FAST AND ENERGY EFFICIENT LEARNING ALGORITHM FOR KOHONEN NEURAL NETWORK REALIZED IN HARDWARE

Marta KOLASA*

*Institute of Electrical Engineering, Faculty of Telecommunication and Electrical Engineering, University of Technology and Life Sciences, ul. Kaliskiego 7, 85-796, Bydgoszcz, Poland

markol@utp.edu.pl

Abstract: A new fast energy efficient learning algorithm suitable for hardware implemented Kohonen Self-Organizing Map (SOM) is proposed in the paper. The new technique is based on a multistage filtering of the quantization error. The algorithm detects such periods in the learning process, in which the quantization error is decreasing (the 'activity' phases), which can be interpreted as a progress in training, as well as the 'stagnation' phases, in which the error does not decrease. The neighborhood radius is reduced by 1 always just after the training process enters one of the 'stagnation' phases, thus shortening this phase. The comprehensive simulations on the software model (in C++) have been carried out to investigate the influence of the proposed algorithm on the learning process. The learning process has been assessed by the used of five criteria, which allow assessing the learning algorithm in two different ways i.e., by expressing the quality of the vector quantization, as well as the topographic mapping. The new algorithm is able to shorten the overall training process by more than 90% thus reducing the energy consumed by the SOM also by 90%. The proposed training algorithm is to be used in a new high performance Neuroprocessor that will find a broad application in a new generation of Wireless Body Area Networks (WBAN) used in the monitoring of the biomedical signals like, for example, the Electrocardiogram (ECG) signals.

Keywords: Kohonen Neural Network, CMOS Implementation, WBAN, Optimized Learning Process, Low Energy Consumption

1. INTRODUCTION

In the literature one can notice many attempts to employ artificial neural networks (ANNs) in the analysis of biomedical signals, including ECG signals (Chudáček et al., 2009; Fernández et al., 2001; Lagerholm and Peterson, 2000; Leite et al., 2010; Osowski and Linh, 2001; Talbi et al., 2010; Tighiouart et al., 2003; Valenza et al., 2008; Wen et al., 2009). These attempts aim to develop such methods and tools that will enable automatic analysis of the biomedical signals thus aiding medical staff in their work. One of the significant directions is to enable a quick detection of atypical sequences in such signals that usually indicate various problems. One of the main problems encountered in this area is that all applications of the ANNs involve PC computers or other programmable devices and as such are not suitable for the application in the Wireless Body Area Networks (WBAN).

The author of the paper is going to develop a new ultra-low energy consumption ANN realized as a specialized CMOS chip – a Neuroprocessor. The proposed chip will find the application in modern medical diagnostics tools based on WBAN systems. The chip will offer advanced data processing and analysis abilities directly in particular sensors (nodes) of the WBAN. As a result, instead of using a battery that enlarges the sizes of the sensor, an alternative supply source based on the energy scavenged from the environment (e.g. the body heat) will be used. This, in turn, will allow miniaturization of the sensors, making the overall wearable system much more convenient for the patients than the systems offered on the market today.

Due to the rapid growth in this research area a variety of learning algorithms and the architectures of the ANNs have been invented. Looking from the hardware realization point of view of such networks the most interesting solutions are those offering relatively simple learning algorithms. In this case, as simple arithmetic operations are being used, the algorithms require significantly less hardware resources. The resultant chips dissipate less power and occupy less chip area and thus are much more suitable for the application in the WBAN. A very simple and simultaneously fast learning algorithm is offered by the Kohonen Self-Organizing Map (SOM). This algorithm requires only basic arithmetic operations like addition, subtraction and multiplication. The Kohonen SOM is commonly used in the analysis and classification of the ECG signals (Leite et al., 2010; Tighiouart et al., 2003; Valenza et al., 2008; Wen et al., 2009). In case of the classification tasks, the reported results for this type of the ANN are comparable or even better than the results achieved in the case of using other algorithms (Chudáček et al., 2009; Fernández et al., 2001; Lagerholm and Peterson, 2000; Osowski and Linh, 2001; Talbi et al., 2010; Tighiouart et al., 2003; Valenza et al., 2008). In case of the analysis of the ECG signals the reported efficiency of even 97% is possible for the number of neurons not exceeding 150.

The Kohonen SOM already found the application in a wearable system that enables analysis of the acquired data in the real-time (Valenza et al., 2008). This system is able to recognize most significant cardiac arrythmias. The system is based on the Master Processing Unit (MPU) realized as the off-the-shelf SoC with the analog ECG signal conditioning circuit. The efficiency and sensitivity reported in (Valenza et al., 2008) are at high level of up to 99 %.

