Processing Symbolic Sequences by the BCM Neuron

Pavel Petrovič[1], Peter Tiňo[2] and Lubica Beňušková[2]

[1]*Comenius University, Faculty of Mathematics and Physics, Institute of Informatics, 84215 Bratislava, Slovakia*
e-mail: ppetrovic@acm.org

[2]*Slovak University of Technology, Dept. of Computer Science and Engineering, 81219 Bratislava, Slovakia*
e-mail: {tino, benus}@dcs.elf.stuba.sk

Abstract. We have investigated the biologically motivated model of a neuron, the synaptic weights of which modify according to the Bienenstock, Cooper and Munro (BCM) theory of synaptic plasticity. The crucial concept of the BCM theory is the dynamic synaptic modification threshold which determines whether a neuron's activity at any given instant will lead to strengthening or weakening of its synaptic weights. We design, perform and discuss several experiments with complex two-symbol ('0' and '1') time sequences in order to investigate computational properties of the BCM neuron. We measure the statistical structure of sequences using the entropies and the thermodynamic entropy spectra. We have found that for the two sequences with different entropies and the same number of symbols '1', the asymptotic value of the BCM neuron response differs, in the sense that it is dependent on the internal structure of the input sequence. By comparing the properties of input symbolic sequences and symbolic sequences derived from the evolution of the synaptic modification threshold, we conclude that both nonlinear and linear BCM neuron with the shortest "internal memory", i.e. $\tau \leq 2$, are able to model the input sequences, whereas the BCM neurons with a longer internal memory function as "low-pass filter".

1 Introduction

The BCM theory has been introduced in order to explain plasticity in the developing visual cortex [3]. Later it has been used for explanation of experience-dependent plasticity in the mature brain cortex [2]. Besides biological applications, the BCM learning rule was applied in several different situations. It was shown, for example, that a modified version can perform efficient projection pursuit, detect bimodality, statistical skewness and curtosis in the distribution of input data [1, 4]. The original motivation for our experiments with time series was synaptic modification threshold θ_M of the BCM neuron which is a nonlinear function of its past activity. We believed that it should have an interesting impact on the ability of the BCM neuron to process time sequences. We studied behavior of a single BCM neuron exposed to symbolic time sequences built from 2 symbols, namely '0' and '1'. Symbols from the input sequence were feeding the weighted neuron's input. The weight w and synaptic modification threshold θ_M were updated after each input symbol. Fig. 1 shows a scheme of this circuit. The evolution of both, neuron's weight w and threshold θ_M, was observed.
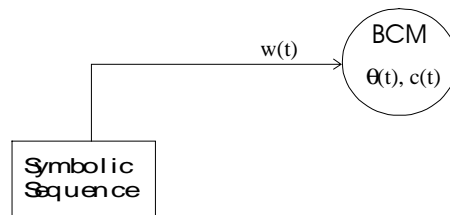

Fig. 1. Scheme of a circuit for our experiments with symbolic time series and the BCM neuron (*c(t)* is the neuron's response proportional to the input weight *w(t)*).

We performed the experiments with the following variants of the BCM neuron: linear neuron, linear neuron with noise added to the input, nonlinear neuron (with sigmoid activation function), and nonlinear neuron with noise added to the input. In order to have a possibility to compare the behavior of the BCM neuron for sequences with different degrees of determinism, we used three different kinds of sequences:

- deterministic (produced by a final-state automaton) (DET)
- sequence produced by a Hidden Markov Model (HMM) [8]
- random sequence (Bernoulli source) (RND)

We worked with 4 different sets of sequences S = A, B, C, and D. Each set contained one periodic deterministic sequence (S1), one HMM sequence (S2) and one random sequence (S3). The difference between the sets of sequences was primarily in the nature of HMM automaton for HMM sequences. Sequences within each set contained the same number of symbols '1', but differed more or less in the values of entropy. We can consider the values of the entropy as

an appropriate measure of the different degrees of determinism of sequences in one set [8]. Even more information can be traced from entropy spectra. We describe these concepts in the next section.

