector with $j$th component as 1 if and only if $s_i \in C_j$, $i \in \{1, \ldots, p\}$,

**(III)** $c \cdot K$ copies of each of $(0^{|C|}, 1, 0, 0^{n_1+1}; 1)$, $(0^{|C|}, 1/(2m), 1, 0^{n_1+1}; 1)$, and $(0^{|C|}, -1/(2m), 1, 0^{n_1+1}; 0)$, where the component $|C|+1$ is nonzero in all three points and the component $|C|+2$ is nonzero in the latter two points, $m = |C|$,

**(IV)** $c{\cdot}K$ copies of each of $(0^{|C|+2}, \boldsymbol{p}_i, 1; 0)$, $(0^{|C|+2}, \boldsymbol{p}_0, 1; 1)$, $(0^{|C|+2}, \tilde{\boldsymbol{z}}_i, 1; 1)$, $(0^{|C|+2}, \bar{\boldsymbol{z}}_i, 1; 0)$, where the points $\boldsymbol{p}_i$, $\tilde{\boldsymbol{z}}_i$, $\bar{\boldsymbol{z}}_i$ are constructed as follows: Choose $n_1 + 1$ points in each set $H_i = \{\boldsymbol{x} = (x_1, x_2, \ldots, x_{n_1}) \in \mathbb{R}^{n_1} \,|\, x_i = 0, x_j > 0 \forall j \neq i\}$ (denote the points by $\boldsymbol{z}_1$, $\boldsymbol{z}_2$, ... and the entire set by $Z$) such that any given $n_1 + 1$ different points in $Z$ lie on one hyperplane if and only if they are contained in one $H_i$. For $z_j \in H_i$ define $\tilde{\boldsymbol{z}}_j \in \mathbb{R}^{n_1}$ by $\tilde{\boldsymbol{z}}_j = (z_{j1}, \ldots, z_{ji-1}, z_{ji} + \epsilon, z_{ji+1}, \ldots, z_{jn_1})$, $\bar{\boldsymbol{z}}_j \in \mathbb{R}^{n_1}$ by $\bar{\boldsymbol{z}}_j = (z_{j1}, \ldots, z_{ji-1}, z_{ji} - \epsilon, z_{ji+1}, \ldots, z_{jn_1})$, for some small value $\epsilon$ which is chosen such that the following property holds: if one hyperplane in $\mathbb{R}^{n_1}$ separates at least $n_1 + 1$ pairs $(\tilde{\boldsymbol{z}}_i, \bar{\boldsymbol{z}}_i)$, these pairs coincide with the $n_1 + 1$ pairs corresponding to the $n_1 + 1$ points in some $H_i$, and the separating hyperplane nearly coincides with the hyperplane through $H_i$.

For an exact cover of size $K$, let the corresponding set of indices be $I = \{i_1, \ldots, i_K\}$. Define the weights of a threshold circuit such that the $i$th node in the first hidden layer has the weights $(\mathbf{e}_I, 1, 1/(4m), \boldsymbol{e}_i, 0)$, where the $j$th component of $\mathbf{e}_I \in \{0,1\}^{|S|}$ is 1 if and only if $j \in I$ and $\mathbf{e}_i$ is the $i$th unit vector in $\mathbb{R}^{n_1}$. The remaining nodes in the other layers compute the function $\boldsymbol{x} \mapsto x_1 \wedge \ldots \wedge x_l$ of their inputs $x_i$. Since the cover is exact, this maps all examples correctly except $K$ examples in **(I)**.

