Minimizing the Quadratic Training Error of a Sigmoid Neuron Is Hard

Jiří Šíma*

Department of Theoretical Computer Science,
Institute of Computer Science, Academy of Sciences of the Czech Republic,
P. O. Box 5, 182 07 Prague 8, Czech Republic, sima@cs.cas.cz

Abstract. We first present a brief survey of hardness results for training feedforward neural networks. These results are then completed by the proof that the simplest architecture containing only a single neuron that applies the standard (logistic) activation function to the weighted sum of n inputs is hard to train. In particular, the problem of finding the weights of such a unit that minimize the relative quadratic training error within 1 or its average (over a training set) within $13/(31n)$ of its infimum proves to be NP-hard. Hence, the well-known back-propagation learning algorithm appears to be not efficient even for one neuron which has negative consequences in constructive learning.

1 The Complexity of Neural Network Loading

Neural networks establish an important class of learning models that are widely applied in practical applications to solving artificial intelligence tasks [13]. The most prominent position among successful neural learning heuristics is occupied by the *back-propagation* algorithm [31] which is often used for training feedforward networks. This algorithm is based on the gradient descent method that minimizes the quadratic regression error of a network with respect to a training data. For this purpose, each unit (neuron) in the network applies a differentiable activation function (e.g. the *standard logistic sigmoid*) to the weighted sum of its local inputs rather than the discrete *Heaviside (threshold) function* with binary outputs. However, the underlying optimization process appears very time consuming even for small networks and training tasks. This was confirmed by an empirical study of the learning time required by the back-propagation algorithm which suggested its exponential scaling with the size of training sets [35] and networks [36]. Its slow convergence is probably caused by the inherent complexity of training feedforward networks.

The first attempt to theoretically analyze the time complexity of learning by feedforward networks is due to Judd [21] who introduced the so-called *loading problem* which is the problem of finding the weight parameters for a given fixed network architecture and a training task so that the network responses are perfectly consistent with all training data. For example, an efficient loading

* Research supported by grants GA AS CR B2030007 and GA ČR No. 201/00/1489.

algorithm is required for the proper PAC learnability [6] (besides the polynomial VC-dimension that the most common neural network models possess [30, 38]). However, Judd proved the loading problem for feedforward networks to be NP-complete even if very strong restrictions are imposed on their architectures and training tasks [21]. The drawback of Judd's proofs is in using quite unnatural network architectures with irregular interconnection patterns and a fixed input dimension while the number of outputs grows which do not appear in practice. On the other hand, his arguments are valid for practically all the common unit types including the sigmoid neurons. Eventually, Judd provided a polynomial-time loading algorithm for restricted shallow architectures [20] whose practical applicability was probably ruled out by the hardness result for loading deep networks [32]. Further, Parberry proved a similar NP-completeness result for loading feedforward networks with irregular interconnections and only a small constant number of units [27]. In addition, Wiklicky showed that the loading problem for higher-order networks with integer weights is even algorithmically not solvable [39].

In order to achieve the hardness results for common layered architectures with complete connectivity between neighbor layers, Blum and Rivest in their seminal work [5] considered the smallest conceivable two-layer network with only 3 binary neurons (two hidden and one output units) employing the Heaviside activation function. They proved the loading problem for such a 3-node network with n inputs to be NP-complete and generalized the proof for a polynomial number of hidden units (in terms of n) when the output neuron computes logical AND [5]. Hammer further replaced the output AND gate by a threshold unit [11] while Kuhlmann achieved the proof for the output unit implementing any subclass of Boolean functions depending on all the outputs from hidden nodes [23]. Lin and Vitter extended the NP-completeness result even for a 2-node cascade architecture with one hidden unit connected to the output neuron that also receives the inputs [24]. Megiddo, on the other hand, showed that the loading problem for two-layer networks with a *fixed* number of real inputs and the Heaviside hidden nodes, and the output unit implementing an arbitrary Boolean function is solvable in polynomial time [26].

Much effort has been spent to generalize the hardness results also for continuous activation functions, especially for the standard sigmoid used in the back-propagation heuristics for which the loading problem is probably at least algorithmically solvable [25]. DasGupta et al. proved that loading a 3-node network whose two hidden units employ the continuous saturated-linear activation function while the output neuron applies the threshold function for dichotomic *classification* purposes is NP-complete [8]. Further, Höffgen showed the NP-completeness of loading a 3-node network employing the standard activation function for *exact interpolation* but with the severe restriction to binary weights [15]. A more realistic setting as concerns the back-propagation learning was first considered in [33] where loading a 3-node network with two standard sigmoid hidden neurons was proved to be NP-hard although an additional constraint on the weights of the output threshold unit used for binary classification was assumed which is satisfied e.g. when the output bias is zero. Hammer replaced this constraint by requiring the output unit with bounded weights to respond with outputs that are in absolute value greater than a given *accuracy* which excludes a small output interval around zero from the binary classification [12]. This approach also allows to generalize the hardness result for a more general class of activation functions than just the standard sigmoid. On the other hand, there exist activation functions that have still appropriate mathematical properties and for which the feedforward networks are always loadable [34].