Each experiment produced a sequence of cell responses c (which are proportional to the input weight in the case of one input), and a sequence of values of θ_M. For the two sequences with different entropies and the same number of symbols '1', we have found out that the asymptotic value of weight differs, i.e. it is dependent on the entropy of the input sequence (or at least has a strong relation with it).

We were interested also in the nature of the dynamics of the threshold θ_M. The role of the threshold θ_M is to compensate for changes in input. Thus it was interesting to investigate the sequence of differences from θ_M's average (expected value). This sequence of positive and negative arithmetic differences from the θ_M's expected value can be transformed into a symbolic time sequence of '1's and '0's. We have computed entropy spectra for these θ_M symbolic sequences and compared them with the entropy spectra of the original input sequences. Further details can be found in the section with results.

2   Theoretical Background

In this section, we will give the theoretical description of the BCM neuron as well as the measures used for reasoning about the results. Let d is the input to neuron, w is the weight of input connection, θ_M is the threshold for weight modification and ϕ is the polynomial function used in the BCM weight modification rule. The neuron's response c at time t is then given by:

$$c(t) = w(t).d(t) \tag{1}$$

for a linear neuron and

$$c(t) = \sigma(w(t).d(t)) \tag{2}$$

for a nonlinear neuron, where $\sigma(c) = 1/(1+ \exp(-c))$. The BCM weight modification rule is expressed by

$$\Delta w(t) = \eta \ \phi (c(t), \theta_M(t)) \ d(t) \tag{3}$$

for a linear neuron and

$$\Delta w(t) = \eta \ \phi (c(t), \theta_M (t)) \ \sigma'(c) \ d(t) \tag{4}$$

for a nonlinear neuron, where according to [5]

$$\phi(c, \theta_M) = c \ (c - \theta_M) \tag{5}$$

In cases with additive noise, we used $d(t)+noise(t)$ instead of $d(t)$, where $noise(t)$ is a Bernoulli source, uniformly distributed in the interval $[-\alpha,\alpha]$. The modificationn threshold θ_M is defined by equation (6). In addition, a scaling parameter c_0 allows scaling of this time average of the squared cell response, such that

$$\theta_M(t) = \frac{1}{c_0} \left\langle c^2(t) \right\rangle_\tau = \frac{1}{c_0} \frac{1}{\tau} \int_{-\infty}^{t} c^2(t') \ e^{-\left[\frac{(t-t')}{\tau}\right]} dt' \tag{6}$$

Here, the parameter τ determines the length of the recent past over which the squared neuron's response is averaged. We call it the "internal memory" of a neuron for its past activity. The smaller τ, the shorter memory of the neuron to its past activity.

For determining the entropy of symbolic sequences, we have used the calculation of entropies for increasing length of window [6, 8]. If the probability of some event is P_i, its information content is given by the formula:

$$I_i = \log_2 \frac{1}{P_i} \tag{7}$$

Based on this, Shannon introduced the entropy of the data as the sum of products of probabilities and information contents of all possible events:

$$\left\langle I \right\rangle_p = \sum_i P_i I_i \tag{8}$$

For a window of length n there exist 2^n different words (having only two symbols '0' and '1'). Let the probability of word w_i of length n is $P_n(w_i)$. Entropy of the sequence for window of length n is thus:

$$H_n = -\sum_{|w|=n} P_n(w)\log_2 P_n(w) \tag{9}$$

Entropy can be understood as a measure of uncertainty. The uncertainty is high for large H_n, i.e. all possible words are almost equally likely to occur in the sequence. For small values of H_n, the uncertainty is smaller, i.e. the sequence is more deterministic.

