Bithreshold neurons learning

(Uczenie neuronów dwuprogowych)

Sc.D., Prof. FEDIR GECHE[1], VLADYSLAV KOTSOVSKY[1], Ph.D., Assoc.Prof. ANATOLIY BATYUK Ph.D., Assoc.Prof. ALEXANDER MITSA[1] , PhD KONRAD GROMASZEK[3]

[1] Uzhgorod National University, [2] Lviv National Polytechnic University, [3] Lublin University of Technology, Poland

Artificial neural networks on the base of neural-like computational units have many applications and are intensively used for solving numerous important practical tasks [1]. It should be mentioned that many different models of neuron has been proposed. The one of more important features of these units is the activation function determining their outputs. Historically, the first proposed units had activation functions of threshold type according to developed models of brain cells. Using this type of activation Rosenblatt [2] designed the incremental consistent algorithm for the perceptron learning. The simple proof of it convergence was given by Novikoff [3]. Then Minsky and Papert [4] proved that Rosenblatt's algorithm is inefficient in general case. Peled and Simeone were the first to produce a polynomial time algorithm for the threshold recognition problem [5]. They proposed linear programming approach based on polynomial-time Karmarker's algorithm.

It is well known that the threshold unit is incapable solving many rather easy recognition tasks [4] (e. g. the famous XOR-problem). The using of neurons with more complicated activation functions allowed surmounting this constrain. Historically, the one of the first designed advanced device were multi-threshold neural units [6]. But the efficient learning techniques for multi-threshold neuron based neural networks aren't developed even in the case of one node.

The present paper is devoted to the study of simplest case of multi-threshold units, namely bithreshold neurons. The paper consists of four sections. In the second section we focus our attention on complexity of learning of bithreshold neurons. Our main result is that the learning of bithreshold neurons is NP-complete. In the third section we describe the type of decision lists which represents bithreshold Boolean functions. This question is important in view of the known polynomial learning methods for decision lists [7-9]. Thereby we can separate the subclass of rather "well learnable" bithreshold Boolean functions. In the last section we consider the learning of feedforward neural nets on the base of "smoothed" bithreshold neurons.

Main definitions

The bithreshold neurons with n inputs is defined by a triplet ($\mathbf{w}$, t_1, t_2), where $\mathbf{w} \in \mathrm{R}^n$ is the weight vector and $t_1, t_2 \in \mathrm{R}$ ($t_1 < t_2$) are the thresholds. The neuron output y is defined by

$$y = \begin{cases} a, & \text{if } t_1 < (\mathbf{w},\mathbf{x}) < t_2, \\ b, & \text{otherwise.} \end{cases} \tag{1}$$

We consider neurons with binary ($\{a,b\} = \mathrm{Z}_2$) or bipolar ($\{a,b\} = E_2$) outputs, where $\mathrm{Z}_2 = \{0,1\}$, $E_2 = \{-1,1\}$. If $t_1 = -\infty$, then we obtain the ordinary threshold neuron. The triplet ($\mathbf{w}$, t_1, t_2) is the structure vector of the bithreshold neuron.

The bithreshold neuron with bipolar output performs a classification of R^n by mapping every vector in R^n to a +1 or a –1. Geometrically, the bithreshold neuron has two separating parallel hyperplanes that define its decision region, as opposed to just one separating surface that defined the decision region of the traditional threshold neuron.

Let A is the finite set in the space R^n. Then bithreshold neuron makes such dichotomy (A^+, A^-) of the set A:

$$A^- = \{\mathbf{x} \in A \mid t_1 < (\mathbf{w},\mathbf{x}) < t_2\},\ A^+ = A \setminus A^-.$$

This partition we call a "bithreshold" dichotomy and we call "bithreshold separable" the sets A^+ and A^-. In the most important special case $A = \mathrm{Z}_2^n$ or $A = E_2^n$. We call Boolean function $f(x_1, \ldots$ $\mathrm{Z}_2^n \to \mathrm{Z}_2$ a "bithreshold function", if exists bithreshold neuron the structure ($\mathbf{w}$, t_1, t_2) that $f(\mathbf{x}) = 0 \Leftrightarrow t_1 < (\mathbf{w},\mathbf{x}) < t_2$. Let L denote the set of all n-valued bithreshold Boolean function.