Furthermore, the loading problem assumes the correct classification of all training data while in practice one is typically satisfied by the weights yielding a small training error. Therefore, the complexity of *approximately interpolating* a training set with in general *real outputs* by feedforward neural networks has further been studied. Jones considered a 3-node network with n inputs, two hidden neurons employing any monotone Lipschitzian sigmoidal activation function (e.g. the standard sigmoid) and one linear output unit with bounded weights [19]. For such a 3-node network he proved that learning the patterns with real outputs from $[0,1]$ each within a small absolute error $0 < \varepsilon < 1/10$ is NP-hard implying that the problem of finding the weights that minimize the quadratic regression error within a fixed ε of its infimum (or absolutely) is also NP-hard. This NP-hardness proof was generalized for polynomial number k of hidden neurons and a *convex* linear output unit (with zero bias and nonnegative weights whose sum is 1) when the *total* quadratic error is required to be within $1/(16k^5)$ of its infimum (or within $1/(4k^3)$ for the Heaviside hidden units) [19].

In addition, Vu found the *relative* error bounds (with respect to the error infimum) for hard approximate interpolation which are independent on the training set size p by considering the *average* quadratic error that is defined as the total error divided by p. In particular, he proved that it is NP-hard to find weights of a two-layer network with n inputs, k hidden sigmoid neurons (satisfying some Lipschitzian conditions) and one linear output unit with zero bias and positive weights such that for a given training data the relative average quadratic error is within a fixed bound of order $O(1/(nk^5))$ of its infimum [37]. Moreover, for two-layer networks with k hidden neurons employing the Heaviside activations and one sigmoid (or threshold) output unit, Bartlett and Ben-David improved this bound to $O(1/k^3)$ which is even independent on the input dimension [4]. In the case of the threshold output unit used for classification, DasGupta and Hammer proved the same relative error bound $O(1/k^3)$ on the fraction of correctly classified training patterns which is NP-hard to achieve for training sets of size $k^{3.5} \leq p \leq k^4$ related to the number k of hidden units [7]. They also showed that it is NP-hard to approximate this success ratio within a relative error smaller than $1/2244$ for two-layer networks with n inputs, two hidden sigmoid neurons and one output threshold unit (with bounded weights) exploited for the classification with an accuracy $0 < \varepsilon < 0.5$. On the other hand, minimizing the ratio of the number of *misclassified* training patterns within every constant larger than 1 for feedforward threshold networks with zero biases in the first hidden layer is NP-hard [7].

The preceding results suggest that training feedforward networks with *fixed* architectures is hard indeed. However, the possible way out of this situation might be the *constructive learning algorithms* that adapt the network architecture to a particular training task. It is conjectured that for a successful generalization the network size should be kept small, otherwise a training set can easily be wired into the network implementing a look-up table [34]. A constructive learning algorithm usually requires an efficient procedure for minimizing the training error by adapting the weights of only a *single* unit that is being added to the architecture while the weights of remaining units in the network are already fixed (e.g. [9]). Clearly, for a single binary neuron employing the Heaviside activation function the weights that are consistent with a given training data can be found in polynomial time by linear programming provided that they exist (although this problem restricted to binary weights is NP-complete [28] and also to decide whether the Heaviside unit can implement a Boolean function given in a disjunctive or conjunctive normal form is co-NP-complete [14]). Such weights do not often exist but a good approximate solution would be sufficient for constructive learning. However, several authors provided NP-completeness proofs for the problem of finding the weights for a single Heaviside unit so that the number of misclassified training patterns is at most a given constant [16, 29] which remains NP-complete even if the bias is assumed to be zero [1, 18]. In addition, this issue is also NP-hard for a fixed error that is a constant multiple of the optimum [3].

Hush further generalized these results for a single *sigmoid* neuron by showing that it is NP-hard to minimize the training error under the L_1 norm strictly within 1 of its infimum [17]. He conjectured that a similar result holds for the quadratic error corresponding to the L_2 norm which is used in the back-propagation learning. In the present paper this conjecture is proved. In particular, it will be shown that the issue of deciding whether there exist weights of a single neuron employing the standard activation function so that the total quadratic error with respect to a training data is at most a given constant is NP-hard. The presented proof also provides an argument that the problem of finding the weights that minimize the relative quadratic training error within 1 or its average within $13/(31n)$ of its infimum is NP-hard. This implies that the popular back-propagation learning algorithm may be not efficient even for a single neuron and thus has negative consequences in constructive learning. For the simplicity, we will consider only the standard sigmoid in this paper while in the full version we plan to reformulate the theorem for a more general class of sigmoid activation functions.