However, we are interested in some measure normalized for one symbol. For this purpose we use the entropy per 1 symbol:

$$h_n = \frac{H_n}{n} \qquad (10)$$

The desired entropy of the symbolic sequence is then:

$$h = \lim_{n \to \infty} h_n \qquad (11)$$

To gain a better picture of the internal structure of the sequence, we employed entropy spectra. They consist of entropies of sequence computed for different temperatures $T = 1/\beta$. Instead of $P_n(w)$, we will use

$$\tilde{P}_{n,\beta}(w) = \frac{P_n^\beta(w)}{\sum_{w \in a^n} P_n^\beta(w)} \qquad (12)$$

This transformed probability enables to reveal words which are more likely to take over the less frequent words when the temperature is positive and larger than 1 (i.e. $\beta < 1$). Opposite effect happens for negative temperatures ($\beta < 0$), when the least frequent words are dominant. Therefore entropies computed for different temperatures uncover more information about the histogram of the sequences. (A special case when $T \to \infty$, i.e. $\beta \to 0$ is called a topological case. In this case, all words, which are present in the input sequence, are represented in the histogram by 1, and all words, which are not present in the input sequence, by 0).

3   Results

The experiments with the linear neuron showed that the neuron is not very sensitive to differences in entropy of the input sequences (see Fig. 2). Tuning the parameters did not make the difference. Therefore, we switched to a nonlinear neuron with sigmoid transition function (equation 2), which improved the situation. In most cases, the neuron did recognize the differences in the input sequences and the resulting weight (after the convergent process) discriminated sequences with different entropies for $\tau \le 10$ (see Fig. 3).
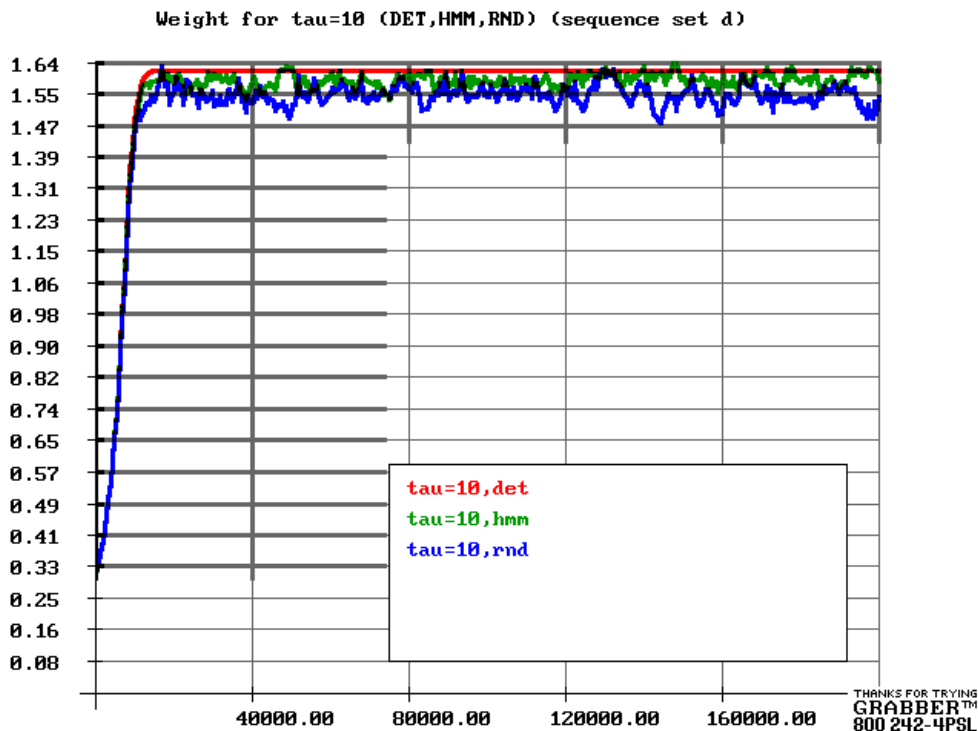

Fig. 2. Evolution of the weight for linear neuron We can see that the weight doesn't clearly disciriminate the sequences. We used the following parameters: learning speed $\eta=0.001$, $c_0=0.85$, length of sequence: 200 000 symbols. The entropy of det. sequence $\to$ 0, entropy of hmm sequence was approx. 0.9 and the entropy of rnd. sequence $\to$ 1.