Complexity of synthesis procedure

A polynomial time algorithm is one with running time $O(r^s)$, wh r is the size of input and s, $s \leq 1$ is some fixed integer. The s of an input to an algorithm can be measured in various ways. algorithms working with neurons it is naturally to take as a size input the capacity of learning sample.

We shall show that if the $\mathrm{Z} \neq \mathrm{NP}$ conjecture is true, then d exist a polynomial time verification algorithm checking the possibility of realization of the arbitrary Boolean function on one hreshold unit. The learning of bithreshold Boolean function is complete all the more.

Let C is a class of Boolean function: $C = \{C_n\}_{n \geq 1}$, $n \in$ $C_n \subset \{f \mid f : \mathrm{Z}_2^n \to \mathrm{Z}_2\}$. In the complexity theory the follo problem is well-known.

MEMBERSHIP(C)

Instance: A disjunctive normal form formula φ in n variables.

Question: Does the function f represented by φ belong to C.

Anthony proved [8] that MEMBERSHIP(C) is NP-complete all classes satisfying following properties:

1) for every $f \in C_n$ and arbitrary $i \in \{1,\ldots, n\}$, both functi $f(x_1,\ldots,x_{i-1},1,x_{i+1},\ldots,x_n)$ and $f(x_1,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n)$ belong to C

2) for every $n \in \mathrm{N}$, the identically 1-function belongs to C

3) there exists $k \in \mathrm{N}$ such that $C_k \neq \{f \mid f : \mathrm{Z}_2^k \to \mathrm{Z}_2\}$.

Proposition 1. *The task of verification of the membership to class of bithreshold Boolean functions is NP-complete.*

Proof. We show that class $LBT = \{LBT_n\}_{n>1}$ satisfies condi 1-3. Condition 1 follows from Shannon expansion $f(x_1,\ldots$ $f(x_1,\ldots,x_{n-1},0)\overline{x}_n \vee f(x_1,\ldots,x_{n-1},1)x_n$. If Boolean function $f($ $x_n)$ can be generated on realized on the bithreshold neuron the structure $(\mathbf{w} = (w_1,\ldots,w_{n-1},w_n),t_1,t_2)$, then $f(x_1,\ldots,x$ and $f(x_1,\ldots,x_{n-1},0)$ can be realized on bithreshold ne ns with structures $((w_1,\ldots,w_{n-1}),t_1 - w_n,t_2 - w_n)$ respect $((w_1,\ldots,w_{n-1}),t_1,t_2)$. Condition 2 is evident. Condition 3 follow the fact that if $n > 2$ Boolean function $x_1 \oplus x_2 \oplus \ldots \oplus x_n$ doesn long to LBT_n [10]. Therefore subject to [8] MEMBERSHIP(LB NP-complete.

Proposition 2. *The task of verification of the bithreshold sepa bility of the finite set A^+ and A^- is* NP-*complete even in the c $A^+ \cup A^- \subset \{a,b\}^n$, where $a \in \mathrm{R}$, $b \in \mathrm{R}$ ($a \neq b$) and the we coefficients may be restricted to be from the set $\{-1,+1\}$.*

Proof. We use the results of Blum and Rivest from [11], where shown that the following training problem to be NP-complete:

The 3-Node Network with AND output node restricted so any or all of the weights for one hidden node are required to opposite to *the* corresponding weights of the other and ar all the weights are required to belong to $\{-1,+1\}$, since the known NP-complete problem Set-Splitting [12] can be reduce this task.

It is easy to verify that the arbitrary *dichotomy* (A^+,A^-) is hreshold if and only if it can be realized on neural networ mentioned type. Really, $\mathbf{x} \in A^- \Leftrightarrow (\mathbf{w},\mathbf{x}) < t_2$ and $(-\mathbf{w},\mathbf{x}) <$

and the transformation from the basis $\{a,b\}$ to the basis Z_2 can be made using a standard linear transformation of variables (the same is true for synaptic weights).

