IEEE Second International Conference on Data Stream Mining & Processing
August 21-25, 2018, Lviv, Ukraine

Finite Generalization of the Offline Spectral Learning

Vladyslav Kotsovsky
IMST Department
Uzhhorod National University
Uzhhorod, Ukraine
kotsavlad@gmail.com

Fedir Geche
Department of Cybernetics and Applied Mathematics
Uzhhorod National University
Uzhhorod, Ukraine
fgeche@hotmail.com

Anatoliy Batyuk
ACS Department
Lviv Polytechnic National University
Lviv, Ukraine
abatyuk@gmail.com

***Abstract*—We study the problem of offline learning discrete functions on polynomial threshold units over specified set of polynomial. Our approach is based on the generalization of the classical "Relaxation" method of solving linear inequalities. We give theoretical reason justifying heuristic modification improving the performance of spectral learning algorithm. We demonstrate that if the normalizing factor satisfies sufficient conditions, then the learning procedure is finite and stops after some steps, producing the weight vector of the polynomial threshold unit realizing the given threshold function. Our approach can be applied in hybrid systems of computational intelligence.**

Keywords—offline learning, polynomial threshold unit, threshold function, artificial neural network.

I. Introduction

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 have been proposed. Polynomial threshold units (PTU) are ones of the most powerful between neural-like units with threshold activation function. They are based on separation of the n-dimensional space by the polynomial hypersurface.

Our offline learning algorithm for PTU uses the basic idea of "Relaxation" method introduced by Motzkin and Schoenberg [2]. Many different modifications of this method concerning online learning algorithms for perceptron-like devices proposed in [1, 3]. The offline modification of the algorithm with similar learning rules described in [4] for linear threshold units. Its generalization for PTU may be found in [5]. The main lack of these algorithms is the possibility of infinite learning time and convergence to the boundary point of the acceptable solution set. Hampson and Kibler proposed the modification of the choice of the amount of correction for online learning [6]. They announced that slightly larger normalizing factor in Reflection algorithm improves performance, but their reasons are rather "heuristic" and based only on empirical data.

The paper has the following organization: first the structure of PTU over given set of polynomials X and the notion of X-threshold function are given. Then basic concepts of our learning framework are described by using of the spectral technics similar to proposed in [7]. The rule of the choice of learning coefficients is discussed. In the next chapter the finiteness of the learning is proved. Finally, we analyze the results of computer simulation and make conclusions.

II. Polynomial Threshold Units

A computation unit with n inputs $x_1,\dots,x_n$ and one output $y\in\{-1,1\}$ is said to be a *polynomial threshold unit* if

$$y=\operatorname{sgn}\left(\sum_{k=1}^{m} w_k \prod_{i=1}^{n} x_i^{j_{ki}}\right),\quad j_{ki}\in\{0,1,\dots\},\ i=1,\dots,n, \qquad (1)$$

where $\mathbf{w}=(w_1,\dots,w_n)$ is the weight vector and the activation function is the sign function, given by

$$\operatorname{sgn}(x)=\begin{cases}1 & \text{if } x>0,\\ 0 & \text{if } x=0,\\ -1 & \text{if } x<0.\end{cases}$$

It should be noted that if the weighted sum $\sum_{k=1}^{m} w_k \prod_{i=1}^{n} x_i^{j_{ki}}$ is equal to 0, then the output value of the PTU is incorrect (i.e. it does not belong to bipolar set $\{-1,1\}$). But it is easy to prove that using "small" changes of PTU weights we can always avoid this inconvenience.

PTU and neural nets on their base are useful in pattern recognition methods providing sets separation in n-dimensional Euclidean space. One of the most important tasks is the one of separating the Boolean hypercube $\{-1,1\}^n$ vertices to two different sets. For such task we can limit oneself in (1) with the terms of the form $w_{i_1\dots i_k} x_{i_1}\dots x_{i_k}$. This is because $x^{2l}=1,\ x^{2l+1}=x\ (l\in \mathrm{N})$ in such case. From here we will restrict our attention only to PTU with inputs belonging to the bipolar set $\{-1, 1\}$.