To test the model robustness, we have introduced a noise parameter. A randomly distributed additive noise (not Gaussian) on the input did have only a quantitative, not the qualitative impact on the model. The sequence discrimination was not as clear as in the case without the noise, which can be explained by changing the properties of the input sequence. HMM generated sequence is already "quite" random and adding noise results in a sequence, which cannot be represented by the original HMM automaton and therefore has different properties. Because the noise didn't influence the quality of the model, we removed it from further experiments.

The shape of sigmoid function drives the degree of nonlinearity and has a major influence on the final point of weight convergence. For a nonlinear neuron, we used the sigmoid with the asymptotic values of –2 and 2. Also the θ_M scaling constant c_0 is used to drive the process of weight evolution. It is usually smaller than 2 (we used $c_0 = 0.85$). Increasing c_0 makes the ϕ parabola steeper and the input weight less stable. Stability of the asymptotic input weight is guaranted by using very small learning speed η. A reasonable stability is achieved for value 0.001, which we used in the experiments.
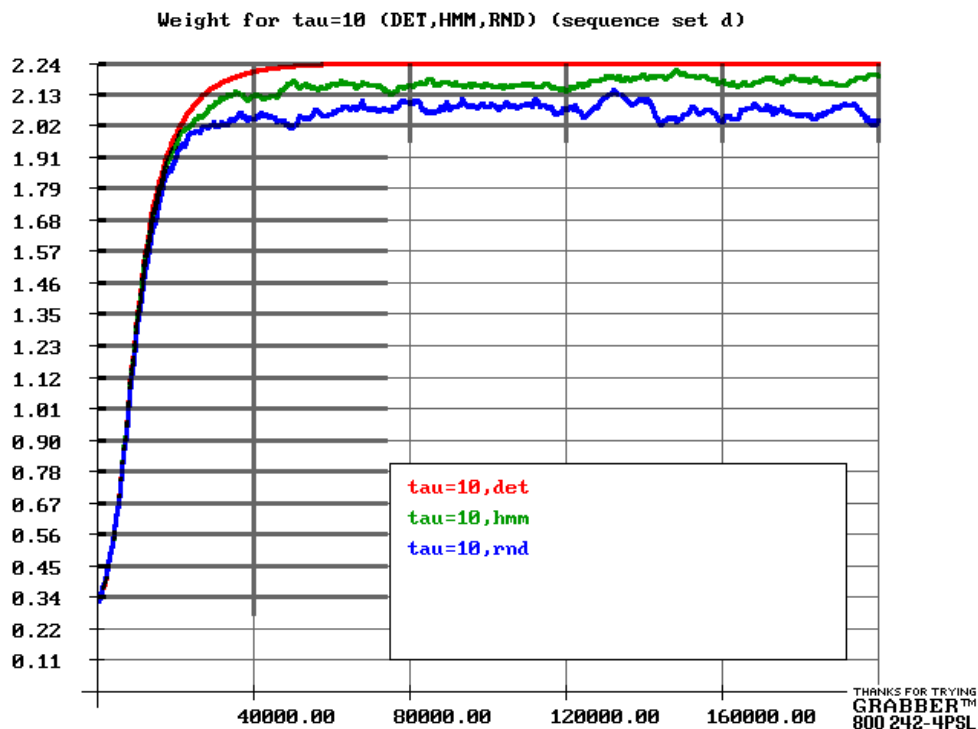

Fig. 3. Evolution of the weight for nonlinear neuron. All other parameters in this experiment were the same as for the linear case in Fig. 2. Asymptotic values for the sigmoid were –2 and 2.