The author of the paper recently designed programmable architecture of the SOM that is able to operate with different topologies of the SOM and different neighborhood functions on a single chip. The author proposed a fully parallel and asynchronous neighborhood mechanism that independently on the sizes of the map, allows for determining the distances from the winning neuron to all neighboring neurons in the period less than 11 ns. The adaptation is then performed also in parallel in all neurons covered by the neighborhood range in a given learning cycle (Długosz et al., 2011; Kolasa et al., 2012).

This paper presents one of the very important steps in the overall design process of the new chip - a new learning algorithm suitable for low power ANNs realized in hardware. The new algorithm enables shortening the overall learning process of the SOM even by 90% thus reducing the energy consumption also by 90%.

2. KOHONEN NEURAL NETWORK

Teuvo Kohonen in 1975 proposed a new class of neural networks that use competitive unsupervised learning algorithms (Kohonen, 2001). His neural networks (KNNs) in their classical approach, also called self-organized map (SOM), contain one layer of neurons that form a map. The number of the outputs of the network equals the number of neurons, while all neurons have common inputs, whose number depending on the application can vary in-between two and even several dozen. SOMs are used in data visualization and analysis (Boniecki, 2005; Brocki, 2007; Mokriš and Forgáč, 2004).

The competitive unsupervised learning in KNNs relies on presenting the network with the learning vectors X in order to make the neurons' weight vectors W resemble presented data. For each training vector X KNN determines Euclidean distances (d_{EUC}) between this vector and the weights vectors W in each neuron, which for n network's inputs are calculated using the following formula:

$$d_{\mathrm{EUC}}(X, W_i) = \sqrt{\sum_{l=1}^{n}\left(x_l - w_{i,l}\right)^2} \quad (1)$$

The neuron, whose weights are the most similar to the training vector X becomes a winner and is allowed to adapt own weights. Two general types of such networks can be distinguished. In the Winner Takes All (WTA) approach only the winning neuron is allowed to adapt the weight, while in the Winner Takes Most (WTM) algorithm also neurons that belong to the winner's neighborhood are allowed to adapt the weights, according to the following formula:

$$W_j(l+1) = W_j(l) + \eta(k)G\big(R, d(i,j)\big)[X(l) - W_j(l)] \quad (2)$$

where η is a learning rate that control strength of the learning algorithm, W_j denotes the weights' vector of a given j^{th}, neuron, and $X(l)$ is a given input pattern in the l^{th} cycle. Particular neurons that belong to the winner's neighborhood are adapted with different intensities, whose values depend on the neighborhood function $G()$. The commonly used neighborhood functions are: rectangular and Gaussian neighborhood function. Different neighborhood functions were defined by the author in (Kolasa, 2012).

One of the important parameters is the network topology, which can be defined as a grid of neurons. This feature determines which neurons belong to the winner's neighborhood for a given value of the radius R (Boniecki, 2005; Kohonen, 2001; Mokriš and Forgáč, 2004). The commonly used topologies are: a hexagonal one (Hex) in which particular neurons have maximum six neighbors and a rectangular with four (Rect4) and eight (Rect8) neighbors.

The quality of the learning process can be evaluated by means of the quantization error (Q_{err}) and the topographic error (E_{T1}), which are a commonly used criterias in such cases. In this paper the effectiveness of the learning process of the SOM is evaluated on the basis of five criteria described in (Lee and Verleysen, 2002). The quantization error is defined as:

$$Q_{\mathrm{err}} = \frac{\sum_{j=1}^{m}\sqrt{\sum_{l=1}^{n}(x_{j,l} - w_{i,l})}}{m} \quad (3)$$

where m is the number of learning patterns in the input data set, n *is* the number of the network inputs, while i identify the winning neuron. This criterion illustrates a way of fitting of the map to input data (Uriarte and Martin, 2005). A second measure used to assess the quantization quality is the percentage of dead neurons (PDN), which tells us about the ratio of inactive (dead) neurons versus all neurons. Dead neurons are those neurons that never won the competition and as such have not become representatives of any input data. These errors are detrimental to the assessment of the topological order of the map.