We think of a Boolean function as a mapping $f:E_2^n\to E_2$, where $E_2=\{-1,1\}$. Let us define functions $\chi_j:E_2^n\to E_2$ by $\chi_j(\mathbf{a})=a_1^{j_1}\cdot\ldots\cdot a_n^{j_n}$, where $\mathbf{a}=(a_1,\dots,a_n)\in E_2^n$, $j=j_1 2^{n-1}+j_2 2^{n-2}+\dots+j\ (j_i\in\{0,1\},\ i=1,\dots,n)$. Thus

$\chi_j(\mathbf{a}) = a_{l_1} \cdot \ldots \cdot a_{l_r}$, if $j = 2^{n-l_1} + 2^{n-l_2} + \ldots + 2^{n-l_r}$, $1 \le l_1 < \ldots < l_r \le n$. The mappings χ_j, $j = 0,1,\ldots,2^n - 1$ are well-known as characters of the group E_2^n over the real number field R.

Let $X = \{\chi_{i_1}, \ldots, \chi_{i_m}\}$ be an arbitrary set of characters and $f : E_2^n \to E_2$ is the given Boolean function. If there exists a such weight vector $\mathbf{w} = (w_1, \ldots, w_m) \in R^m$ that

$$\text{for all } \mathbf{a} = (a_1, \ldots, a_n) \in E_2^n \quad f(\mathbf{a}) = \operatorname{sgn}\left(\sum_{j=1}^{m} w_j \chi_{i_j}(\mathbf{a})\right),$$

then we say that f is realizable on the PTU, or f is X-threshold function.

Furthermore, the weight vector $\mathbf{w}$ must satisfy the following condition: for all $\mathbf{a} \in E_2^n$ $(\mathbf{w}, \chi(\mathbf{a})) \ne 0$, where $(\mathbf{w}, \chi(\mathbf{a})) = w_1 \chi_{i_1}(\mathbf{a}) + \ldots + w_m \chi_{i_m}(\mathbf{a})$ is the inner product of the vector $\mathbf{w}$ with $\chi(\mathbf{a}) = (\chi_{i_1}(\mathbf{a}), \ldots, \chi_{i_m}(\mathbf{a}))$. We call such weight vectors X-acceptable. Note that the notion of X-threshold function is the generalization of the notion of threshold function [4]. We shall denote by $W_X(f)$ the set of all weight vectors of all PTU realizing the given X-threshold function f.

III. Learning Framework

We assume that in our model of supervised learning the set of polynomials (characters) X is fixed and an arbitrary X-threshold function $f : E_2^n \to E_2$ is given. We will be interested in algorithm of finding some finite sequence of weight X-acceptable vectors

$$\mathbf{w}^0, \mathbf{w}^1, \ldots, \mathbf{w}^r \tag{2}$$

that the function f can be realized on PTU with weight vector $\mathbf{w}^r$.

For function $f(x_1, \ldots, x_n)$ we will define the spectral vector $\mathbf{s}_X(f) = (s_1, \ldots s_m)$ respectively to the set X in the following way:

$$s_j = \sum_{\mathbf{a} \in E_2^n} \chi_{i_j}(\mathbf{a})\, f(\mathbf{a}), \quad j = \overline{1,m}.$$

Let $\mathbf{w} \in W_X(f) \subset R^m$ be an arbitrary weight vector of the PTU realizing the function f over the set X. It is important for us that the set $W_X(f)$ is a cone. Suppose that the first k+1 members $\mathbf{w}^0, \mathbf{w}^1, \ldots, \mathbf{w}^k$ of the sequence (2) are already chosen and let $\mathbf{w}^k$ be an m-dimensional X-acceptable real vector. If the function f can be realized on PTU with the weight vector $\mathbf{w}^k$, then the learning succeed.