Representation of bithreshold Boolean function by decision lists

Decision lists are proposed by Rivest *in* [7]. For many application [7, 8] decision lists are more useful than classical disjunctive or conjunctive normal forms.

Let $K=\{f_1,f_2,...,f_r\}$ is arbitrary *finite* sequence of Boolean functions of *n* variables. A function $f:Z_2^n\to Z_2$ is said to be decision list based on sequence *K* if it can be evaluated using a sequence of *if then else* command as follows, for some fixed $\{c_1, c_2,..., c_r\}$, $(c_i\in Z_2, i=1,...,r)$:

if $f_1(\mathbf{x})=1$ then set $f(\mathbf{x})=c_1$

else if $f_2(\mathbf{x})=1$ then set $f(\mathbf{x})=c_2$

....

else if $f_r(\mathbf{x})=1$ then set $f(\mathbf{x})=c_r$
else set $f(\mathbf{x})=0$.

More formally, a decision list based on *K* is defined by a sequence

$$f=(f_1,c_1),(f_2,c_2),...,(f_r,c_r),$$

where $f_i\in K, c_i\in Z_2, (i=1,2,...,r)$. The values of the function f are defined by

$$f(\mathbf{x})=\begin{cases}c_j, & \text{if} j=\min\{i:f_i(\mathbf{x})=1\} \text{ exists,}\\ 0, & \text{otherwise.}\end{cases}$$

Example. Let $K=\{x_1\bar{x}_3,x_2,\bar{x}_1\}$. The decision list

$$f=(x_1\bar{x}_3,0),(x_2,1),(\bar{x}_1,1).$$

may be thought of as operating in the following way on Z_2^3. First, those points for which $x_1\bar{x}_3$ is true are assigned the value 0: these are $(1,0,0)$, $(1,1,0)$. Next the remaining points for which x_2 is satisfied are assigned the value 1: these are $(0,1,0)$, $(0,1,1)$, $(1,1,1)$. Finally, the remaining points for which $\bar{x}_1$ is true are assigned the value 1: this accounts for $(0,0,0)$, $(0,0,1)$, leaving only $(1,0,1)$, which is assigned value 0. At easy to verify that we obtain the following function $\bar{x}_1\bar{x}_2\vee\bar{x}_1\bar{x}_3\vee x_2x_3$.

The relationship between decision lists and threshold Boolean functions was established in [9]. Antony showed (see [8]) that any 1-decision list (that is, a decision list based over the set *K* of single literals) is the threshold function.

We present the similar result concerning the representation of bithreshold Boolean functions.

Proposition 3. *If the members of the decision list*

$$f=(f_1,c_1),(f_2,c_2),...,(f_{r-1},c_{r-1}),(f_r,c_r)$$

satisfy following conditions:

1) f_i is an arbitrary Boolean function of two variables assigned the value 1 *on two points* $(i=1,2,...,r-1)$;

2) $c_i=1, i=1,2,...,r,$

and the function f_r is bithreshold, then f is the bithreshold Boolean function.

Proof. We use the induction on *r*, that the number of members in the decision list. The base case, $r=1$, is easily seen to be true because every Boolean function of two variables is bithreshold (it is sufficient to verify the realizability of the functions $x\oplus y$ and $x\Leftrightarrow y$, as other 14 functions can be realized on single threshold units). Suppose, as an inductive hypothesis, that our proposition is true for all decision lists of cardinality no more *r*. Let we have the following decision lists $f=(f_1,c_1),(f_2,c_2),...,(f_r,c_r),(f_{r+1},c_{r+1})$ of the length *r* + 1. By the inductive hypothesis the decision list $f'=(f_2,c_2),...,(f_r,c_r),(f_{r+1},c_{r+1})$ defines a bithreshold Boolean function. Let the corresponding bithreshold neuron has structure $(\mathbf{w}',t_1',t_2')$, and let $d=\sum_{i=1}^{n}|w_i'|+|t_1|+|t_2|+1$. From conditions 1)-2) follow that the term (f_1, c_1) can has the following values:

1. (0,1);
2. (1,1);
3. $(x_i,1)$;
4. $(\bar{x}_i,1)$;
5. $(x_i\bar{x}_j\vee x_ix_j,1)$;
6. $(\bar{x}_ix_j\vee x_i\bar{x}_j,1)$.