2 Training a Standard Sigmoid Neuron

In this section the basic definitions regarding a sigmoid neuron and its training will be reviewed. A single (perceptron) *unit (neuron)* with n real inputs $x_1, \ldots, x_n \in \mathbb{R}$ first computes its real *excitation*

$$\xi = w_0 + \sum_{i=1}^{n} w_i x_i \tag{1}$$

where $\boldsymbol{w} = (w_0, \ldots, w_n) \in \mathrm{I\!R}^{n+1}$ is the corresponding real *weight* vector including a *bias* w_0. The *output* y is then determined by applying a nonlinear activation function σ to its excitation:

$$y = \sigma(\xi)\,. \tag{2}$$

We fix σ to be the *standard (logistic) sigmoid*:

$$\sigma(\xi) = \frac{1}{1 + e^{-\xi}} \tag{3}$$

which is employed in the widely used *back-propagation* learning heuristics. Correspondingly, we call such a neuron the *standard sigmoid unit*.

Furthermore, a *training set*

$$T = \{(\boldsymbol{x}_k, d_k);\ \boldsymbol{x}_k = (x_{k1}, \ldots, x_{kn}) \in \mathrm{I\!R}^n,\ d_k \in [0,1],\ k = 1, \ldots, p\} \tag{4}$$

is introduced containing p pairs—*training patterns*, each composed of an n-dimensional real input $\boldsymbol{x}_k$ and the corresponding desired scalar output value d_k from $[0,1]$ to be consistent with the range of activation function (3). Given a weight vector $\boldsymbol{w}$, the *quadratic training error*

$$E_T(\boldsymbol{w}) = \sum_{k=1}^{p} \left(y(\boldsymbol{w}, \boldsymbol{x}_k) - d_k\right)^2 = \sum_{k=1}^{p} \left(\sigma\left(w_0 + \sum_{i=1}^{n} w_i x_{ki}\right) - d_k\right)^2 \tag{5}$$

of a neuron with respect to the training set T is defined as the difference between the actual outputs $y(\boldsymbol{w}, \boldsymbol{x}_k)$ depending on the current weights $\boldsymbol{w}$ and the desired outputs d_k over all training patterns $k = 1, \ldots, p$ measured by the L_2 regression norm. The main goal of learning is to minimize the training error (5) in the weight space. The decision version for the problem of minimizing the error of a neuron employing the standard sigmoid activation function with respect to a given training set is formulated as follows:

Minimum Sigmoid-Unit Error (MSUE)
Instance: A training set T and a positive real number $\varepsilon > 0$.
Question: Is there a weight vector $\boldsymbol{w} \in \mathrm{I\!R}^{n+1}$ such that $E_T(\boldsymbol{w}) \leq \varepsilon$?

3 Minimizing the Training Error Is Hard

In this section the main result that training even a single standard sigmoid neuron is hard will be proved:

Theorem 1. *The problem MSUE is NP-hard.*

Proof. In order to achieve the NP-hardness result, a known NP-complete problem will be reduced to the MSUE problem in polynomial time. In particular, the following *Feedback Arc Set* problem is employed which is known to be NP-complete [22]:

Feedback Arc Set (FAS)
Instance: A directed graph $G = (V, A)$ and a positive integer $a \leq |A|$.
Question: Is there a subset $A' \subseteq A$ containing at most $a \geq |A'|$ directed edges such that the graph $G' = (V, A \setminus A')$ is acyclic?

The FAS problem was also exploited for a corresponding result concerning the Heaviside unit [29]. However, the reduction is adapted here for the standard sigmoid activation function and its verification substantially differs.

Given a FAS instance $G = (V, A)$, a, a corresponding graph $G_r = (V_r, A_r)$ is first constructed so that every directed edge $(u, v) \in A$ in G is replaced by five parallel oriented paths

$$P_{(u,v),h} = \{(u, u_v), (u_v, u_{vh1}), (u_{vh1}, u_{vh2}), \ldots, (u_{vh,r-1}, v)\} \tag{6}$$

for $h = 1, \ldots, 5$ in G_r sharing *only* their first edge (u, u_v) and vertices u, u_v, v. Each path $P_{(u,v),h}$ includes

$$r = 8a + 6 \tag{7}$$

additional vertices $u_v, u_{vh1}, u_{vh2}, \ldots, u_{vh,r-1}$ unique to $(u, v) \in A$, i.e. the subsets of edges

$$A_{(u,v)} = \bigcup_{h=1}^{5} P_{(u,v),h} \tag{8}$$

corresponding to different $(u, v) \in A$ are pairwise disjoint. Thus,

$$\begin{aligned} V_r = V \cup \{u_v;\, (u, v) \in A\} & \\ \cup \{u_{vh1}, u_{vh2}, \ldots, u_{vh,r-1};\, (u, v) \in A,\, h = 1, \ldots, 5\} & \qquad (9) \\ A_r = \bigcup_{(u,v)\in A} A_{(u,v)} \,. & \qquad (10) \end{aligned}$$

