УДК 004.85

НАВЧАННЯ НЕЙРОМЕРЕЖ ІЗ ФУНКЦІЄЮ АКТИВАЦІЇ ДВОПОРОГОВОГО ТИПУ

*В. М. КОЦОВСЬКИЙ**

Доцент кафедри інформаційних управляючих систем та технологій, ДВНЗ "Ужгородський національний університет", Ужгород, УКРАЇНА
**email: kotsavlad@gmail.com*

АНОТАЦІЯ *Робота присвячена питанням, які стосуються навчання нейронних елементів і нейромереж із двопороговою функцією активації. Показано, що задача навчання двопорогового нейрона належить до класу NP-повних задач. Наведено достатні умови, які забезпечують можливість генерації двопорогових булевих функцій за допомогою списків рішень. Досліджена задача навчання нейромереж прямого поширення, функції активації яких є згладженими аналогами двопорогових функцій. Продемонстровано результати комп'ютерних експериментів навчання модельних функцій, які свідчать про переваги запропонованого у роботі підходу із використанням функцій активації двопорогового типу.*

Ключові слова: *нейронний елемент, двопороговий нейрон, нейромережа, навчання, алгоритмічна складність.*

АННОТАЦИЯ *Работа посвящена вопросам обучения нейронных элементов и нейросетей с двупороговой функцией активации. Показано, что задача обучения двупорогового нейрона принадлежит к классу NP-полных задач. Приведены достаточные условия реализуемости двупороговых булевых функций с помощью списков решений. Рассмотрена задача обучения нейросетей прямого распространения, функции активации которых являются сглаженными аналогами двупороговой функции. Также приведены результаты компьютерных экспериментов обучения модельных функции, которые демонстрируют преимущества нейросетей с функциями активации двупорогового типа.*

Ключевые слова: *нейронный элемент, двупороговый нейрон, нейросеть, обучение, алгоритмическая сложность.*

LEARNING OF NEURAL NETS WITH BITHRESHOLD-LIKE ACTIVATION FUNCTION

V. KOTSOVSKY

Associate professor of the department of information managing systems, Uzhgorod state university, Uzhgorod, UKRAINE

ABSTRACT *The paper is devoted to the study of the properties of the simplest multithreshold generalization of McCulloch-Pitts neurons, namely bithreshold neurons. The main reason of application of multithreshold device is their more powerful capabilities in comparison with classical threshold units. But multithreshold devices are quite unused because the effective learning algorithm is unknown for such units.*

It is possible to mark out three main goals of the present paper. The first one is the study of the existence of effective learning technique for bithreshold neurons and networks. The second one is the analysis of the relation between Boolean function realizable on bithreshold units and decision lists. The last goal is the study of capabilities of feedforward neural networks with smoothed bithreshold activation function and closely related question of their learning by means of backpropagation.

It is shown that the learning of one bithreshold neural unit is NP-complete. Furthermore, the paper contains the proof of the fact that the task of verification of the bithreshold separability of the finite sets A^+ and A^- is NP-complete even in the case $A^+ \cup A^- \subset \{a,b\}^n$, where $a \in \mathrm{R}$, $b \in \mathrm{R}$ $(a \neq b)$ and the weight coefficients of the neuron may be restricted to be from the set $\{-1,+1\}$. Two ways of overcoming the intractability of bithreshold neurons learning are proposed. Firstly, we can restrict ourselves to consider only those bithreshold units, which are capable to be learned in polynomial time. In particular, it is shown that if we have the decision list $f = (f_1,1),(f_2,1),\ldots,(f_{r-1},1),(f_r,1)$, where f_i $(i = 1,2,\ldots,r-1)$ is an arbitrary Boolean function of two variables assigned the value 1 on two points, and the function f_r is realizable on bithreshold unit, then the same is true for Boolean function f. The second way is based on gradient learning algorithms for neural networks with smoothed bithreshold-like activation function. The simulation results are given confirming the validity of this approach.

Keywords: *neuron, bithreshold neuron, neural network, learning, complexity.*

Introduction

Neural-like units (neurons) are intensively used for solving numerous important practical problems [1]. 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 its convergence is due to Novikoff [1].

Then Minsky and Papert [3] 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 [4]. 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 (e. g. the famous XOR-problem [1, 3]). The using of neurons with more complicated activation functions allowed surmounting this constrain. Historically, the one of the first designed advanced device were multithreshold neural units [4, 5]. But the efficient learning techniques for multithreshold neuron based neural networks aren't developed even in the case of the network with one node.

Goal

The present paper has three main goals. The first one is study of the existence of effective learning technique for bithreshold neurons and networks. The second one is the analysis of the relation between Boolean function realizable on bithreshold units and decision lists. The last goal is the study of capabilities of feedforward neural networks with smoothed bithreshold-like activation function and related question of their learning by means of backpropagation.