Dynamic changes in the input are compensated for by the dynamic threshold θ_M. Increased potentiation in the weight results in increasing the average response and in increasing of θ_M. The long row of symbols '0' ('1') in input results in decreasing (increasing) the average neuron response and also in decreasing (increasing) the threshold θ_M. In this way, the dynamic threshold θ_M adapts to the input. We should also note that although the probability of '1' is the important factor for the resulting converged weight, it is not the only one. All sequences in one set contained the same number of symbols '1', but the resulting weights are different for a nonlinear neuron. To find another properties of the input time sequences, which might be significant, we have employed entropy spectra.

It holds that the sequences with more complicated internal structures have higher entropy. The transformed entropies for high temperatures show the structure with respect to very likely words. Entropies for low temperatures show the structure with respect to very unlikely words. A special case for T→ ∞ is called topological entropy and determines how many different words are in the sequence. Deterministic sequences have constant and low entropies for all temperatures, because they consist of a periodical subsequence repeated many times. On the other hand, random sequences contain all possible words for a given window length, although not periodically but randomly distributed. Therefore the entropy is high. We computed the entropy spectra up to the length of window = 12, due to the computational complexity and huge amount of data. It can be also observed for completely random sequences (Bernoulli source) that the entropy decreases for high temperatures and this is caused only by the bounded length of the sequence (the number of words for large window is high) – we worked with sequences comprised of 200,000 symbols.
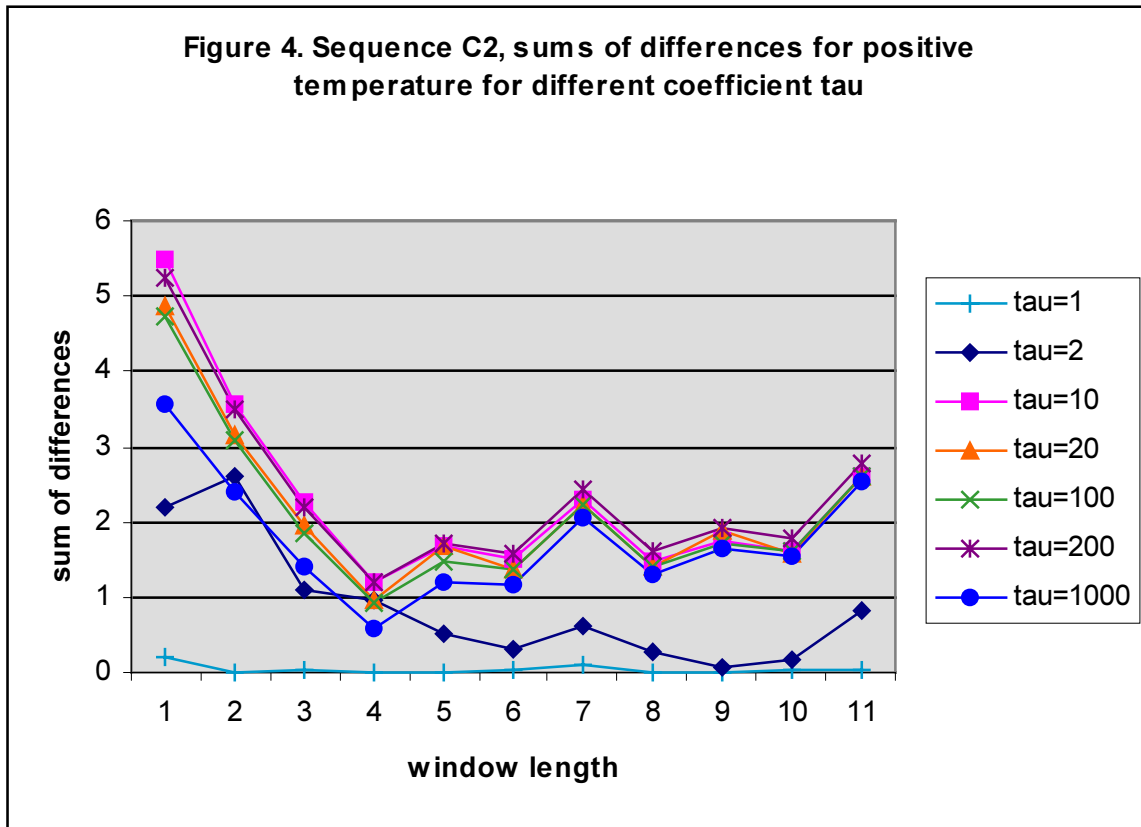
Fig. 4. Entropic spectra distances for hmm sequence, sums of differences for temperatures 1,2,3,5,10,20,50,100.
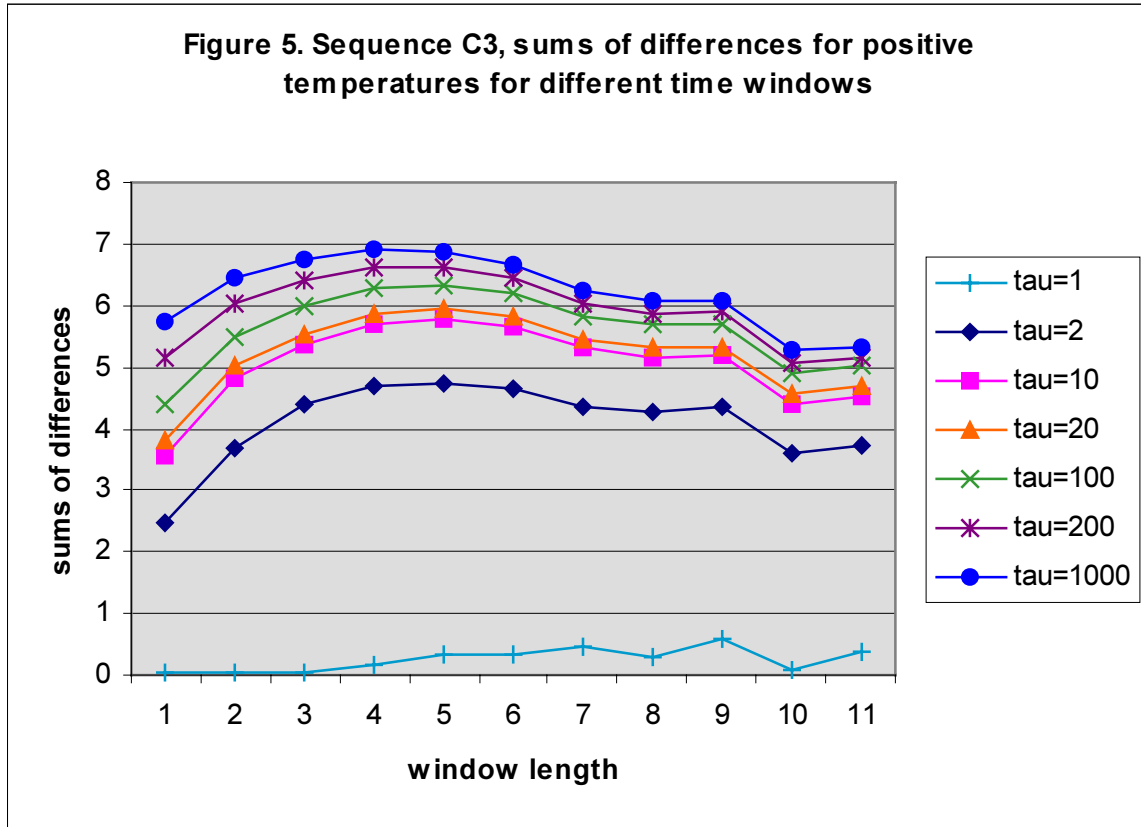

Fig. 5. Entropic spectra distances for rnd sequence, sums of differences for temperatures 1,2,3,5,10,20,50,100.