It follows that $n = |V_r| = |V| + (5r - 4)|A|$ and $s = |A_r| = (5r + 1)|A|$. Obviously, the FAS instance G, a has a solution iff the FAS problem is solvable for G_r, a. The graph G_r is then exploited for constructing the corresponding MSUE instance with a training set $T(G)$ for the standard sigmoid unit with $n = (40a + 26)|A| + |V| = O(|A|^2 + |V|)$ inputs:

$$\begin{aligned} T(G) = \big\{ \big(\boldsymbol{x}_{(i,j)}, 1\big), \big(-\boldsymbol{x}_{(i,j)}, 0\big) \,;\, (i, j) \in A_r, & \\ \boldsymbol{x}_{(i,j)} = (x_{(i,j),1}, \ldots, x_{(i,j),n}) \in \{-1, 0, 1\}^n \big\} & \qquad (11) \end{aligned}$$

that contains $p = 2s = (80a + 62)|A| = O(|A|^2)$ training patterns, for each edge $(i, j) \in A_r$ one pair $(\boldsymbol{x}_{(i,j)}, 1), (-\boldsymbol{x}_{(i,j)}, 0)$ such that

$$x_{(i,j),\ell} = \begin{cases} -1 & \text{for } \ \ell = i \\ 1 & \text{for } \ \ell = j \\ 0 & \text{for } \ \ell \neq i, j \end{cases} \qquad \ell = 1, \ldots, n\,, \qquad (i, j) \in A_r\,. \tag{12}$$

In addition, the error in the MSUE instance is required to be at most

$$\varepsilon = 2a + 1\,. \tag{13}$$

Clearly, the present construction of the corresponding MSUE instance can be achieved in a polynomial time in terms of the size of the original FAS instance.

Now, the correctness of the reduction will be verified, i.e. it will be shown that the MSUE instance has a solution iff the corresponding FAS instance is solvable. So first assume that there exists a weight vector $\boldsymbol{w} \in \mathbb{R}^{n+1}$ such that

$$E_{T(G)}(\boldsymbol{w}) \leq \varepsilon \,. \tag{14}$$

Define a subset of edges

$$A' = \{(u,v) \in A;\, w_u \geq w_v\} \subseteq A \tag{15}$$

in G. First observe that graph $G' = (V, A \setminus A')$ is acyclic since each vertex $u \in V \subseteq V_r$ is evaluated by a real weight $w_u \in \mathbb{R}$ so that any directed edge $(u,v) \in A \setminus A'$ in G' satisfies $w_u < w_v$.

Moreover, it must be checked that $|A'| \leq a$. For this purpose, the error $E_{T(G)}(\boldsymbol{w})$ introduced in (5) is expressed for the training set $T(G)$ by using (12) and (10) as follows:

$$\begin{aligned} E_{T(G)}(\boldsymbol{w}) &= \sum_{(i,j)\in A_r} (\sigma(w_0 - w_i + w_j) - 1)^2 + \sum_{(i,j)\in A_r} \sigma^2(w_0 + w_i - w_j) \\ &= \sum_{(u,v)\in A} \sum_{(i,j)\in A_{(u,v)}} \left(\sigma^2(-w_0 + w_i - w_j) + \sigma^2(w_0 + w_i - w_j)\right) \end{aligned} \tag{16}$$

where the property $\sigma(-\xi) = 1 - \sigma(\xi)$ of the standard sigmoid (3) is employed. This error is lower bounded by considering only the edges from $A' \subseteq A$:

$$E_{T(G)}(\boldsymbol{w}) \geq \sum_{(u,v)\in A'} EA_{(u,v)} \tag{17}$$

where each term $EA_{(u,v)}$ for $(u,v) \in A'$ will below be proved to satisfy

$$EA_{(u,v)} = \sum_{(i,j)\in A_{(u,v)}} \left(\sigma^2(-w_0 + w_i - w_j) + \sigma^2(w_0 + w_i - w_j)\right) > \frac{\varepsilon}{a+1} \,. \tag{18}$$

Clearly, e.g. $w_0 \geq 0$ can here be assumed without loss of generality. For each $(u,v) \in A'$ let $P_{(u,v)}$ be a path with the minimum error

$$EP_{(u,v)} = \sum_{(i,j)\in P_{(u,v)}} \left(\sigma^2(-w_0 + w_i - w_j) + \sigma^2(w_0 + w_i - w_j)\right) \tag{19}$$

among paths $P_{(u,v),h}$ for $h = 1, \ldots, 5$. Furthermore, sort the edges $(i,j) \in P_{(u,v)}$ with respect to associated *decrements* $w_i - w_j$ in nonincreasing order and denote by $(c,d),(e,f) \in P_{(u,v)}$ the first two edges, respectively, in the underlying sorted sequence, i.e. $w_c - w_d \geq w_e - w_f \geq w_i - w_j$ for all $(i,j) \in P_{(u,v)} \setminus \{(c,d),(e,f)\}$.