The quality of the topographic mapping is assessed using three measures (Lee and Verleysen, 2002). The first one is the Topographic Error E_{T1}, which is defined as follows:

$$E_{\mathrm{T1}} = 1 - \frac{1}{m}\sum_{h=1}^{m}\lambda(X_{\mathrm{h}}) \quad (4)$$

This is one of the measures proposed by Kohonen (Kohonen, 2001; Uriarte E. and Martin F., 2005). The value of $\lambda(X_h)$ equals 1 when for a given pattern X two neurons whose weight vectors that resemble this pattern to the highest extent are also direct neighbors in the map. Otherwise the value of $\lambda(X_h)$ equals 0. The lower the value of E_{T1} is, the better the SOM preserves the topology (Beaton et al., 2010; Uriarte and Martin, 2005). In an ideal case, the optimal value of E_{T1} equals 0.

The remaining two measures of the quality of the topographic mapping do not require the knowledge of the input data. In the second criterion, in the first step, the Euclidean distances between the weights of an ρ^{th} neuron and the weights of all other neurons are calculated. In the second step, it has to be check if all p direct neighbors of neuron ρ are also the nearest ones to this neuron in the sense of the Euclidean distance measured in the future space. To express this requirement in a formal manner, let us assume that neuron ρ has $p = |N(\rho)|$ direct neighbors, where p depends on type of the map topology. Let us also assume that function $g(\rho)$ returns the value equal to the number of the direct neighbors that are also the closest to neuron ρ in the feature space. As a result, the E_{T2} criterion for P neurons in the map can be defined as follows:

$$E_{\mathrm{T2}} = \frac{1}{P}\sum_{\rho=1}^{P}\frac{g(\rho)}{|N(\rho)|} \quad (5)$$

The optimal value of E_{T2} equals 1. Considering the third criterion, it is built around each neuron ρ a neighborhood in the feature space (Euclidean neighborhood) defined as a sphere with the radius:

$$R(\rho) = \max_{s \epsilon N(\rho)}\|W_\rho - W_s\| \quad (6)$$

where W_ρ are the weights of a given neurons ρ, while W_S are the weights of its particular direct neighbors. Then it is necessary to count those neurons, which are not the closest neighbors of the neuron ρ, but are located inside $R(\rho)$. The E_{T3} criterion, with the optimal value equal to 0, is defined as follows:

$$E_{\mathrm{T3}} = \frac{1}{P}\sum_{\rho=1}^{P}\left|\{s | s \neq \rho, s \notin N(\rho), \|W_\rho - W_s\| < R(\rho)\}\right| \quad (7)$$

3. THE PROPOSED ALGORITHM

In the Kohonen learning algorithm it is often assumed that the neighborhood range R_{max}, which is the maximal neighborhood range R set up before starting the learning process, should cover at least half of the map (Kohonen, 2001) and then gradually decrease to zero. The reduction of the value of this parameter can be realized in the following manner:

$$R_k = 1.00001 + (R_{\max} - 1) \cdot (1 - \frac{k}{l_{\max}}) \tag{8}$$

where k stands for the k^{th} iteration, l_{max} is the total number of the iterations in the ordering phase of the learning process.

In practice, as number of iterations usually is much larger than the maximum value (R_{max}) of the neighborhood radius R, therefore the radius decreases always by '1' after the number of iterations equals to:

$$l = \mathrm{round}(\frac{l_{\max}}{R_{\max}}) \tag{9}$$

Value of the l parameter usually is in the range in-between 20 and 200, depending on dimensions of the map. In case of an example map with 15x15 neurons, R_{max} equals 29 or 14, for Rect4 and Rect8 topologies, respectively.

3.1. Applied methodology

The author completed a series of simulations using the software model (in C++) of the map to verify the commonly used 'linear' approach. Simulations have been carried out for all three topologies (Hex, Rect4 and Rect8), sizes of the map varying in-between 4x4 and 64x64 neurons, different numbers of inputs, different values of the initial neighborhood size, R_{max}, different neighborhood functions (rectangular, triangular and Gaussian) and different training sets. The network was trained with 2D and 3D data regularly placed in the input space, as well as with data randomly distributed in this space. Here the author reports on some selected results which can be regarded as being representative to the overall suite of experiments. The author presents results for 8x8 and 16x16 neurons for two example 2D data sets. The results for 2D sets have been selected for a better illustration (Lee et al., 2001; Lee and Verleysen, 2002; Su, 2002; Uriarte and Martin, 2005). In the first data set, data are divided into P classes (centers), where P equals the number of neurons in the map. Each center is represented by an equal number of learning patterns. The centers are placed uniformly in the input data space, as shown in Fig. 1a. This data set is in the paper called CREG. To achieved comparable results in this case, the input space was fitted to input data. For example, for the map with 8x8 neurons the values of the input signals were in the range of 0 to 1, while for 16x16 neurons the values was in the range of 0 to 2. As a result, in all cases the optimal value of Q_{err} equals 16.2e-3, while the optimal values of the remaining parameters (PDN/E_{T1}/E_{T2}/E_{T3}) are equal to 0/0/1/0, respectively. The optimal nonzero value of Q_{err} results from the arrangement of data. The regular arrangement allows for ideal distribution of all neurons over the input data space, assuming the training process was optimal. This approach facilitates a direct comparison of the results for different combinations of particular parameters mentioned above (Li, 2009).