Suppose that previous assumption is wrong. This implies that f is not realizable on PTU with the weight vector $\mathbf{w}^k$. Let us describe how we can obtain the vector $\mathbf{w}^{k+1}$ closer to the all $\mathbf{w} \in W_X(f)$ than the previous vector $\mathbf{w}^k$, i.e.

$$\left\|\mathbf{w}^{k+1} - \mathbf{w}\right\| < \left\|\mathbf{w}^k - \mathbf{w}\right\|, \tag{3}$$

where $\|\mathbf{x}\| = \sqrt{(\mathbf{x}, \mathbf{x})}$ is the Euclidean norm in the space R^m. The condition (3) is well-known Fejér condition [2].

Let us use the learning rule:

$$\mathbf{w}^{k+1} = \mathbf{w}^k + \mathbf{z}^k, \tag{4}$$

where $\mathbf{z}^k$ is the correction vector. Now we consider the question of the choice of the increment $\mathbf{z}^k$. Let $f^k(x_1, \ldots x_n)$ be a Boolean function realizable over the set X on PTU with weight vector $\mathbf{w}^k$. From previous assumption it follows that $f^k \ne f$ and $\mathbf{s}_X(f^k) \ne \mathbf{s}_X(f)$. Similarly to [4] we select the increment vector $\mathbf{z}^k$ in the following way:

$$\mathbf{z}^k = \alpha_k \left(\mathbf{s}_X(f) - \mathbf{s}_X(f^k)\right),$$

where α_k is a some positive coefficient.

We proved in [8] that inequality (3) is held for fixed $\mathbf{w} \in W_X(f)$, if the coefficient α_k is chosen to be $t_k \alpha_k^0$, where

$$\alpha_k^0 = \frac{\left(\mathbf{w}^k, \mathbf{s}_X(f^k) - \mathbf{s}_X(f)\right)}{\left\|\mathbf{s}_X(f) - \mathbf{s}_X(f^k)\right\|^2},$$

and the value of the amount of correction t_k satisfies the following inequality:

$$0 < t_k < 2\left(1 + \frac{\left(\mathbf{w}, \mathbf{s}_X(f) - \mathbf{s}_X(f^k)\right)}{\left(\mathbf{w}^k, \mathbf{s}_X(f^k) - \mathbf{s}_X(f)\right)}\right). \tag{5}$$

According to the terminology of [4] we call the coefficient t_k a normalizing increment coefficient.

From now on, we restrict ourselves consideration of increment vectors $\mathbf{z}^k$ of the form

$$\mathbf{z}^k = t_k \alpha_k^o \left(\mathbf{s}_X(f) - \mathbf{s}_X(f^k)\right), \tag{6}$$

where the normalizing factor t_k satisfies (5).

Remark 1. Note that for increments (6) the inequality (3) holds for each weight vector $\mathbf{w} \in W_X(f)$ under the condition that the normalizing coefficient t_k satisfies inequality $0 < t_k \le 2$. Sometimes, it is more convenient to require the Fejér condition for all $\mathbf{w} \in A \subset W_X(f)$. Under this condition it can be possible to obtain the upper bound in (5) greater than 2 (for all $\mathbf{w} \in A$).

Remark 2. When we chose the increment vector $\mathbf{z}^k$ in the form (6) it is necessary to require that for all $\mathbf{a} \in E_2^n$ $\left(\mathbf{w}^{k+1}, \chi(\mathbf{a})\right) \ne 0$. In another case the weight vector $\mathbf{w}^{k+1}$ is not X-acceptable and in according to our definition of PTU it is impossible to use $\mathbf{w}^{k+1}$ for representing any Boolean function. We always can reach it by small changes of the increment vector (changing the normalizing increment coefficient t_k in such way that condition (3) holds). Let $\mathbf{w}^k$ be an X-acceptable vector, but $\mathbf{w}^{k+1} = \mathbf{w}^k + \mathbf{z}^k$ is already an unacceptable one. Then the set $A_k = \left\{\mathbf{a} \,\middle|\, \mathbf{a} \in E_2^n, \left(\mathbf{w}^{k+1}, \chi(\mathbf{a})\right) = 0\right\}$ is nonempty. We can use the increment $\tilde{\mathbf{z}}^k = (1 - \beta_k)\, \mathbf{z}^k$, where β_k is an arbitrary factor satisfying following condition:

$$0 < \beta_k \le \frac{1}{2} \min\left\{1, \frac{\min\left\{\left|\left(\mathbf{w}^{k+1}, \chi(\mathbf{a})\right)\right| \,\middle|\, \mathbf{a} \in E_2^n \setminus A_k\right\}}{\max\left\{\left|\left(\mathbf{w}^{k+1}, \chi(\mathbf{a})\right)\right| \,\middle|\, \mathbf{a} \in A_k\right\}}\right\}.$$

It is easy to verify that $\tilde{\mathbf{w}}^{k+1} = \mathbf{w}^k + \tilde{\mathbf{z}}^k$ is the X-acceptable weight vector.

IV. Finite Modification Of Learning Algorithm

We have already mentioned that the offline modification of the rule (4) is due to Dertouzos [4] (see also [5]). In both works the coefficients t_k is from $(1,2]$, the algorithm is not guaranteed to terminate after a finite number of steps, and some rather complicated techniques are used. The first condition guaranting the finitness of the spectral offline learning was established in [8], where the rule was proposed of the normalizing coefficients choice, which utilization in spectral learning algorithm for X-threshold Boolean functions *f* ensures the finiteness of the learning procedure (i.e. after finite number of steps we certainly obtain the weight vector $\mathbf{w}^r \in W_X(f)$). If the coefficients t_k are chosen according to

$$t_k = 2 + \frac{1}{\left(\mathbf{w}^k, \mathbf{s}_X\left(f^k\right) - \mathbf{s}_X(f)\right)}, \qquad (7)$$

then the learning process (4), (6) is finite.

Now we can state our main results generalizing above mentioned one. For our purpose we will choose the coefficients t_k in the following way:

$$t_k = \sigma_k + \frac{\tau_k}{\left(\mathbf{w}^k, \mathbf{s}_X\left(f^k\right) - \mathbf{s}_X(f)\right)}, \qquad (8)$$

where $\sigma_k > 0,\ \tau_k > 0\ (k = 0,1,\ldots)$. It easy to see that the rule (8) is the generalization of (7) with two additional sequences of parameters.

Proposition 1. If a Boolean function *f* is X-threshold, the sequence (2) of X-acceptable weight vectors $\left\{\mathbf{w}^k\right\}$ is built on the base (4) and (6), the normalizing coefficients t_k satisfy (8) (accordingly to Remark 2), coefficients σ_k satisfy

$$0 \le \sigma_k \le 2\ \ (k = 0,1,\ldots), \qquad (9)$$

and the sequence $\{\tau_k\}$ is bounded and does not converge to zero, then the learning process terminates after finite number of steps on some weight vector $\mathbf{w}^r \in W_X(f)$.

Proof. We need two simple lemmas established in [9].

Lemma 1. If for all $\mathbf{x} \in X$ $\|\mathbf{v} - \mathbf{x}\| < \|\mathbf{w} - \mathbf{x}\|$ then for all $\mathbf{y} \in \operatorname{conv} X$ the inequality $\|\mathbf{v} - \mathbf{y}\| < \|\mathbf{w} - \mathbf{y}\|$ holds, where $\operatorname{conv} X$ is the convex hull of set *X* in some Euclidean space.

Lemma 2. Let $\left\{\mathbf{x}^n\right\}$ be a sequence of the vectors of R^n, $Z \subset \mathrm{R}^n$ and the set *Z* contains at least one inner point. Then if each $\mathbf{z} \in Z$ satisfies inequality $\|\mathbf{x}^{n+1} - \mathbf{z}\| < \|\mathbf{x}^n - \mathbf{z}\|$, then the sequence $\left\{\mathbf{x}^n\right\}$ is convergent.