First consider the case when $w_0+w_e-w_f \geq \ln 2$, i.e. $\sigma^2(w_0+w_e-w_f) \geq 4/9$ according to (3). It follows from definition of $P_{(u,v)}$ and (8) that

$$\begin{aligned}
EA_{(u,v)} &\geq \sum_{(i,j)\in A_{(u,v)}\setminus\{(u,u_v)\}} \left(\sigma^2(-w_0+w_i-w_j)+\sigma^2(w_0+w_i-w_j)\right) \\
&\geq 5 \cdot \sum_{(i,j)\in P_{(u,v)}\setminus\{(u,u_v)\}} \left(\sigma^2(-w_0+w_i-w_j)+\sigma^2(w_0+w_i-w_j)\right) \\
&\geq 5\cdot\sigma^2(w_0+w_e-w_f) \geq \frac{20}{9} > \frac{\varepsilon}{a+1} \qquad (20)
\end{aligned}$$

since $P_{(u,v)}\setminus\{(u,u_v)\}$ contains an edge $(i,j)\in\{(c,d),(e,f)\}$ with $\sigma^2(w_0+w_i-w_j) \geq \sigma^2(w_0+w_e-w_f)$ by definition of (e,f) due to σ^2 is increasing. This proves inequality (18) for $w_0+w_e-w_f \geq \ln 2$.

On the other hand suppose that $w_0+w_e-w_f < \ln 2$. In this case vertices $i \in V_r$ on path $P_{(u,v)}$ ($(u,v)\in A'$) will possibly be re-labeled with new weights $w_i' \in \mathbb{R}$ except for fixed w_u, w_v so that there is at most one edge $(c,d)\in P_{(u,v)}$ with a positive decrement $w_c-w_d>0$ or all the edges $(i,j)\in P_{(u,v)}$ are associated with nonnegative decrements $w_i-w_j \geq 0$ while the error $EP_{(u,v)}$ introduced in (19) is not increased. Note that error $EP_{(u,v)}$ depends only on decrements w_i-w_j rather than on the actual weights w_i, w_j. For example, these decrements can arbitrarily be permuted along path $P_{(u,v)}$ producing new weights whereas $EP_{(u,v)}$ and w_u, w_v do not change. Recall from definition of $(c,d),(e,f)$ that for all $(i,j)\in P_{(u,v)}\setminus\{(c,d)\}$ it holds

$$-w_0+w_i-w_j \leq w_0+w_i-w_j \leq w_0+w_e-w_f < \ln 2\,. \qquad (21)$$

Now, suppose that there exists an edge $(i,j)\in P_{(u,v)}\setminus\{(c,d)\}$ with a positive decrement $0<w_i-w_j\leq w_c-w_d$ together with an edge $(\ell,m)\in P_{(u,v)}$ associated with a negative decrement $w_\ell-w_m<0$. Then these decrements are updated as follows:

$$w_i'-w_j' = w_i-w_j-\Delta \qquad (22)$$
$$w_\ell'-w_m' = w_\ell-w_m+\Delta \qquad (23)$$

where $\Delta=\min(w_i-w_j, w_m-w_\ell)>0$. This can be achieved e.g. by permuting the decrements along path $P_{(u,v)}$ so that w_i-w_j follows immediately after $w_\ell-w_m$ (this produces new weights but preserves $EP_{(u,v)}$) and by decreasing the weight of the middle vertex that is common to both decrements by Δ which clearly influences error $EP_{(u,v)}$. However, for $\xi<\ln 2$ the first derivative $(\sigma^2)'$ is increasing because

$$\left(\sigma^2(\xi)\right)'' = \frac{2e^{-\xi}\left(2e^{-\xi}-1\right)}{\left(1+e^{-\xi}\right)^4} > 0 \qquad (24)$$

for $\xi<\ln 2$ according to (3). Hence,

$$\sigma^2(-w_0+w_i-w_j)+\sigma^2(-w_0+w_\ell-w_m) > \sigma^2(-w_0+w_i-w_j-\Delta) + \sigma^2(-w_0+w_\ell-w_m+\Delta) \quad (25)$$

$$\sigma^2(w_0+w_i-w_j)+\sigma^2(w_0+w_\ell-w_m) > \sigma^2(w_0+w_i-w_j-\Delta) + \sigma^2(w_0+w_\ell-w_m+\Delta) \quad (26)$$

according to (21). This implies that error $EP_{(u,v)}$ only decreases while $w'_i - w'_j = 0$ or $w'_\ell - w'_m = 0$. By repeating this re-labeling procedure eventually at most one positive decrement $w_c - w_d > 0$ remains or all the negative decrements are eliminated.

Furthermore,

$$w_c - w_d + \sum_{(i,j)\in P_{(u,v)}\setminus\{(c,d)\}} (w_i - w_j) = w_u - w_v \geq 0 \quad (27)$$

due to $(u,v) \in A'$ which implies

$$w_d - w_c \leq \sum_{(i,j)\in P_{(u,v)}\setminus\{(c,d)\}} (w_i - w_j)\,. \quad (28)$$