Basic 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

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

The graph of corresponding activation function of bithreshold neuron is shown in Fig. 1 (in the case where $a = -1,\ b = 1$).

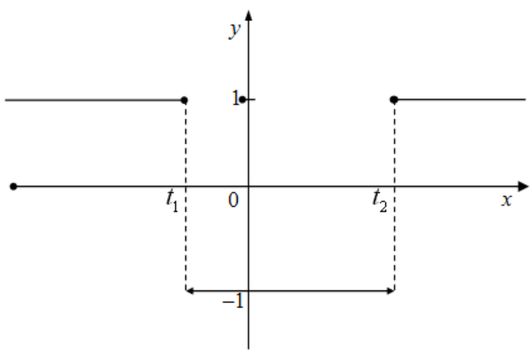

Fig. 1. − The graph of bithreshold activation function

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 +1 or –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 be 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 $f(x_1, \ldots, x_n) : \mathrm{Z}_2^n \to \mathrm{Z}_2$ a "bithreshold function", if exists bithreshold neuron with the structure $(\mathbf{w}, t_1, t_2)$ that $f(\mathbf{x}) = 0 \Leftrightarrow t_1 < (\mathbf{w}, \mathbf{x}) < t_2$. Let LBT_n denote the set of all n-place bithreshold Boolean functions.

Complexity of learning procedure

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

We shall show that if the $\mathrm{P} \ne \mathrm{NP}$ conjecture is true, then don't exist a polynomial time verification algorithm checking the possibility of realization of the arbitrary Boolean function on one bithreshold unit. The learning of bithreshold Boolean function is NP-complete all the more.

Let C be a class of Boolean function: $C = \{C_n\}_{n \ge 1}$, $n \in \mathrm{N}$, $C_n \subset \{f \mid f : \mathrm{Z}_2^n \to \mathrm{Z}_2\}$. In the complexity theory the following 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 [6] that MEMBERSHIP(C) is NP-complete for all classes satisfying following properties:

1) for every $f \in C_n$ and arbitrary $i \in \{1,...,n\}$, both functions $f(x_1,...,x_{i-1},1,x_{i+1},...,x_n)$ and $f(x_1,...,x_{i-1},0,x_{i+1},...,x_n)$ belong to C_{n-1};
2) for every $n \in \mathrm{N}$, the identically 1-function belongs to C_n;
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 the class of bithreshold Boolean functions is NP-complete.*

Proof. We show that class $LBT = \{LBT_n\}_{n \geq 1}$ satisfies conditions 1-3. Condition 1 follows from Shannon expansion $f(x_1,...,x_n) = f(x_1,...,x_{n-1},0)\overline{x}_n \vee \vee f(x_1,...,x_{n-1},1)x_n$. If Boolean function $f(x_1,\ldots,x_n)$ can be realized on the bithreshold neuron with the structure $(\mathbf{w} = (w_1,...,w_{n-1},w_n),t_1,t_2)$, then the functions $f(x_1,...,x_{n-1},1)$ and $f(x_1,...,x_{n-1},0)$ can be realized on bithreshold neurons with structures $((w_1,...,w_{n-1}), t_1 - w_n, t_2 - w_n)$ respectively, $((w_1,...,w_{n-1}),t_1,t_2)$. Condition 2 is evident. Condition 3 follows from the fact that if $n > 2$ Boolean function $x_1 \oplus x_2 \oplus ... \oplus x_n$ doesn't belong to LBT_n [7]. Therefore subject to [6] MEMBERSHIP(*LBT*) is NP-complete.

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

Proof. We use the results of Blum and Rivest from [8], where was shown that the following training problem is NP-complete:

The 3-Node Network with AND output node restricted so that any or all of the weights for one hidden node are required to be opposite to the corresponding weights of the other and any or all the weights are required to belong to $\{-1,+1\}$, since the well-known NP-complete problem Set-Splitting [9] can be reduced to this task.

It is easy to verify that the arbitrary dichotomy (A^+, A^-) is bithreshold if and only if it can be realized on neural network of mentioned type. Really, $\mathbf{x} \in A^- \Leftrightarrow (\mathbf{w},\mathbf{x}) < t_2$ and $(-\mathbf{w},\mathbf{x}) < -t_1$ 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 functions by decision lists

Decision lists were proposed by Rivest in [10]. For many application [10, 11] decision lists are more useful than classical disjunctive or conjunctive normal forms.