In the first case let $\mathbf{w}=\mathbf{w}', t_1=t_1', t_2=t_2'$. In the second case let $\mathbf{w}=0, t_1=1, t_2=2$. In the third case let $\mathbf{w}=\mathbf{w}'+d\mathbf{e}_i$, $t_1=t_1', t_2=t_2'$, where $\mathbf{e}_i=(0,...,0,1,0,...,0)$. In the forth case let $\mathbf{w}=\mathbf{w}'-d\mathbf{e}_i$, $t_1=t_1'-d, t_2=t_2'-d$. In the fifth we can as sume $\mathbf{w}=\mathbf{w}'+d\mathbf{e}_i+d\mathbf{e}_j$, $t_1=t_1'+d, t_2=t_2'+d$. In the last case let $\mathbf{w}=\mathbf{w}'+d\mathbf{e}_i-d\mathbf{e}_j$, $t_1=t_1', t_2=t_2'$.

Prove that in each case the decision list *f* is the bithreshold Boolean function realizable on the bithreshold unit with the structure $(\mathbf{w}, t_1, t_2)$. It is evident in two first cases.

In the third case for every $\mathbf{x}=(x_1,...,x_i,...,x_n)\in Z_2^n$ $(\mathbf{w},\mathbf{x})=(\mathbf{w}'+d\mathbf{e}_i,\mathbf{x})=(\mathbf{w}',\mathbf{x})+dx_i$.

If $x_i=1$, then the output value of the decision list is equal to 1 and

$$(\mathbf{w},\mathbf{x})=(\mathbf{w}',\mathbf{x})+d\ge-\sum_{j=1}^{n}|w_j'|+\sum_{j=1}^{n}|w_j'|+|t_2'|+1>t_2'=t_2.$$

Thus, in this case the output value for the bithreshold neuron is equal to one for the decision list. If $x_i=0$ then $(\mathbf{w},\mathbf{x})=(\mathbf{w}',\mathbf{x})$. By the inductive hypothesis the decision list $f'=(f_2,c_2),...,(f_r,c_r),(f_{r+1},c_{r+1})$ is the bithreshold function realizable on the bithreshold neuron with the structure $(\mathbf{w}',t_1',t_2')$. Since $t_1'=t_1, t_2'=t_2$ that in the case $x_i=0$ the output of the bithreshold neuron is identical to the out of the decision list. Thus, the function *f* is realizable on the bithreshold with the structure $(\mathbf{w}, t_1, t_2)$. The proof in case 4 is similar.

Let us consider case 5. Let $\mathbf{x}\in Z_2^n$. If $x_i=0$ and $x_j=0$, then

$$(\mathbf{w},\mathbf{x})=(\mathbf{w}',\mathbf{x})\le\sum_{k=1}^{n}|w_k'|<\sum_{k=1}^{n}|w_k'|+|t_2'|+1\le t_1'+d=t_1.$$

If $x_i=1$ and $x_j=1$, then

$$(\mathbf{w},\mathbf{x})=(\mathbf{w}',\mathbf{x})+2d\ge-\sum_{k=1}^{n}|w_k'|+$$

$$+2d>|t_2'|+\sum_{k=1}^{n}|w_k'|+|t_1'|+|t_2'|+1\ge t_2'+d=t_2.$$

In both cases the output of the bithreshold neuron is equal to 1. It corresponds to the output value of the decision list. If $x_i=1, x_j=0$ or $x_i=0, x_j=1$, then $(\mathbf{w},\mathbf{x})=(\mathbf{w}',\mathbf{x})=d$.

Since $t_1=t_1'+d, t_2=t_2'+d$, then in both cases the output value of bithreshold neuron with the structure $(\mathbf{w}, t_1, t_2)$ is equal to one of the neuron with the structure $(\mathbf{w}',t_1',t_2')$, which by the inductive hypothesis is equal to the output of the decision list. The proof in case 6 can be given by similar reasons.