Thus,

$$w_d - w_c \leq w_i - w_j \quad (29)$$

can be assumed for all $(i,j) \in P_{(u,v)}$ since the decrements $w_i - w_j$ for $(i,j) \in P_{(u,v)} \setminus \{(c,d)\}$ in sum (28) can be made all nonpositive or all nonnegative. According to (29) inequality (18) would follow from

$$EA_{(u,v)} \geq EP_{(u,v)} \geq \sigma^2(-w_0+w_c-w_d) + r\cdot\sigma^2(-w_0+w_d-w_c) + \sigma^2(w_0+w_c-w_d) + r\cdot\sigma^2(w_0+w_d-w_c) > \frac{\varepsilon}{a+1} \quad (30)$$

because there are r edges (i,j) on path $P_{(u,v)}$ except for (c,d) and σ^2 is increasing. The particular terms of addition (30) can suitably be coupled so that it suffices to show

$$\sigma^2(\xi) + r\cdot\sigma^2(-\xi) > \frac{\varepsilon}{2(a+1)} \quad (31)$$

for any excitation $\xi \in \mathbb{R}$. For this purpose, a boundary excitation

$$\xi_b = \ln\left(\frac{\varepsilon + \sqrt{2\varepsilon(a+1)}}{2(a+1)-\varepsilon}\right) = \ln\left(2a+1+\sqrt{4a^2+6a+2}\right) \quad (32)$$

is derived from (3), (13) such that

$$\sigma^2(\xi_b) = \frac{\varepsilon}{2(a+1)}\,. \quad (33)$$

Thus, $\sigma^2(\xi) > \sigma^2(\xi_b)$ for $\xi > \xi_b$ due to σ^2 is increasing which clearly implies (31) for $\xi > \xi_b$ according to (33). For $\xi \leq \xi_b$, on the other hand, it will even be proved that

$$\sigma^2(\xi) + r\cdot\sigma^2(-\xi) \geq 1 > \frac{\varepsilon}{2(a+1)} \quad (34)$$

which reduces to

$$r \geq \frac{1-\sigma^2(\xi)}{\sigma^2(-\xi)} = 2e^{\xi} + 1 \tag{35}$$

by using (3). Moreover, it is sufficient to verify (35) only for $\xi = \xi_b$, i.e.

$$r \geq 2e^{\xi_b} + 1 \tag{36}$$

since $2e^{\xi}+1$ is increasing. Inequality (36) can be checked by substituting (7) for r and (32) for ξ_b which completes the argument for (30) and consequently for (18).

Finally, by introducing (14) and (18) into inequality (17) it follows that

$$\varepsilon \geq E_{T(G)}(\boldsymbol{w}) > |A'| \cdot \frac{\varepsilon}{a+1} \tag{37}$$

which gives $|A'| < a+1$ or equivalently $|A'| \leq a$. This completes the proof that A' is a solution of the FAS problem.

On the other hand, assume that there exists a solution $A' \subseteq A$ of the FAS instance containing at most $a \geq |A'|$ directed edges making graph $G' = (V, A \setminus A')$ acyclic. Define a subset

$$A'_r = \{(u, u_v);\ (u, v) \in A'\} \tag{38}$$

containing $|A'_r| = |A'| \leq a$ edges from A_r. Clearly, graph $G'_r = (V_r, A_r \setminus A'_r)$ is also acyclic and hence its vertices $i \in V_r$ can be evaluated by *integers* w'_i so that any directed edge $(i,j) \in A_r \setminus A'_r$ satisfies $w'_i < w'_j$. Now, the corresponding weight vector $\boldsymbol{w}$ is defined as

$$w_i = K \cdot w'_i \tag{39}$$

for $i \in V_r$ where $K > 0$ is a sufficiently large positive constant, e.g.

$$K = \ln\left(\sqrt{p} - 1\right) = \ln\left(\sqrt{2s} - 1\right) \tag{40}$$

(recall $p = |T(G)| = 2s$ where $s = |A_r|$) while $w_0 = 0$ which will be proved to be a solution for the MSUE instance. The error (16) can be rewritten for $\boldsymbol{w}$:

$$\begin{aligned} E_{T(G)}(\boldsymbol{w}) &= \sum_{(i,j)\in A_r} 2\sigma^2(w_i - w_j) \\ &= 2\sum_{(i,j)\in A'_r} \sigma^2(w_i - w_j) + 2\sum_{(i,j)\in A_r \setminus A'_r} \sigma^2(w_i - w_j)\,. \end{aligned} \tag{41}$$

For $(i,j) \in A_r \setminus A'_r$ it holds

$$w_i - w_j = K(w'_i - w'_j) \leq -K < 0 \tag{42}$$

according to (39) where $w'_i - w'_j \leq -1$ due to w'_i, w'_j are integers. This implies

$$\sigma^2(w_i - w_j) \leq \sigma^2(-K) = \frac{1}{2s} \tag{43}$$

for $(i,j) \in A_r \setminus A'_r$ by formulas (3), (40) due to σ^2 is increasing. Hence, the error (41) can be upper bounded as

$$E_{T(G)}(\boldsymbol{w}) \leq 2\,|A'_r| + 2s\,\sigma^2(-K) \leq 2a + 1 = \varepsilon \tag{44}$$

by using $|A'_r| \leq a$, $\sigma^2(\xi) < 1$, and $|A_r \setminus A'_r| \leq s$. Therefore $\boldsymbol{w}$ is a solution of the MSUE problem. This completes the proof of the theorem. □

The proof of Theorem 1 also provides the NP-hardness result regarding the relative (average) error bounds:

Corollary 2. *Given a training set T containing $p = |T|$ training patterns, it is NP-hard to find a weight vector $\boldsymbol{w} \in \mathbb{R}^{n+1}$ of the standard sigmoid neuron with n inputs for which the quadratic error $E_T(\boldsymbol{w})$ with respect to T is within 1 of its infimum, or the average quadratic error $E_T(\boldsymbol{w})/p$ is within $13/(31n)$ of its infimum.*

Proof. Given a FAS instance $G = (V, A)$, a, a corresponding MSUE instance $T(G)$, ε is constructed according to (11), (13) in polynomial time. Assume that a weight vector $\boldsymbol{w}^* \in \mathbb{R}^{n+1}$ could be found such that

$$E_{T(G)}(\boldsymbol{w}^*) \leq \inf_{\boldsymbol{w} \in \mathbb{R}^{n+1}} E_{T(G)}(\boldsymbol{w}) + 1\,. \tag{45}$$

The corresponding subset of edges $A^* \subseteq A$ making graph $G^* = (V, A \setminus A^*)$ acyclic can be then read from $\boldsymbol{w}^*$ according to (15). It will be proved in the following that $|A^*| \leq a$ iff the original FAS instance has a solution. This means that finding the weight vector $\boldsymbol{w}^*$ that satisfies (45) is NP-hard.

It suffices to show that for $|A^*| \geq a + 1$ there is no subset $A' \subseteq A$ such that $|A'| \leq a$ and $G' = (V, A \setminus A')$ is acyclic since the opposite implication is trivial. On the contrary suppose that such a subset A' exists. It follows from (45), (37), and (13) that

$$\inf_{\boldsymbol{w} \in \mathbb{R}^{n+1}} E_{T(G)}(\boldsymbol{w}) \geq E_{T(G)}(\boldsymbol{w}^*) - 1 > |A^*| \cdot \frac{2a+1}{a+1} - 1 \geq 2a\,. \tag{46}$$

On the other hand, a weight vector $\boldsymbol{w}' \in \mathbb{R}^{n+1}$ corresponding to subset $A' \subseteq A$ could be defined by (39) that would lead to an error

$$E_{T(G)}(\boldsymbol{w}') \leq 2a + 2s\,\sigma^2(-K) \tag{47}$$

according to (44). However, from (3), (5), and (46) there exists $K > 0$ such that

$$2s\,\sigma^2(-K) < -2a + \inf_{\boldsymbol{w} \in \mathbb{R}^{n+1}} E_{T(G)}(\boldsymbol{w}) < p = 2s \tag{48}$$

which provides a contradiction $E_{T(G)}(\boldsymbol{w}') < \inf_{\boldsymbol{w} \in \mathbb{R}^{n+1}} E_{T(G)}(\boldsymbol{w})$ by using (47).

Finally, it follows from the underlying reduction that approximating the average quadratic error $E_T(\boldsymbol{w})/p$ within $13/(31n)$ of its infimum is also NP-hard due to $p < 31n/13$. □

4 Conclusions

The hardness results for loading feedforward networks are completed by the proof that the approximate training of only a single sigmoid neuron, e.g. by using the popular back-propagation heuristics, is hard. This suggests that the constructive learning algorithms that minimize the training error gradually by adapting unit by unit may also be not efficient. In the full version of the paper we plan to formulate the conditions for a more general class of sigmoid activation functions under which the proof still works.

References

1. E. Amaldi: On the complexity of training perceptrons. In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas (eds.), *Proceedings of the ICANN'91 First International Conference on Artificial Neural Networks*, 55–60, North-Holland, Amsterdam: Elsevier Science Publisher, 1991.
2. M. Anthony and P. L. Bartlett: *Neural Network Learning: Theoretical Foundations.* Cambridge, UK: Cambridge University Press, 1999.
3. 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**(2):317–331, 1997.
4. P. L. Bartlett and S. Ben-David: Hardness results for neural network approximation problems. In P. Fischer and H.-U. Simon (eds.), *Proceedings of the EuroCOLT'99 Fourth European Conference on Computational Learning Theory*, LNAI **1572**, 50–62, Berlin: Springer-Verlag, 1999.
5. A. L. Blum and R. L. Rivest: Training a 3-node neural network is NP-complete. *Neural Networks* **5**(1):117–127, 1992.
6. A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth: Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM* **36**(4):929–965, 1989.
7. B. DasGupta and B. Hammer: On approximate learning by multi-layered feedforward circuits. In H. Arimura, S. Jain, and A. Sharma (eds.), *Proceedings of the ALT'2000 Eleventh International Conference on Algorithmic Learning Theory*, LNAI **1968**, 264–278, Berlin: Springer-Verlag, 2000.
8. B. DasGupta, H. T. Siegelmann, and E. D. Sontag: On the complexity of training neural networks with continuous activation functions. *IEEE Transactions on Neural Networks* **6**(6):1490–1504, 1995.
9. S. E. Fahlman and C. Lebiere: The cascade-correlation learning architecture. In D. S. Touretzky (ed.), *Advances in Neural Information Processing Systems (NIPS'89)*, Vol. **2**, 524–532, San Mateo: Morgan Kaufmann, 1990.
10. M. R. Garey and D. S. Johnson: *Computers and Intractability: A Guide to the Theory of NP-Completeness.* San Francisco: W. H. Freeman, 1979.
11. B. Hammer: Some complexity results for perceptron networks. In L. Niklasson, M. Boden, and T. Ziemke (eds.), *Proceedings of the ICANN'98 Eight International Conference on Artificial Neural Networks*, 639–644, Berlin: Springer-Verlag, 1998.
12. B. Hammer: Training a sigmoidal network is difficult. In M. Verleysen (ed.) *Proceedings of the ESANN'98 Sixth European Symposium on Artificial Neural Networks*, 255–260, D-Facto Publications, 1998.
13. S. Haykin: *Neural Networks: A Comprehensive Foundation.* Upper Saddle River, NJ: Prentice-Hall, 2nd edition, 1999.