Let $K = \{f_1, f_2, ..., f_r\}$ be an arbitrary finite sequence of Boolean functions of *n* variables. A function $f : \mathrm{Z}_2^n \to \mathrm{Z}_2$ 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 $\{c_1, c_2, ..., c_r\}$, $(c_i \in \mathrm{Z}_2,\ i = 1,....,r)$:

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 $f_i \in K, c_i \in \mathrm{Z}_2, (i = 1,2,...,r)$. 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 $K = \{x_1\overline{x}_3, x_2, \overline{x}_1\}$. The decision list $f = (x_1\overline{x}_3, 0), (x_2, 1), (\overline{x}_1, 1)$ may be thought of as operating in the following way on Z_2^3. First, those points for which $x_1\overline{x}_3$ 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 $\overline{x}_1$ 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 $\overline{x}_1\overline{x}_2 \vee \overline{x}_1\overline{x}_3 \vee x_2x_3$.

The relationship between decision lists and threshold Boolean functions was established in [10]. Antony showed (see [6]) that any 1-decision list (that is, a decision list based over the set *K* of single literals) is a 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* (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) $(\bar{x}_i\bar{x}_j\vee x_ix_j,1)$;
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 $\mathbf{e}_i=(0,...,0,\underset{i}{1},0,...,0)$. In the fourth 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 assume $\mathbf{w}=\mathbf{w}'+d\mathbf{e}_i+d\mathbf{e}_j$, $t_1=t_1'+d$, $t_2=t_2'+d$. 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 $\mathbf{x}=(x_1,...,x_i,...,x_n)$ $(\mathbf{w},\mathbf{x})=(\mathbf{w}'+d\mathbf{e}_i,\mathbf{x})=(\mathbf{w}',\mathbf{x})+dx_i$.

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

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

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 $(\mathbf{w},t_1,t_2)$. In case 4 the proof is similar.

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

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

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

$$(\mathbf{w},\mathbf{x})=(\mathbf{w}',\mathbf{x})+2d\geq-\sum_{k=1}^{n}|w_k'|+2d>|t_2'|+$$
$$+\sum_{k=1}^{n}|w_k'|+|t_1'|+|t_2'|+1\geq t_2'+d=t_2.$$

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:*

$$f(x_1,...,x_n)=g(x_1,...,x_n)\vee x_{i_1}^{\alpha_1}\vee...\vee x_{i_l}^{\alpha_l}\vee$$
$$\vee x_{j_1}^{\beta_1}x_{k_1}^{\gamma_1}\vee x_{j_1}^{\bar{\beta}_1}x_{k_1}^{\bar{\gamma}_1}\vee...\vee x_{j_m}^{\beta_m}x_{k_m}^{\gamma_m}\vee x_{j_m}^{\bar{\beta}_m}x_{k_m}^{\bar{\gamma}_m},$$

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 [6] 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*

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

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

The proof follows from the proposition 3 and from [6] (according to the theorem 3.9 from [6] the decision list of the following form $\left(x_{r+1}^{\alpha_1}, c_{r+1}\right),\ldots,\left(x_{r+m}^{\alpha_m}, c_{r+m}\right)$ is a threshold and so a 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 earlier these task is hard even for one neuron. These difficulties can be overcome in the same way as 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:

$$y = 1 - 2e^{-x^2}, \qquad (2)$$

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

Their graphs are shown on Fig. 2 (the graph of the function (3) is "closer" to the graph of the hard bithreshold function (1)).

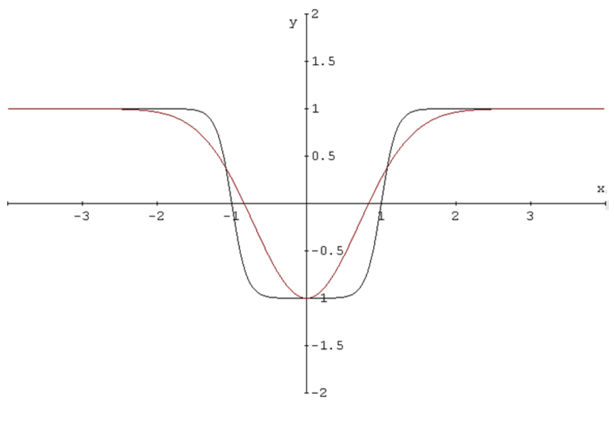

Fig. 2. − The graphs of the smoothed bithreshold activation functions (2)-(3)

We describe here a fairly simple neural net based on smoothed bithreshold neurons, namely the feedforward net (i.e. the multilayer perceptron). We used back-propagation 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 uniformly distributed in hypercube $[-1,1]^{103}$. Then 1000000 iterations of backpropagation procedure are applied for every net. The learning rate parameter was individually chosen for every type of activation function.

Table 1 − Learning in the case of uniform distributed samples

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 results 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 sample size was equal to 1000. For every net 300000 iterations of backpropagation procedure are applied.

Table 2 − Learning XOR function

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).