Conversely, assume that every cover has size at least $c \cdot K$. Assume some weight setting misclassifies less than $c \cdot K$ examples. We can assume that the activation of every node is different from 0 on the training set: for the examples in **(IV)** the weight $w_n$ serves as a threshold, for the points in **(I)**, **(II)**, and **(III)** except for $(0^{|C|}, 1, 0^{n_1+2}; 1)$ the weight $w_{|C|+2}$ serves as a threshold, hence one can slightly change the respective weight which serves as a threshold without changing the classification of these examples such that the activation becomes nonzero. Assuming that the activation of $(0^{|C|}, 1, 0^{n_1+2}; 1)$ is zero we can slightly increase the weight $w_{|C|+1}$ such that the sign of the activation of all other points which are affected does not change. Because of the multiplicity of the examples the examples in **(II)-(IV)** are correctly classified. We can assume that the output of the circuit has the form $\beta_{\mathcal{A}}(\boldsymbol{w}, \boldsymbol{x}) = f_1(\boldsymbol{x}) \wedge \ldots \wedge f_{n_1}(\boldsymbol{x})$ where $f_i$ is the function computed by the $i$th hidden node in the first hidden layer, because of the points in **(IV)**. This is due to the fact that the points $\tilde{\boldsymbol{z}}_i$ and $\bar{\boldsymbol{z}}_i$ enforce the respective weights of the nodes in the first hidden layer to nearly conincide with weights describing the hyperplane with $i$th coefficient zero. Hence the points $\boldsymbol{p}_i$ are mapped to the entire set $\{0,1\}^{n_1}$ by the hidden nodes in the first hidden layer and determine the remainder of the circuit function. Hence all nodes in the first hidden layer classify all positive examples except less than $c \cdot K$ points of **(I)** correctly and there exists one node in the first hidden layer which classifies the negative example in **(III)** correctly as well. Consider this last node. Denote by $\boldsymbol{w}$ the weights of this node. Because of **(III)**, $w_{|C|+1} > 0$. Define $I = \{i \in \{1, \ldots, |C|\} \,|\, |w_i| \geq w_{|C|+1}/(2m)\}$.

Assume $\{C_i | i \in I\}$ forms a cover. Because of **(III)** we find $w_{|C|+1}/(2m) + w_{|C|+2} > 0$ and $-w_{|C|+1}/(2m) + w_{|C|+2} < 0$. Hence one of the examples in **(I)** is classified wrong for every $i \in I$. Hence at least $c \cdot K$ examples are misclassified.

Assume that $\{C_i \,|\, i \in I\}$ does not form a cover. Then one can find for some $i \leq |S|$ and the point $(\mathbf{e}_{s_i}, -1, 1, 0^{n_1+1})$ in **(II)** an activation $< m \cdot w_{|C|+1}/(2m) - w_{|C|+1} + w_{|C|+2} = w_{|C|+2} - w_{|C|+1}/2$ which is negative because $-w_{|C|+1}/(2m) + w_{|C|+2} < 0$, $w_{|C|+1} > 0$ **(III)**. This yields a misclassified example with multiplicity $c \cdot K$. □

One can obtain an even stronger result indicating that not only approximation within an arbitrary factor is NP hard but even approximation within a factor which is exponential in the input length is not possible unless NP $\subset$ DTIME($n^{\text{poly}(\log n)}$). For this purpose, we use a reduction from the so called label cover problem:

**Definition 5 (Label Cover).** *Given a bipartite graph $G = (V, W, E)$ with $E \subset V \times W$, labels $B$, $D$, and a set $\Pi \subset E \times B \times D$. A* labeling *consists of functions $P : V \to 2^B$ and $Q : W \to 2^D$ which assign labels to the nodes in the graph. The* cost *of a labeling is the number $\sum_{v \in V} |P(v)|$. An edge $e = (v, w)$ is* covered *if both, $P(v)$ and $Q(w)$ are not empty and for all $d \in Q(w)$ some $b \in P(v)$ exists with $(e, b, d) \in \Pi$. A* total cover *is a labeling such that each edge is covered.*

For the set-covering problem the following result holds, showing that it is almost NP-hard to obtain weak approximations:

**Theorem 9.** [2, 18] *For every $\epsilon > 0$ there exists a quasipolynomial time reduction from the satisfiability problem to the label cover problem which maps an instance $\varphi$ of size $n$ to an instance $(G, \Pi)$ of size $N \leq 2^{\textit{poly}(\log n)}$ with the following properties:*
*If $\varphi$ is satisfiable then $(G, \Pi)$ has a total cover with cost $|V|$.*
*If $\varphi$ is not satisfiable then every total cover has cost at least $2^{\log^{0.5-\epsilon} N}|V|$.*
*Furthermore, $(G, \Pi)$ has in both cases the property that for each edge $e = (v, w)$ and $b \in B$ at most one $d \in D$ exists with $(e, b, d) \in \Pi$.*

Via this Theorem and ideas of Arora et.al. [2] the following can be prooved:

**Theorem 10.** *Assume that we are given a layered $H$-circuit where the thresholds of the nodes in the first hidden layer are fixed to $0$ and let $\epsilon > 0$ be any given constant. If the problem of approximating minimum failure ratio $m_f$ while learning in the presence of errors within a factor of $2^{\log^{0.5-\epsilon} N}$, $N$ being the size of the respective input, is polynomial time, then NP $\subset$ DTIME($n^{\textit{poly}(\log n)}$).*

*Proof.* Assume that we are given a formula $\varphi$. Transform this formula with the given constant $\epsilon$ to an instance $(G, \Pi)$ of the label cover problem with the properties as described in Theorem 9. W.l.o.g. does the network contain at least one hidden layer.

First, we delete all $(e = (v, w), b, d)$ in $\Pi$ such that for some edge $e'$ incident to $v$ no $d'$ exists with $(e', b, d') \in \Pi$. Those labels are called *valid*. The costs for a total cover remain $|V|$ if $\varphi$ is satisfiable. Otherwise, this can at most increase the costs. For each $e \in E$ and $b \in B$ a unique $d \in D$ exists such that $(e, b, d) \in \Pi$. We denote this element by $d(e, b)$. We can assume that a total cover exists, since this can be polynomially tested.

Now transform this instance to an instance of the loading problem. The input dimension is $n = n_2 + 2 + n_1 + 1$ where $n_1$ denotes the number of hidden nodes in the first hidden layer, $n_2 = |V||B| + |W||D|$, $E \subset V \times W$ are the edges, $B$ and $D$ are the labels. The following examples from $\mathbb{Q}^n \times \{0, 1\}$ are constructed: ($m = \max\{|B|, |D|\}$, $K = |B| \cdot |E|$, the first $n_2$ components are successively identified with the tupels in $V \times B$ and $W \times D$ and denoted via corresponding indices.)

**(I)** $K$ copies of each of $(0^{n_2+2}, \boldsymbol{p}_i, 1; 0)$ $(i \geq 1)$, $(0^{n_2+2}, \boldsymbol{p}_0, 1; 1)$, $(0^{n_2+2}, \tilde{\boldsymbol{z}}_i, 1; 1)$, $(0^{n_2+2}, \bar{\boldsymbol{z}}_i, 1; 0)$, where the points $\boldsymbol{p}_i$, $\tilde{\boldsymbol{z}}_i$, $\bar{\boldsymbol{z}}_i$ are the same points as in the proof of Theorem 8.

**(II)** $K$ copies of $(0^{|n_2|}, 1, 0, 0^{n_1+1}; 1)$,

**(III)** $K$ copies of $(0^{|n_2|}, 1/(16m^2), 1, 0^{n_1+1}; 1)$, $(0^{|n_2|}, -1/(16m^2), 1, 0^{n_1+1}; 0)$,

**(IV)** $K$ copies of each of the points $(\boldsymbol{e}_v, -1, 1, 0^{n_1+1}; 1)$, $(\boldsymbol{e}_w, -1, 1, 0^{n_1+1}; 1)$, where $\boldsymbol{e}_v$ is $1$ precisely at those places $(v, b)$ such that $b$ is a valid label for $v$ and $0$ otherwise, and $\boldsymbol{e}_w$ is $1$ precisely at the places $(w, d)$ such that $d \in D$ ($v \in V$, $w \in W$).