Second data set was composed of 1000 patterns randomly distributed over the selected region, as shown in Fig. 1b. This data set is called SQUARE in the paper. In this case input data are in the constant range, independently from the size of the map. As a result, for larger maps the Q_{err} achieves smaller values.

3.2. Selected simulation results

Selected simulation results illustrating an example learning process are shown in Fig. 2. The Figure presents example illustrative waveforms of the Q_{err} over time i.e. for particular iterations. The results are shown for example maps with 8x8 and 16x16 neurons, the triangular neighborhood function and Rect8 topology, but similar results were commonly observed for different input data, network topologies and neighborhood functions.

Observing the quantization error in time domain one can notice that the 'linear' approach is not optimal.

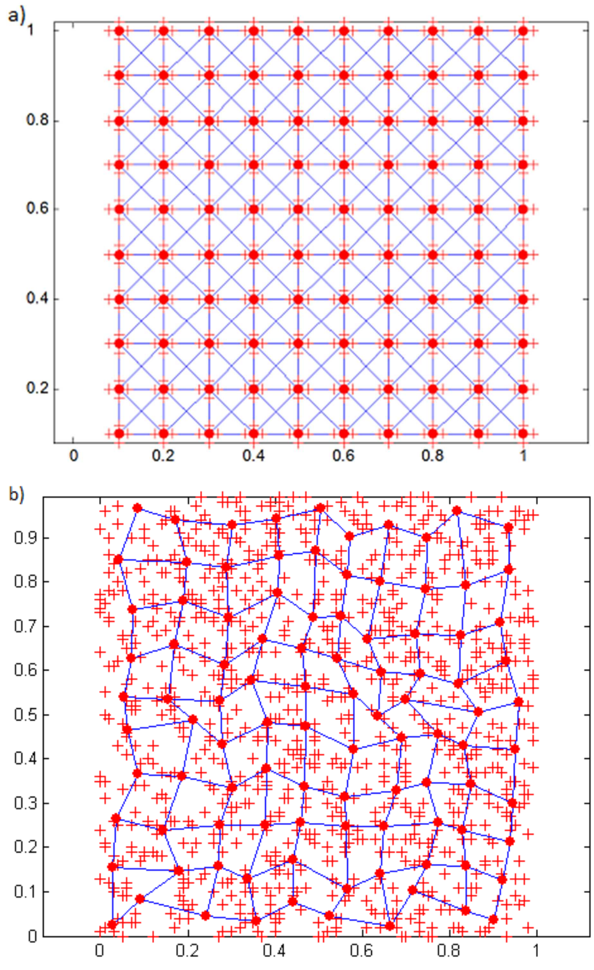

Fig. 1. Input data sets and the final placement of neurons for: 2D data a) regularly; b) randomly distributed in the input space

The first important observation is that when the neighborhood radius R is larger than some *critical* value, the quantization error does not decrease, so in this period the network does not make any progress in training. For example, in diagram (b), for R_{max}=6, the Q_{err} starts decreasing only around the 800th iteration, for R = 2 i.e. for about 1/3 of the map size. So the conclusion is that the learning process may start with the value of the radius R, which is smaller than the maximal value R_{max}. This significantly shorts the overall training process.