Results and discussion

It was demonstrated that the basic forms of the task of learning one bithreshold neural unit are hard. For example, it was proved that the task of verification of the bithreshold separability of the finite sets A^+ and A^- is NP-complete even in the case $A^+ \cup A^- \subset \{a,b\}^n$, where $a \in \mathrm{R}$, $b \in \mathrm{R}$ $(a \neq b)$ and the weight coefficients of the neuron may be restricted to be from the set $\{-1,+1\}$.

The relation between bithreshold realizability and realizability by means of decision list was stated. The main result in this domain asserts that if we have the decision list $f = (f_1,1),(f_2,1),...,(f_{r-1},1),(f_r,1)$, where f_i is an arbitrary Boolean function of two variables assigned the value 1 on two points $(i = 1,2,...,r-1)$, and the function f_r is realizable on bithreshold unit, then the function *f* is also bithreshold.

The simulation results given in last section show that multilayer feedforward neural network with smoothed bithreshold-like activation functions can be learnt on the training sample using backpropagation. The data of table 2 confirms that they are capable to solve hard problem of the learning of the XOR-function of several variables.

Conclusions

Neural-like systems on the base of bithreshold neurons were studied. The NP-hardness of bithreshold neurons learning was established. Two ways were proposed to overcome the hardness of learning procedure. The conditions were found providing that decision list realizes a bithreshold logic function. The approach was proposed concerning neural networks with smoothed bithreshold activation functions. The experimental results confirming effectiveness of this approach were given. It seems that bithreshold neurons can be useful in areas of traditional applications of neural-like devices.

Список літератури

1 **Haykin, S.** Neural networks: A comprehensive foundation / **S. Haykin**. − Upper Saddle River, NJ: Prentice Hall, Inc., 1999.

2 **Rosenblatt, F.** Principles of neurodynamics / **F. Rosenblatt**. − Washington, DC: Spartan Books, 1962.

3 **Minsky, M. L.** Perceptrons, expanded editions / **M. L. Minsky, S. A. Papert**. − Cambridge, MA: MIT Press, 1988.

4 **Peled, U.** Polynomial-time algorithm for regular set covering and threshold synthesis / **U. Peled, B. Simeone** //, *Discrete Applied Mathematics*. − 1985. – V. 12. − P. 57-69.

5 **Бахарев, А.** Оптимизация многопороговых моделей / **А. Бахарев** // *Проблемы случайного поиска*. – 1975. − Вып. 4. – С. 209-2014.

6 **Anthony, M.** Discrete mathematics of neural networks / **M. Anthony**. − Philadelphia: SIAM, 2001.

7 **Geche, F.** Bithreshold neurons learning / **F. Geche, V. Kotsovsky, A. Batyuk, A. Mitsa, K. Gromaszek** // *Wydzial Elektroniki i Informatyki Politechniki Lubelskiej: Elektronika*. — Warszava: SIGMA-NOT, 2013. – № 8. − P. 70-72.

8 **Blum, A.** Training a 3-*node neural network is* NP-complete / **A. Blum A., R. Rivest** // *Neural Networks*. – 1992. – V. 5, № 1. − P. 117-127.

9 **Garey, M. R.** Computers and intractability / **M. R. Garey, D. S. Johnson**. − New York: W. H. Freeman, 1979.

10 **Rivest, R.** Learning decision lists / **R. Rivest** // *Machine Learning*. – 1987. − V. 2. − P. 229-246.

11 **Marchand, M.** On learning simple neural concepts: From halfspace intersection to neural decision lists / **M. Marchand, M. Golea** // *Networks: Computation in Neural Systems*. – 1993. − V. 4. − P. 67-85.

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 **Minsky, M. L., Papert, S. A.** Perceptrons, expanded editions. Cambridge, MA: MIT Press, 1988.

4 **Peled, U., Simeone, B.** Polynomial-time algorithm for regular set covering and threshold synthesis. *Discrete Applied Mathematics*, 1985, **12**, 57-69.

5 **Baharyev, A. T.** Optimization of multithreshold models. *Problemy sluchaiynogo poiska*, 1975, **4**, 209-214.

6 **Anthony, M.** Discrete mathematics of neural networks. Philadelphia: SIAM, 2001.

7 **Geche, F., Kotsovsky, V., Batyuk, A., Mitsa, A., Gromaszek, K.** Bithreshold neurons learning. *Wydzial Elektroniki i Informatyki Politechniki Lubelskiej: Elektronika*. Warszava: SIGMA-NOT, 2013, **8**, 70-72.

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

9 **Garey, M. R, Johnson, D. S.** Computers and Intractability. New York: W. H. Freeman, 1979.

10 **Rivest, R.** Learning decision lists. *Machine Learning*, 1987, **2**, 229-246.

11 **Marchand, M., Golea, M.** On learning simple neural concepts: From halfspace intersection to neural decision lists. *Networks: Computation in Neural Systems*, 1993, **4**, 67-85.

Надійшла (received) 05.10.2015