14. T. Hegedüs and N. Megiddo: On the geometric separability of Boolean functions, *Discrete Applied Mathematics* **66**(3):205–218, 1996.
15. K.-U. Höffgen: Computational limitations on training sigmoid neural networks. *Information Processing Letters* **46**(6):269–274, 1993.
16. K.-U. Höffgen, H.-U. Simon, and K. S. Van Horn: Robust trainability of single neurons. *Journal of Computer and System Sciences* **50**(1):114–125, 1995.
17. D. R. Hush: Training a sigmoidal node is hard. *Neural Computation* **11**(5):1249–1260, 1999.
18. D. S. Johnson and F. P. Preparata: The densest hemisphere problem. *Theoretical Computer Science* **6**(1):93–107, 1978.
19. L. K. Jones: The computational intractability of training sigmoidal neural networks. *IEEE Transactions on Information Theory* **43**(1):167–173, 1997.
20. J. S. Judd: On the complexity of loading shallow networks. *Journal of Complexity* **4**(3):177–192, 1988.
21. J. S. Judd: *Neural Network Design and the Complexity of Learning.* Cambridge, MA: The MIT Press, 1990.
22. R. M. Karp: Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher (eds.), *Complexity of Computer Computations*, 85–103, New York: Plenum Press, 1972.
23. Ch. Kuhlmann: Hardness results for general two-layer neural networks. In N. Cesa-Bianchi and S. A. Goldman (eds.), *Proceedings of the COLT 2000 Thirteenth Annual Conference on Computational Learning Theory*, 275–285, 2000.
24. J.-H. Lin and J. S. Vitter: Complexity results on learning by neural nets. *Machine Learning* **6**:211–230, 1991.
25. A. Macintyre and E. D. Sontag: Finiteness results for sigmoidal "neural" networks. In *Proceedings of the STOC'93 Twenty-Fifth Annual ACM Symposium on the Theory of Computing*, 325–334, New York: ACM Press, 1993.
26. N. Megiddo: On the complexity of polyhedral separability. *Discrete Computational Geometry* **3**:325–337, 1988.
27. I. Parberry: On the complexity of learning with a small number of nodes. In *Proceedings of the International Joint Conference on Neural Networks*, Vol. **3**, 893–898, 1992.
28. L. Pitt and L. G. Valiant: Computational limitations on learning from examples. *Journal of the ACM* **35**(4):965–984, 1988.
29. V. P. Roychowdhury, K.-Y. Siu, and T. Kailath: Classification of linearly non-separable patterns by linear threshold elements. *IEEE Transactions on Neural Networks* **6**(2):318–331, 1995.
30. V. P. Roychowdhury, K.-Y. Siu, and A. Orlitsky (eds.): *Theoretical Advances in Neural Computation and Learning.* Boston: Kluwer Academic Publishers, 1994.
31. D. E. Rumelhart, G. E. Hinton, and R. J. Williams: Learning representations by back-propagating errors. *Nature* **323**:533–536, 1986.
32. J. Šíma: Loading deep networks is hard. *Neural Computation* **6**(5):842–850, 1994.
33. J. Šíma: Back-propagation is not efficient. *Neural Networks* **9**(6):1017–1023, 1996.
34. E. D. Sontag: Feedforward networks for interpolation and classification. *Journal of Computer and System Sciences* **45**(1):20–48, 1992.
35. G. Tesauro: Scaling relationships in back-propagation learning: dependence on training set size. *Complex Systems* **1**(2):367–372, 1987.
36. G. Tesauro and B. Janssens: Scaling relationships in back-propagation learning. *Complex Systems* **2**(1):39–44, 1988.

37. V. H. Vu: On the infeasibility of training neural networks with small squared errors. In M. I. Jordan, M. J. Kearns, and S. A. Solla (eds.), *Advances in Neural Information Processing Systems (NIPS'97)*, Vol. **10**, 371–377, The MIT Press, 1998.
38. M. Vidyasagar: *A Theory of Learning and Generalization.* London: Springer-Verlag, 1997.
39. H. Wiklicky: The neural network loading problem is undecidable. In J. Shawe-Taylor and M. Anthony (eds.), *Proceedings of the EuroCOLT'93 First Conference on Computational Learning Theory*, 183–192, Oxford: Clarendon Press, 1994.