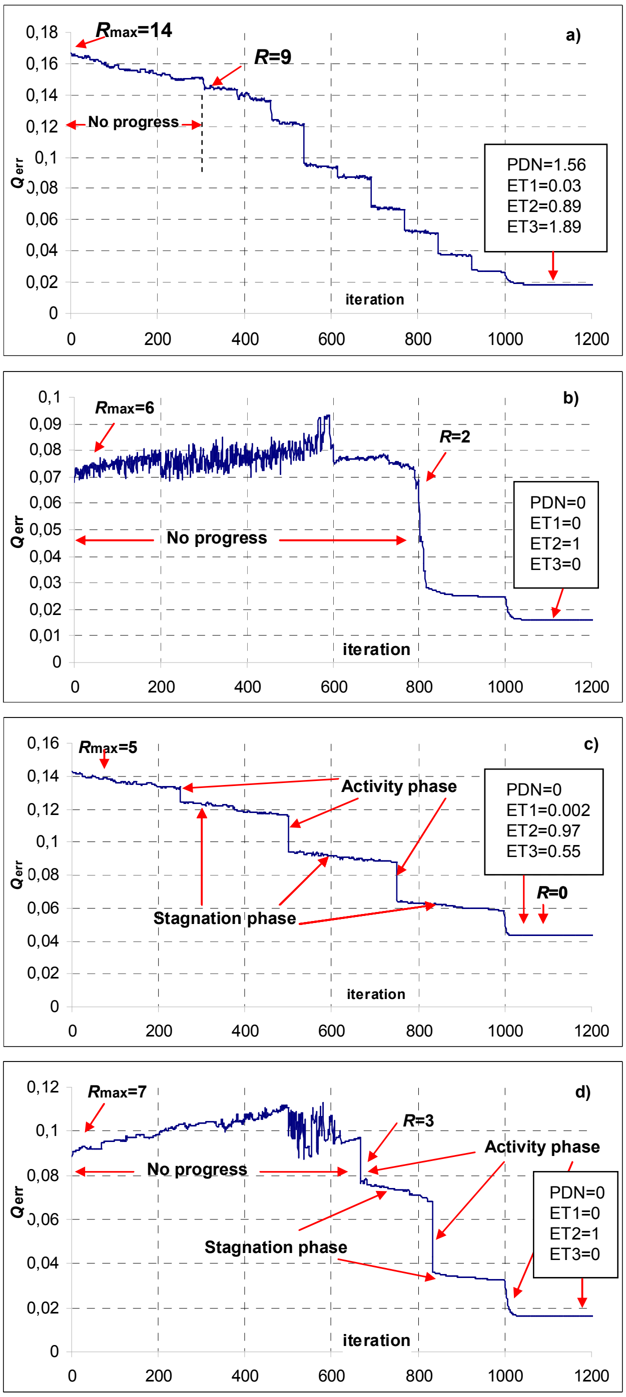

Fig. 2. The quantization error as a function of the number of iterations, for: a) 16x16 SQUARE; b) 16x16 CREG2D; c) 8x8 SQUARE; d) 8x8 CREG2D data file

The main observation is that the quantization error Q_{err} does not decrease monotonically during the overall learning process. One can notice some distinct ‘activity’ phases, in which the error decreases rapidly and then the ‘stagnation’ phases, in which the value of the error remains almost constant. The activity phases take place immediately after the radius *R* is switched to a smaller value. Note that the stagnation phases usually are much longer than the activity phases, which in practice means that the network makes a progress in training only in short periods of the overall process.

The algorithm proposed in this paper relies on shortening the stagnation phases. First it is necessary to detect automatically the activity and the stagnation phases, which is performed by the use of a set of linear and nonlinear filters. Such a multistage filtering of the quantization error detects the activity phases and controls the neighborhood radius *R* in such a way to significantly shorten the stagnation phases.

This technique uses a special decision mechanism that automatically switches over the radius *R* just after a given activity phase is finished. This starts a new activity phase, but for the new, smaller value of the radius *R*. As a result, the learning process may be even 90% faster than in the classic approach, in which the radius *R* decreases linearly.