We prove the statement of the theorem by contradiction. Suppose that the sequence $\left\{\mathbf{w}^k\right\}$ is infinite. It is easy to verify that for an arbitrary set of characters X and for an arbitrary Boolean function *g* the coordinates of the spectral vector $\mathbf{s}_X(g)$ are even integer numbers. Hence, for an arbitrary fixed integer vector $\mathbf{w}' \in W_X(f)$ (note that such vectors are always presented in the convex cone $W_X(f)$) the numerator $\left(\mathbf{w}', \mathbf{s}_X(f) - \mathbf{s}_X\left(f^k\right)\right)$ of the fraction in the right part of (5) is an even natural number. Let *A* be a subset of integer weighted vectors belonging to set $W_X(f)$ $\left(A = W_X(f) \cap \mathrm{Z}^m\right)$ and $\tau = \sup\{\tau_k\}$. For all weight vectors from the set $\tau A = \{\tau \mathbf{w} \,|\, \mathbf{w} \in A\}$ the choice of the increment coefficients according to (8), (9) ensures (5). Thus, the Fejér condition for the set $\tau W_X(f)$ (inequality (3)) also holds for each member of the sequence $\left\{\mathbf{w}^k\right\}$. Let us also consider integer vectors $\mathbf{u}^j = 2\mathbf{w}' - \mathbf{e}^j$, $\mathbf{v}^j = 2\mathbf{w}' + \mathbf{e}^j$, $(j = 1,\ldots,m)$, where $\mathbf{e}^j$ are the unit basis vector of the space R^m. It easy to see that $\mathbf{u}^j \in W_X(f)$ and $\mathbf{v}^j \in W_X(f)$, $j = \overline{1,m}$. Let us see $B = \operatorname{conv}\{\mathbf{u}^1, \ldots, \mathbf{u}^m, \mathbf{v}^1, \ldots \mathbf{v}^m\}$. Now, using Lemma 1 to the polyhedron τB, we obtain that for each $\mathbf{z} \in \tau B$ and for all members of the sequence $\left\{\mathbf{w}^k\right\}$ the following inequality holds: $\|\mathbf{w}^{k+1} - \mathbf{z}\| < \|\mathbf{w}^k - \mathbf{z}\|$. The set of the interior points of the polyhedron τB is nonempty (e.g. it contains the vector $2\tau \mathbf{w}'$). Hence, following Lemma 2, the sequence $\left\{\mathbf{w}^k\right\}$ is convergent. But from (6) it follows that for increment vectors

$$\alpha_k = \left(\sigma_k + \frac{\tau_k}{\left(\mathbf{w}^k, \mathbf{s}_X\left(f^k\right) - \mathbf{s}_X\left(f\right)\right)}\right)\frac{\left(\mathbf{w}^k, \mathbf{s}_X\left(f^k\right) - \mathbf{s}_X\left(f\right)\right)}{\left\|\mathbf{s}_X\left(f^k\right) - \mathbf{s}_X\left(f\right)\right\|^2} =$$

$$= \frac{\sigma_k\left(\mathbf{w}^k, \mathbf{s}_X\left(f^k\right) - \mathbf{s}_X\left(f\right)\right) + \tau_k}{\left\|\mathbf{s}_X\left(f^k\right) - \mathbf{s}_X\left(f\right)\right\|^2}.$$

The denominator of the last fraction is bounded and $\left(\mathbf{w}^k, \mathbf{s}_X\left(f^k\right) - \mathbf{s}_X\left(f\right)\right) > 0$ [8]. Then, there exists such $\tau_{\min} > 0$ that for each k_0 there exists such $k > k_0$ that numeartor of the last fraction is not less than $\tau_{\min}$. So, the condition $\mathbf{z}^k \to 0$ is violated, which is the necessary condition for the convergence of the sequence $\left\{\mathbf{w}^k\right\}$. Therefore, on some step of the learning algorithm we obtain X-acceptable weight vector $\mathbf{w}^r \in W_X\left(f\right)$.