**(V)** $K$ copies of each of the points $(-\boldsymbol{e}_{v \to w, d}, 1, 1, 0^{n_1+1}; 1)$, where $-\boldsymbol{e}_{v \to w, d}$ is $-1$ precisely at those places $(v, b)$ such that $b$ is a valid label for $v$ and $d$ is not assigned to $(v \to w, b)$ and at the place $(w, d)$ and $0$ otherwise ($v \to w \in E$).

**(VI)** $(-\boldsymbol{e}_{v,b}, 0, 1, 0^{n_1+1}; 1)$, where $-\boldsymbol{e}_{v,b}$ is $-1$ precisely at those places $(v, b)$ such that $b$ is a valid label for $v$.

Assume that a label cover with costs $|V|$ exists. Define the weights for the neurons in the first computation layer by $w_{(v,b)} = 1 \iff b$ is assigned to $v$, $w_{(w,d)} = 1 \iff d$ is assigned to $w$, $w_{n_2+1} = 1$, $w_{n_2+2} = 1/(32m^2)$. If a hidden layer is contained, the remaining coefficients of the $i^{\text{th}}$ hidden neuron in the first hidden layer are defined by $w_{n_2+2+i} = 1$, the remaining coefficients are $0$. The neurons in other layers compute the logical function AND. This maps all points but at most $|V|$ points in **(VI)** to correct outputs. Note that the points in **(V)** are correct since each $v$ is assigned precisely one $b$.

Conversely, assume that a solution of the loading problem is given. We show that it has at least a number of misclassified points which equals the costs of a cover, denoted by $C$. Assume for the sake of contradiction that less than $C$ points are classified wrong. Since a cover has costs at most $K$ we can assume that all points with multiplicities are mapped correctly. Because of the same argumentation as in 8 we can assume that the activation of every node is different from 0 on the training set. Additionally, we can assume that the output of the circuit has the form $\beta_{\mathcal{A}}(\boldsymbol{w}, \boldsymbol{x}) = f_1(\boldsymbol{x}) \wedge \ldots \wedge f_{n_1}(\boldsymbol{x})$ where $f_i$ is the function computed by the $i$th hidden node in the first hidden layer, because of the points in **(I)**. Hence all nodes in the first hidden layer classify all positive examples except less than $C$ points of **(V)** correctly and there exists one node in the first hidden layer which classifies the negative example in **(III)** correctly as well.

Denote by $\boldsymbol{w}$ the weights of this node. Because of **(II)**, $w_{|n_2|+1} > 0$. Label the node $v$ with those valid labels $b$ such that $w_{(v,b)} > w_{n_2+1}/(4m^2)$. Label the node $w$ with those labels $d$ such that $w_{(w,d)} > w_{n_2+1}/(2m)$. If this labeling forms a total cover, then we find for all $b$ assigned to $v$ in **(VI)** an activation smaller than $-w_{n_2+1}/(4m^2) + w_{n_2+2}$. Due to **(III)**, $w_{n_2+2} < 1/(16m^2) \cdot w_{n_2+1}$, hence the activation is smaller than $0$ and leads to a number of misclassified points which is at least equal to the costs $C$.