Our results reveal that the symbolic sequence derived from the θ_M differences from expected value is most similar to the original sequence when $\tau \leq 2$ (Fig. 4). This supports the previous findings that the weight of the input connection discriminates the sequences best in the case of $\tau = 2$.

Second part of the results discusses the entropy spectra of sequences created by comparing the actual value of the threshold θ_M with its long-term average. We computed entropy spectra for fixed length of the windows and for different values of τ. Our goal was to find some kind of correlation between the structure of the sequence, length of the window used for computing the entropy, and the τ parameter (see Fig. 4). When exploring Fig. 4, the original sequence should be observed first. We present the differences of the entropy values in entropy spectra for diffrent lengths of the window for the sequences C2 and C3. The characteristic property of the 4-state C2 HMM automaton is that when the state 2 is achieved, the automaton always generates sequence '110'. Therefore the frequency of words containing the sequence '110' is relatively high.

4   Conclusions

We have investigated the BCM learning rule in experiments with time sequences for both linear and nonlinear transition functions. We have shown that the asymptotic BCM neuron's response depends on the internal structure of the input sequence. All sequences, i.e. deterministic, HMM and random in one set contained the same number of symols '1', but they differed in the complexity of the internal structure. Our measure of this complexity was the value of entropy. We have found that for two sequences with different entropies and the same number of symbols '1', the resulting weight of the BCM neuron differs, i.e. it is dependent on the internal structure of the input sequence. We have supported this statement by exploring the properties of both input symbolic sequences and symbolic sequences derived by comparing the dynamic threshold θ_M to its long-term average. We computed the entropy spectra for the input sequences and for the θ_M sequences. Here, we have found that the both nonlinear and linear BCM neuron with the shortest "internal memory", i.e. $\tau \leq 2$, is able to model the input sequence most closely. We discussed the role of the parameters of the model and demonstrated that and how the nonlinearity can improve the characteristics of the development of converged input weight. Thanks to the entropy spectra, a better view of the internal structure of sequences can be obtained.

The usual tool for modeling symbolic time sequences are recurrent neural networks. The difficulty lies in building the BCM learning rule for the recurrent connections. The BCM rule has been successfully modified for several complicated architectures, some containing also recurrent connection [1]. The challenge to develop and study recurrent BCM architectures remains for the future work.

Acknowledgements

References

[1] Bachman C.M., Musman S.A., Luong D. & Shultz A.: Unsupervised BCM projection pursuit algorithms for classification of simulated radar presentations. *Neural Networks* 7, 709−728 (1994).

[2] Beňušková Ľ., Diamond M.E. & Ebner F.F.: Dynamic synaptic modification threshold: computational model of experience-dependent palasticity in adult rat barrel cortex. *Proc. Natl. Acad. Sci. USA* 91, 4791−4795 (1994).

[3] Bienenstock E.L., Cooper L.N & Munro P.W.: Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *J. Neuroscience* .2, 32−48 (1982).

[4] Fyfe C.A.: Comparative study of two neural methods of exploratory projection pursuit. *Neural Networks* 10, 255−262 (1997).

[5] Intrator N. & Cooper L.N.: Objective function formulation of the BCM theory of visual cortical platicity: statistical connections, stability conditions. *Neural Networks* 5, 3−17 (1992).

[6] Katok A. & Hasselblatt B.: *Introduction to the Modern Theory of Dynamical Systems*. Cambridge University Press: Cambridge (1995).

[7] Rabiner R.L. & Juang B.H.: An introduction to hidden markov models. *IEEE ASSP Magazine* 3, 4−16 (1986).

[8] Tiňo P. & Köteles M.: Modeling complex symbolic sequences with neural and hybrid neural based systems. S*ubmitted to IEEE Transactions of Neural Networks* (1996).