Corollary 1. *If a Boolean function of n variable can be represented as follows:*

$$f(x_1,...,x_n)=g(x_1,...,x_n)$$

$$\vee x_{i_1}^{\alpha_1}\vee...\vee x_{i_l}^{\alpha_l}\vee x_{j_1}^{\beta_1}x_{k_1}^{\gamma_1}\vee x_{j_1}^{\bar{\beta}_1}x_{k_1}^{\bar{\gamma}_1}\vee...\vee x_{j_m}^{\beta_m}x_{k_m}^{\gamma_m}\vee x_{j_m}^{\bar{\beta}_m}x_{k_m}^{\bar{\gamma}_m}.$$

where $g(x_1,...,x_n)$ is an arbitrary bithreshold Boolean function, $x^1=x, x^0=\bar{x}$, $\alpha_i\in Z_2$ $(i=1,...,l)$, $\beta_j\in Z_2,\gamma_j\in Z_2$ $(j=1,...,m)$, then f is the bithreshold function.

The proof follows from the proposition 3 and the evident fact [8] that if the decision list satisfies $c_i=1, i=1,...,r$, then $f=f_1\vee...\vee f_r$

Corollary 2. *The Boolean function f defined by the following decision list*

$$f=(f_1,1),...,(f_r,1),(x_{r+1}^{\alpha_1},c_{r+1}),...,(x_{r+m}^{\alpha_m},c_{r+m}),$$

where $\alpha_i \in Z_2, c_{r+i} \in Z_2, i = 1,...,m$ is the bithreshold Boolean function if $f_1,...,f_r$ satisfy the conditions of the proposition 3.

The proof follows from the proposition 3 and from [8] (according to the theorem 3.9 the decision list of the following form $(x_{r+1}^{\alpha_1}, c_{r+1}),...,(x_{r+m}^{\alpha_m}, c_{r+m})$ is the threshold and so the bithreshold Boolean function).

Feedforward neural nets with smoothed bithreshold activation function

Let us consider the problem of learning the neural net on the base of bithreshold neurons. As we have shown in section 2 these task is hard even for one neuron. These difficulties can be bypassed in the same way that one for traditional threshold neurons. It is enough to consider the neurons with continuous differentiable activation function. We call it the smoothed bithreshold function. Corresponding neuron can be named smoothed bithreshold neurons. It is possible to consider numerous smoothed analogue of hard bithreshold activation function (1). The ones of simplest are following:

$$y = 1 - 2e^{-x^2}, \tag{2}$$

$$y = \frac{2}{1+e^{-10(x-1)}} - \frac{2}{1+e^{-10(x+1)}} + 1. \tag{3}$$

Its plots are shown on *Fig. 1* (the plot of the function (3) is "closer" to the plot of the hard bithreshold function (1)).

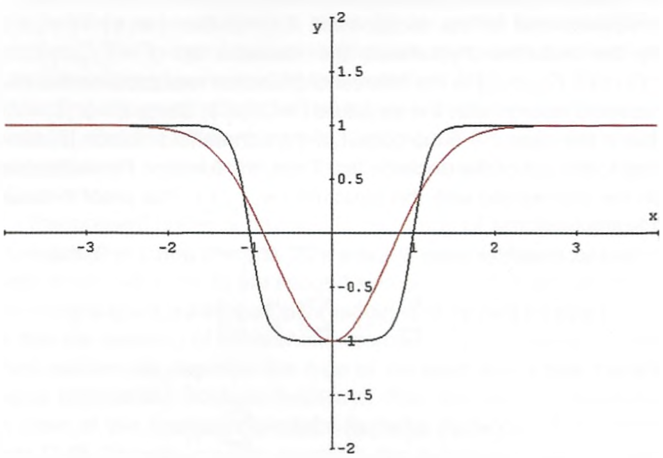

Fig. 1. The plots of the smoothed bithreshold activation functions (2)-(3)
Rys. 1. Wykresy wygładzonych funkcji dwuprogowych

We describe here a fairly simple neural net based on smoothed bithreshold neurons, namely the feedforward net (i.e. the multilayer perceptron). We used backpropagation to learn such nets. The network error and weight corrections are traditional and corresponding formulas are omitted.