V. Empirical Results

To study the dependence of efficiency of PTU learning on the value of coefficients t_k we have implemented a simulation test based on the learning of threshold Boolean functions corresponding to 10000 randomly generated weight vectors for $n = 8$ and $n = 10$ features. We used $X = \{1, x_1, ..., x_n\}$ and constant $t_k = t$ instead of (8) for the sake of simplicity. For the comparison, we learned each function using two online algorithms Reflection and Reflect1 with ShuffleCycle order of the inputs [6].

First, we learned all generated Boolean functions for each $t \in (0,10]$ with the step 0.05 and counted the number of successful learnings and the average amount of corrections. The initial appro $\bar{\varepsilon}^{(l)} = \frac{1}{(\hat{\alpha}^{(l)})^{1/2}} \sum_{s=1}^{p} \mu_s^{(l)}, \quad l = 1, ..., M$ ximations were chosen randomly. The general tendencies are following:

1. In the case $t \in (0,1]$ the learning failed due to violence of the practical X-acceptability of weight vectors (the absolute values of weighted sums became less than 10^{-15}).
2. In the case $t > 1$ all learning terminated successfully.
3. In the case $t > 3$ the quick growth was observed of the number of iterations.

Fig. 1 provides an illustration of growth of the correction curve in the case $t > 3.3$. In the case $t \geq 3.9$ the average number of corrections exceeds 255 (not shown in this figure).

Then we restricted ourselves on the study of the case $t \in (1,3)$. The number of functions left the same and 200 points were chosen uniformly (with the step 0.01). The corresponding curves of the number of iterations are shown in Fig. 2. As can be seen, the iteration number is complicated nonunimodal function which minimum point is in the neighborhood of the point $t = 2$: $t_{\min} = 2.02$, in the case $n = 8$, $t_{\min} = 2.03$ in the case $n = 10$.

Then we thoroughly studied the case $t \in [1.9, 2.1]$ with the step 0.001. The results are shown in Fig. 3. We found that $t_{\min} = 1.991$, $c_{\min} = 3.109$ in the case $n = 8$, and $t_{\min} = 1.997$, $c_{\min} = 4.462$ in the case $n = 10$, where $c_{\min}$ is the corresponding number of iteration in spectral algorithm.

Fig. 1. Average number of iterations in (1.1, 3.9) in the case n = 8.

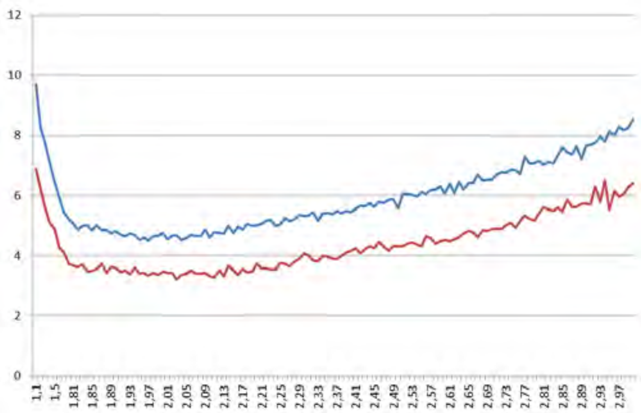

Fig. 2. Average numbers of iterations in (1, 3) in the cases n = 8 (lower curve) and n = 10 (upper curve).

For comparison, the average numbers of adjustments for Reflect1 are, respectively, 42.137 and 63.506 (the performance of Reflection is similar).

Results in Fig. 1-3 are obtained in the case of random initial approximations. The performance of the our learning algorithm can be improved by using the optimized approximation $\mathbf{w}^0 = \mathbf{s}_X\left(f\right)$. The reasons are given in [4, 5] and results are shown in Fig. 4. Comparing Fig. 3 and Fig. 4 we can observe that the correction number has halved roughly in the case $n = 8$ and has decreased by one third in the case $n = 10$.

Fig. 3. Average numbers of iterations in [1.9, 2.1] in the cases n = 8 (lower curve) and n = 10 (upper curve).