Assume conversely that this labeling does not form a total cover. Then some $v$ or $w$ is not labeled, or for some label $d$ for $w$ and edge $v \to w$ no $b$ is assigned to $v$ with $(v \to w, b, d) \in \Pi$. Due to **(IV)** we find $\sum_{b \text{ valid for } v} w_{(v,b)} - w_{n_2+1} + w_{n_2+2} > 0$, hence together with **(III)** $\sum_{b \text{ valid for } v} w_{(v,b)} > w_{n_2+1} - w_{n_2+1}/(16m^2)$, hence at least one $w_{(v,b)}$ is of size at least $w_{n_2+1}/(2m)$. In the same way we find $\sum_d w_{(w,d)} - w_{n_2+1} + w_{n_2+2} > 0$, hence at least one $w_{(w,d)}$ is of size at least $w_{n_2+1}/(2m)$. Consequently, each node is assigned some label. Assume that the node $w$ is assigned some $d$ such that the edge $v \to w$ is not covered. Hence $w_{(w,d)} > w_{n_2+1}/(2m)$. Due to **(V)** we find $-\sum_{b \text{ valid for } v,\, d(v \to w,\, b) \neq d} w_{(v,b)} - w_{(w,d)} + w_{n_2+1} + w_{n_2+2} > 0$ and due to **(IV)** we find $\sum_{b \text{ valid for } v} w_{(v,b)} - w_{n_2+1} + w_{n_2+2} > 0$, hence $\sum_{b \text{ valid for } v,\, d(v \to w,\, b) = d} w_{(v,b)} > w_{n_2+1} - w_{n_2+2} - \sum_{b \text{ valid for } v,\, d(v \to w,\, b) \neq d} w_{(v,b)} > w_{n_2+1} - w_{n_2+2} + w_{(w,d)} - w_{n_2+1} - w_{n_2+2} =$

$w_{(w,d)} - 2w_{n_2+2} > w_{n_2+1}(1/(2m) - 1/(8m^2)) > w_{n_2+1}/(4m)$. Hence at least one weight corresponding to a label which can be used to cover this edge is of size at least $w_{n_2+1}/(4m^2)$. □

## 5 Conclusion

We have shown the NP-hardness of finding approximate solutions for the loading problem in several different situations. We have considered the question as to whether approximating the relative error of $m_L$ within a constant factor is NP-hard. Compared to [3] we considered threshold circuits with correlated number of patterns and hidden neurons and the $(n, 2, 1)$-circuit with the sigmoidal (with $\epsilon$-separation) or the semilinear activation function. Furthermore, we discussed how to avoid training using multiple copies of the example. We considered the case where the number of examples is correlated to the number of hidden nodes. Investigating the problem of minimizing the failure ratio in the presence of errors yields NP-hardness within every constant factor $c > 1$ for multi-layer threshold circuits with zero input biases, and even weak approximation of this ratio is hard under standard complexity-theoretic assumptions.

## 6 Acknowledgements

We would like to thank Eduardo Sontag for bringing the authors together, Peter Bartlett and Ben-David Shai for sending us their research work (references [3] and [5]) and pointing out the implications of Theorem 2 of [3], Elizabeth Sweedyk for sending details about reference [2], and NSF for providing financial support for this research.

## References

1. E. Amaldi and V. Kann, The complexity and approximability of finding maximum feasible subsystems of linear relations, Theoretical Computer Science 147 (1-2), pp.181-210, 1995.
2. S. Arora, L. Babai, J. Stern, and Z. Sweedyk, The hardness of approximate optima in lattices, codes and systems of linear equations, Journal of Computer and System Sciences, 54, pp. 317-331, 1997.
3. P. Bartlett and S. Ben-David, Hardness results for neural network approximation problems, to appear in Theoretical Computer Science (conference version in Fischer P. and Simon H. U. (eds.), Computational Learning Theory, Lecture Notes in Artificial Intelligence 1572, Springer, pp. 639-644, 1999).
4. M. Bellare, S. Goldwasser, C. Lund, and A. Russell, Efficient multi-prover interactive proofs with applications to approximation problems, in Proceedings of the 25th ACM Symposium on the Theory of Computing, pp. 113-131, 1993.
5. S. Ben-David, N. Eiron and P. M. Long, On the difficulty of approximately maximizing agreements, 13th Annual ACM Conference on Computational Learning Theory (COLT), 2000.
6. A. Blum and R. L. Rivest, Training a 3-node neural network is NP-complete, Neural Networks 5, pp. 117-127, 1992.
7. J. Brown, M. Garber, and S. Vanable, Artificial neural network on a SIMD architecture, in Proc. 2nd Symposium on the Frontier of Massively Parallel Computation, Fairfax, VA, pp. 43-47, 1988.