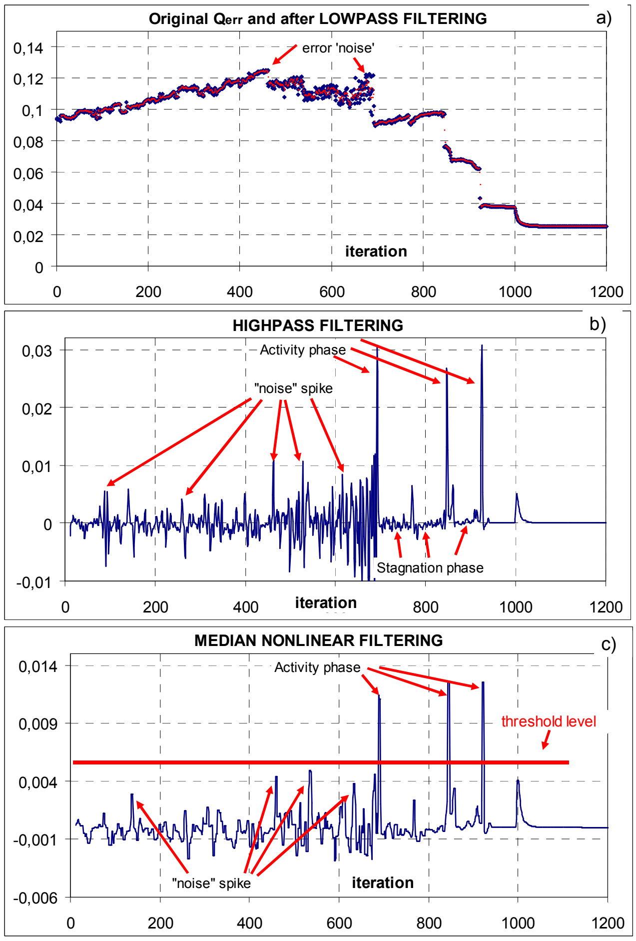

Fig. 3. Proposed 3-stage error filtering: a) the original waveform and the lowpass, b) the highpass, c) the nonlinear median filtering for 16x16 CREG2D data file

3.3. The proposed technique

The proposed 3-stage filtering of the error is presented in Fig. 3 for an example map with 16x16 neurons, triangular neighborhood function, Rect8 topology and CREG data base. In this case three filters have been used. The process of detection of the activity phases starts with a lowpass finite impulse response (FIR) filtering that removes the “noise” from the initial error waveform. This process is shown in Fig. 3a. In this case a simple

Butterworth flat filter has been used with the following coefficients: h_{LPi} = {0.125, 0.375, 0.375, 0.125}.

The next step is the highpass filtering operation that detects edges in the smoothed error waveform. This filter can be very simple, with the length not exceeding 4. In presented example a filter with the coefficients h_{HPi} = {1, 1, -1, -1} has been employed. The resultant waveform is illustrated in Fig. 3b. The spikes in this waveform indicate the activity phases. The problem here is that the "noise" present in the initial error waveform is a source of additional undesired spikes, which often are as high as the 'activity' spikes, although usually are narrower than the 'activity' spikes. To overcome this problem a nonlinear median filter has been additionally applied. The length of this filter has been selected in such a way to even the height of the 'activity' spikes and to eliminate the 'noise' spikes. An example median filter of the length 5 allows to eliminate the 'noise' spikes with the width equal or smaller than 2, as illustrated in Fig. 3c.

The output signals of the highpass and the median filters are used by a decision mechanism that automatically switches over the radius R to smaller values. This procedure starts when the value at the output of the median filter becomes larger than a selected threshold value, which must be high enough to exclude the 'noise' spikes. Switching of the R parameter is performed when the signal at the output of the highpass filter starts falling that means that the training process is just entering the stagnation phase.

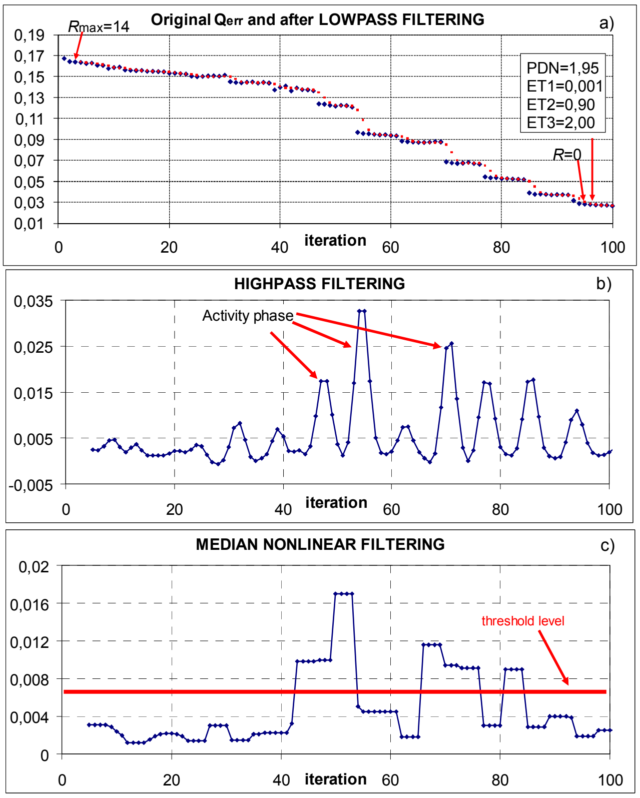

Fig. 4. Proposed 3-stage error filtering for the training process after optimization: a) the original waveform and the lowpass, b) the highpass, c) the nonlinear median filtering for 16x16 SQUARE data file

The proposed algorithm work good with all investigated cases of neighborhood functions and network topologies. It is worth noticing that the proposed algorithm must cooperate with the classic 'linear' method. This is necessary in a situation, in which an 'activity' spike at the output of the median filter would be to small to activate the decision procedure. In this case the 'linear' method will switch over the radius R after l iterations that will stop a given stagnation phase.