Simulation

To compare the performance of feedforward neural nets based on smoothed bithreshold neurons and sigmoid nets we have implemented a simulation tests. We describe results of two typical tests of nets learning in online mode, in which we use the activation function (2) or (3), modified logistic sigmoid $y = \frac{2}{1+e^{-x}} - 1$, $y = \tanh x$ and rational sigmoid $y = \frac{1}{1+|x|}$.

In the first test we learned feedforward 100-10-3 nets (100 inputs, 10 hidden nodes and 3 outputs) for different activation functions on 100 different learning samples, each containing 500 training examples uniformaly distributed in hyperparallelepiped $[-1,1]^{103}$. For every net 1000000 iteration of backpropagation procedure are applied. The learning rate parameter was individualy chosen for every type of activation function.

Tabl. 1. Learning in the case of uniform distributed samples
Tab. 1. Uczenie w przypadku próbek o rozkładzie równomiernym

Activation function	Average total sample error	Maximum error on example
modified logistic	31,27	0,38
tanh *x*	44,81	0,34
rational sigmoid	53,49	0,85
smoothed bithreshold (2)	30,04	0,35

As seen in *Table 1*, the empirical result prove that average total sample error was the least for smoothed bithreshold (2). The maximum error on example for this function is also fine in respect of other functions.

In the second test we trained 100-40-1 feedforward nets to map classical „hard" function XOR of 100 variables (strictly speaking we use the bipolar form of XOR). In the *Table 2* are given the result of computer simulation. The learning sampe size was equal to 1000. For every net 300000 iteration of backpropagation procedure are applied.

Tabl. 2. Learning XOR function
Tab. 2. Uczenie funkcji logicznej XOR

Activation function	Maximum error on example
modified logistic	1,99
tanh *x*	1,99
rational sigmoid	1,87
smoothed bithreshold (3)	0,24

As seen in *Table 2*, learning finished successively only in the case of network based on smoothed bithreshold (3).

Conclusion

Neural-like systems on the base of bithreshold neurons are studied. The hardness of bithreshold neurons learning is established. The conditions are found providing that decision list realizes a bithreshold logic function. The smoothed bithreshold activation functions are proposed. The experimental results confirm how effective developed approach is in learning feedforward neural networks.

References

[1] Haykin S., Neural Networks. A Comprehensive Foundation, Upper Saddle River, NJ: Prentice Hall, Inc., 1999.
[2] Rosenblatt F., Principles of Neurodynamics, Washington, DC: Spartan Books, 1962.
[3] Novikoff A., On convergence proof for perceptrons, Proceeding of Symposium on Mathematical Theory of Automata. Polytechnic Institute of Brooklyn, vol. XII, 1963, 615–622.
[4] Minsky M.L., Papert S.A., Perceptrons, expanded editions, MIT Press, 1988.
[5] Peled U., Simeone B., Polynomial-time algorithm for regular set covering and threshold synthesis, Discrete Applied Mathematics, vol. 12, 1985, 57–69.
[6] Baharyev A.T., Optimization of multithreshold models, Problemy sluchaiynogo poiska, vol. 4, 1975, 209–214 (in Russian).
[7] Rivest R., Learning decision lists, Machine Learning, vol. 2, 1987, 229–246.
[8] Anthony M., Discrete Mathematics of Neural Networks, Philadelphia: SIAM, 2001.
[9] Marchand M., M. Golea., On learning simple neural concepts: From halfspace intersection to neural decision lists, Networks: Computation in Neural Systems, vol. 4, 1993, 67–85.
[10] Geche F., Batyuk A., Kotsovsky W., The properties of Boolean functions realizable on bithreshold neurons, Visnyk nacionalnogo universytetu «Lvivska politehnika» "Kompyuterna inzheneriya ta informaciyni tehnologii", vol. 438, 2001, 22–25 (in Ukrainian).
[11] Blum A., Rivest R., Training a 3-Node Neural Network is NP-Complete, Neural Networks, vol. 5 (1), 1992, 117–127.
[12] Garey M.R., Johnson D.S., Computers and Intractability, New York: W.H. Freeman, 1979.