Fig. 4. Average numbers of iteration in [1.9, 2.1] in the cases n = 8 (lower curve) and n = 10 (upper curve) with optimized initial approximation.

Finally, we studied the case $n = 20$ on 1000 random samples. For the spectral algorithm with optimized initial approximation $t_{\min} = 2.001$, $c_{\min} = 6.809$, and the average numbers of adjustments for Reflect1 is 252,413. The learning times are similar (Reflect is slightly faster). Note that it is possible to improve the performance of the learning algorithm by using t_k in the form (8). E.g., the simulation in the case $n = 10$ showed that we can increase performance by 12% using t_k for which $\sigma_k = 1.995$, $\tau_k = 1$.

VI. Conclusions

We proposed the new modification of offline learning method based on spectral approach. Our rule of the choice of normalizing coefficient ensures finite learning time for all X-threshold functions. In addition, in case of offline learning we confirmed the hypothesis of Hampson that it is reasonable to choose correction amount slightly larger than 2 [6]. The experimental results confirm the effectiveness of developed method. They suggest to apply offline learning with optimized initial approximation $\mathbf{w}^0 = \mathbf{s}_X(f)$ and coefficients $\sigma_k \in [1.99, 2.01]$, $(k = 0,1,...)$ to obtain a good performance.

It should be mentioned that our approach can be utilized in learning PTU to recognize the subsets of an arbitrary finite set in n-dimensional Euclidean space. Our learning technics can be applied to improve performance of basic components proposed in [9]. Nonlinear classifier on the base of PTU may be also used in systems described in [10].

It seems to be also interesting to find the bounds of the number of algorithm iterations, to estimate the size of corresponding integer weights of PTU and to compare them with the ones from [11].

VII. Acknowledgment

We would like to thank M. Zhurbenko for many helpful discussions concerning Proposition 1.

References

[1] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1999.

[2] T. Motzkin and I. Schoenberg, “The relaxation method for linear equalities,” Canadian Journal of Math., vol. 6, pp. 393−404, 1954.

[3] R. Duda, P. Hart and D. Stork, Pattern Classification, 2nd ed. New York: Wiley-Interscience, 2001.

[4] M. L. Dertouzos, Threshold Logic: A Synthesis Approach. Cambridge, MA: The MIT Press, 1965.

[5] F. Geche. Analysis of Discrete Functions and Logical Circuits in Neural Basis. Uzhgorod: Vydavnytstvo V. Padyaka, 2010. (in Ukrainian)

[6] S. Hampson and D. Kibler, “Minimum generalization via reflection: a fast linear threshold learner,” Machine Learning vol. 37(1), pp. 51-73, 1999.

[7] J. Bruck, “Harmonic analysis of polynomial threshold functions,” Siam Journal on Discrete Mathematics, vol. 3 (2), pp. 168–177, 1990.

[8] F. E. Geche, V. M. Kotsovsky and A. Ye. Batyuk, “Learning algorithms for generalized neurons over character set,” Zbirnyk naukovykh prats instytutu problem modelyuvannya v energetytsi NAN Ukrayiny, vyp. 41, pp. 124-136, 2007. (in Ukrainian)

[9] I. Tsmots, V. Teslyuk, T. Teslyuk and I. Ihnatyev, “Basic components of neuronetworks with parallel vertical group data real-time processing,” Advances in Intelligent Systems and Computing, vol. 689, Springer, Cham., pp. 558–576, 2018.

[10] V. Teslyuk, V. Beregovskyi, P. Denysyuk, T. Teslyuk and A. Lozynskyi, “Development and implementation of the technical accident prevention subsystem for the smart home system,” International Journal of Intelligent Systems and Applications, vol. 10, No.1, pp. 1–8, 2018.

[11] F. Geche, V. Kotsovsky and A. Batyuk, “Synthesis of the integer neural elements,” in Proceedings of the International Conference on Computer Sciences and Information Technologies CSIT 2015, Lviv, Ukraine, September 14-17 2015, pp. 121–136.