3.4. Performance analysis of the proposed solution

Illustrative simulation results in case of the optimized training process are shown in Fig. 4 for an example network with 16x16 neurons for SQUARE data set. In this case the entire training process has been shorten 10 times from initial 1000 iterations to 100 iterations.

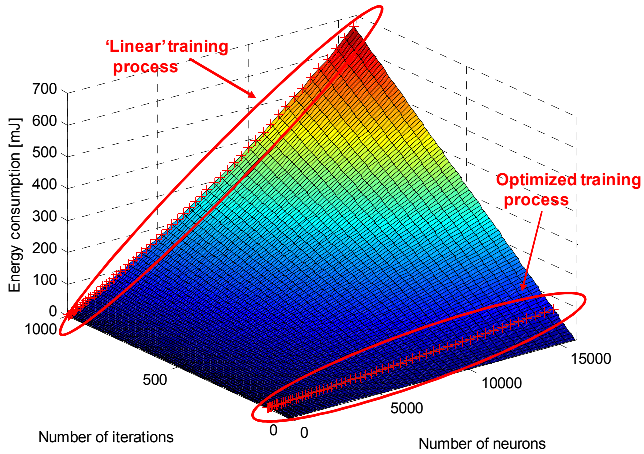

Fig. 5. Estimated energy consumption regarded as a function of the number of neurons and the number of learning iterations

An important parameter is also the energy consumption of a neuron. The simulations of the overall SOM performed in Hspice environment show that a single neuron consumes 25–30 pJ per a single pattern $X(l)$ for the CMOS 0.18µm process (Długosz R. et al., 2011). Fig. 5 shows the estimated energy consumption as a function of the number of neurons and the number of learning iterations for data set comprised of 1000 learning patterns in the worst case scenario, i.e. for the neighborhood range R covering the entire map. In practice, as the values of R are usually small, the energy consumption will be smaller. For an example map with 16x16 neurons the energy consumption is equal to 7.7nJ during presentation of a single input pattern. When the number of iteration is equal 1000 and data set is composed of 1000 patterns, the energy consumption will be equal to 7.7mJ. In case when the overall training process will be shorten 10 times from initial 1000 iterations to 100 iterations the energy consumption wil be equal to 0.77mW. So in case of the optimized training process the energy consumption will be also 10 times smaller. As we can see, one of the main advantages of hardware implemented network is a very low energy consumption. For the comparison, a single software-model test during which the SOM processes 2 million input patterns takes about 20 minutes. Based on the assumption that the power dissipation of a PC computer equals 250W, the energy consumption will equal 300kJ in this case. As a result, the achieved energy consumption in case of hardware implemented neural network is more than eight orders of magnitude smaller than in case of a similar network realized on PC.

Note that these results have been obtained for the CMOS 0.18μm process. For the latest technologies below 65nm, the author expects a substantial improvement of the results.

4. CONCLUSIONS

A new simple learning algorithm for the WTM Kohonen SOM designed for low-power devices has been described in the paper. The proposed technique bases on the observation that the quantization error does not decrease monotonically during the learning process, but there are some activity phases, in which this error decreases very fast and then the stagnation phases, in which the error does not decrease.

The proposed technique using a set of linear and nonlinear filters detects the activity phases and controls the neighborhood R in such a way to shorten the stagnation phases. As a result, the learning process may be more than 10 times faster and more than 10 times energy efficient than in the classic approach, in which the radius R decreases linearly.

The intended application of the proposed solution will be in Wireless Body Area Networks in the classification and analysis of the EMG and the ECG biomedical signals.