8. B. DasGupta, H. T. Siegelmann, and E. D. Sontag, On the Intractability of Loading Neural Networks, in Roychowdhury V. P., Siu K. Y., and Orlitsky A. (eds.), Theoretical Advances in Neural Computation and Learning, Kluwer Academic Publishers, pp. 357-389, 1994.
9. M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness, Freeman, San Franscisco, 1979.
10. B. Hammer, Some complexity results for perceptron networks, in Niklasson L., Bodén M., and Ziemke, T. (eds.), ICANN'98, Springer, pp. 639-644, 1998.
11. B. Hammer, Training a sigmoidal network is difficult, in Verleysen M. (ed.), European Symposium on Artificial Neural Networks, D-Facto publications, pp. 255-260, 1998.
12. K.-U. Höffgen, Computational limitations on training sigmoid neural networks, Information Processing Letters 46(6), pp.269-274, 1993.
13. K.-U. Höffgen, H.-U. Simon, and K. S. Van Horn, Robust trainability of single neurons, Journal of Computer and System Sciences 50(1), pp.114-125, 1995.
14. L. K. Jones, The computational intractability of training sigmoidal neural networks, IEEE Transactions on Information Theory 43(1), pp. 167-713, 1997.
15. J. S. Judd, On the complexity of loading shallow networks, Journal on Complexity 4(3), pp.177-192, 1988.
learning, MIT Press, Cambridge, MA, 1990.
16. J. S. Judd, Neural network design and the complexity of learning, MIT Press, Cambridge, MA, 1990.
17. V. Kann, S. Khanna, J. Lagergren, and A. Panconesi, On the hardness of approximating max-k-cut and its dual, Technical Report CJTCS-1997-2, Chicago Journal of Theoretical Computer Science, 1997.
18. C. Lund and M. Yannakakis, On the hardness of approximate minimization problems, Journal of the ACM, 41(5), pp. 960-981, 1994.
19. W. Maass, G. Schnittger, and E. D. Sontag, A comparison of the computational power of sigmoid versus boolean threshold circuits, in Roychowdhury V. P., Siu K. Y., and Orlitsky A. (eds.), Theoretical Advances in Neural Computation and Learning, Kluwer Academic Publishers, pp. 127-151, 1994.
20. M. Megiddo, On the complexity of holyhedral separability, Discrete Computational Geometry 3, pp. 325-337, 1988.
21. C. H. Papadimtriou and M. Yannakakis. Optimization, Approximation and Complexity Classes, Journal of Computer & System Sciences 43, pp. 425-440, 1991.
22. I. Parberry and G. Schnitger, *Parallel computation with threshold functions*, Journal of Computer and System Sciences, 36, 3 (1988), pp. 278-302.
23. J. Šimà, Back-propagation is not efficient, Neural Networks 9(6), pp. 1017-1023, 1996.
24. K.-Y. Siu, V. Roychowdhury and T. Kailath, *Discrete Neural Computation: A Theoretical Foundation*, Englewood Cliffs, NJ: Prentice Hall, 1994.
25. E. D. Sontag, Feedforward nets for interpolation and classification, Journal of Computer and System Sciences 45, pp.20-48, 1992.
26. M. Vidyasagar, A theory of learning and generalization, Springer, 1997.
27. V. H. Vu, On the infeasibility of training with small squared errors, in Jordan M. I., Kearns M. J., and Solla S. A. (eds.), Advances in Neural Information Processing Systems 10, MIT Press, pp. 371-377, 1998.
28. B. Widrow, R. G. Winter and R. A. Baxter, *Layered neural nets for pattern recognition*, IEEE Transactions on Acoustics, Speech and Signal Processing, 36 (1988), pp. 1109-1117.