REFERENCES

1. **Beaton D., Valova I., MacLean D.** (2010), CQoCO: A measure for comparative quality of coverage and organization for self-organizing maps, *Neurocomputing*, Vol. 73, 2147–2159.
2. **Bolkowski S., Stabrowski M., Skoczylas J., Sroka J., Sikora J., Wincenciak S.** (1993), *Computer analysis methods of electromagnetic field*, WNT, Warsaw.
3. **Boniecki P.** (2005), The Kohonen neural network in classification problems solving in agricultural engineering, *Journal of Research and Applications in Agricultural Engineering*, Vol. 50, No. 1, 37-40.
4. **Brocki L.** (2007), *Recent Advances in Mechatronics*, Springer Berlin-Heidelberg.
5. **Chudáček V., Georgoulas G., Lhotská L., Stylios C., Petrík M., Čepek M.** (2009), Examining cross-database global training to evaluate five different methods for ventricular beat classification, *Physiological Measurem,* Vol. 30, No. 7, 661-677.
6. **Długosz R., Kolasa M., Pedrycz W., Szulc M.** (2011), Parallel Programmable Asynchronous Neighborhood Mechanism for Kohonen SOM Implemented in CMOS Technology, *IEEE Transactions on Neural Networks*, Vol. 22, No. 12, 2091-2104.
7. **Fernández E.A., Willshaw P., Perazzo C.A., Presedo R.J., Barro S.** (2001), Detection of abnormality in the electrocardiogram without prior knowledge by using the quantisation error of a self-organising map, tested on the European ischaemia database, *Medical and Biological Engineering and Computing*, Vol. 39, No. 3, 330-337.
8. **Kohonen T.** (2001), *Self-Organizing Maps*, third ed. Springer, Berlin.
9. **Kolasa M., Długosz R., Pedrycz W., Szulc M.** (2012), A programmable triangular neighborhood function for a Kohonen self-organizing map implemented on chip, *Neural Networks* ,Vol. 25, 146-160.
10. **Lagerholm M., Peterson G.** (2000), Clustering ECG complexes using hermite functions and self-organizing maps, *IEEE Transactions on Biomedical Engineering*, Vol. 47, No. 7, 838-848.
11. **Lee J. A., Donckers N., Verleysen M.**, (2001), Recursive learning rules for SOMs, *Workshop on Self-Organizing Maps, ser. Advances in Self-Organising Maps*, Springer Verlag, 67–72.
12. **Lee J., Verleysen M.** (2002), Self-organizing maps with recursive neighborhood adaptation, *Neural Networks*, Vol. 15, No. 8-9, 993–1003.
13. **Leite C.R., Martin D.L., Sizilio G.R., Dos Santos K.E., de Araujo B.G., Valentim R.A., Neto A.D., de Melo J.D., Guerreiro A.M.** (2010), Classification of cardiac arrhythmias using competitive networks, *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1386-1389.
14. **Li F., Chang C.-H., Siek L.** (2009), A compact current mode neuron circuit with Gaussian taper learning capability, *IEEE Int. Symp. Circuits and Systems*, 2129–2132.
15. **Mokriš, Forgáč R.** (2004), Decreasing the Feature Space Dimension by Kohonen Self-Orgaizing Maps, *2nd Slovakian – Hungarian Joint Symposium on Applied Machine Intelligence*, Herľany, Slovakia.
16. **Osowski S., Linh T.H.** (2001), ECG beat recognition using fuzzy hybrid neural network, *IEEE Transactions on Biomedical Engineering*, Vol. 48, No. 11, 1265-1271.
17. **Su M.-C., Chang H.-T., Chou C.-H.** (2002), A novel measure for quantifying the topology preservation of self-organizing feature maps, *Neural Process. Lett.*, Vol. 15, No. 2,137–145.
18. **Talbi M. L., Charef A., Ravier P.** (2010), Arrhythmias classification using the fractal behavior of the power spectrum density of the QRS complex and ANN, *International Conference on High Performance Computing and Simulation*, 399-404.
19. **Tighiouart B., Rubel P., Bedda M.** (2003), Improvement of QRS boundary recognition by means of unsupervised learning, *Computers in Cardiology*, Vol. 30, 49-52.
20. **Uriarte E., Martin F.** (2005), Topology Preservation in SOM, *Int. J. of Math. and Comput. Sci.*, Vol. 1, No. 1, 19–22.
21. **Valenza G., Lanata A., Ferro M., Scilingo E.P.** (2008), Real-time discrimination of multiple cardiac arrhythmias for wearable systems based on neural networks, *Computers in Cardiology*, Vol. 35, 1053-1056.
22. **Wen C., Lin T.-C., Chang K.-C., Huang C.-H.** (2009), Classification of ECG complexes using self-organizing CMAC, *Measurement: Journal of the International Measurement Confederation,* Vol. 42, No. 3, 399-407.