ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

Initialization Mechanism in Kohonen Neural Network Implemented in CMOS Technology

Tomasz Talaśka [1] and Rafał Długosz [2,3,*]

1- University of Technology and Life Sciences, Institute of Telecommunication,
ul. Kaliskiego 7, 85-791, Bydgoszcz, Poland
2- University of Neuchâtel, Institute of Microtechnology, Rue A.-L. Breguet 2,
CH-2000, Neuchâtel, Switzerland
3- University of Alberta, Department of Electrical and Computer Engineering
ECERF Building, Edmonton, T6G 2V4, Canada

Abstract. An initialization mechanism is presented for Kohonen neural network implemented in CMOS technology. Proper selection of initial values of neurons' weights has a large influence on speed of the learning algorithm and finally on the quantization error of the network, which for different initial parameters can vary even by several orders of magnitude. Experiments with the software model of designed network show that results can be additionally improved when conscience mechanism is used during the learning phase. This mechanism additionally decreases number of dead neurons, which minimizes the quantization error. The initialization mechanism together with experimental Kohonen neural network with four neurons and 3 inputs have been designed in CMOS 0.18 μm technology.

1 Introduction

Training efficiency in artificial neural networks depends on various parameters. One of important problems is an initial polarization before the learning phase, which has a direct influence on both the training speed and the final results [1-3]. One universal initialization method for any network type and any training data does not exist, which means that different heuristic methods usually must be used. Initial values of neurons' weights usually are selected experimentally. Typically a random and even distribution in a given range is used [4, 5], although this method not always is effective.

In this paper we focus on initialization problem in the winner takes all (WTA) Kohonen neural network (KNN) implemented in CMOS technology as analog circuit. In this network properly selected initial values of neurons' weights have influence on number so called dead neurons and finally on the quantization error of the network. Dead neurons usually occur in situations when initial values of neurons' weights and training data cover different areas in the data space.

One of effective initialization methods relies on setting up the initial values of neurons' weights as the first n data from the training vector X or as randomly selected n elements from this training vector. In other method a linear initialization may be used [6]. Different methods may be combined together into one hybrid method. In this case at the beginning of the learning process a random initialization allows for polarization of the network in selected narrow range, while in the next stage the

* this work is supported by EU Marie Curie OIF fellowship, project No. 21926

deterministic Convex Combination Method (CCM) [7] allows for some modifications of the training vector X to "bend" this vector toward neurons' weights. The hybrid method is more difficult in CMOS implementation as deterministic modification of analog input data requires additional circuitry, which can distort the input data.

To overcome this problem and to make the learning process more effective, in our hardware implemented KNN the programmable analog conscience mechanism has been additionally included in the circuit. This mechanism allows for elimination of dead neurons even when the initialization is not the most optimal [8÷11].

2 Initialization methods in software model of designed KNN

Various initialization methods for different distributions of the input data have been first verified in the software model of KNN. To make this model corresponding with the network designed latter in hardware, electrical parameters of particular building blocks such as for example transistors leakage have been included.

Example experimental results for random initialization are shown in Figs. 1-3 for different initial distributions of neurons' weights. The training vector X, shown in Fig. 1, contains 160 signals clustered in 16 areas of data space. This vector during the learning phase has been presented 40 times (40 epochs) to the network that in this case contains 16 neurons. Each neuron can theoretically represent one data class. The learning rate having initially a value of 0.9 after 20 epochs has been decreased to 0.1.

Weights' vectors before and after the learning phase for different initialization parameters are shown in Fig. 2. Initial weights' values have even distribution, which differ in position of the central point and in the drawing range. The final results differ in number of dead neurons, i.e. those which did not take part in the competition. In the first three cases (a)-(c) the conscience mechanism is not used and part of neurons remained dead in each case. The best results are in case (a) i.e. when initial values evenly cover entire data space. On the other hand, when the conscience mechanism was used (d) then the learning process was always more optimal even for not optimal setting of the initialization mechanism, causing the number of dead neurons to be zero in many cases. Properly selected parameters of the initialization mechanism and the conscience mechanism accelerate the adaptation process by several dozen %. This is shown in Fig. 3, which illustrates the quantization error (defined as a distance between the winning neuron and the input data) in particular cases from Fig. 2 as well as number of competitions won by particular neurons.

3 Initialization mechanism in CMOS implemented KNN

An experimental WTA KNN with four neurons, each having three weights (3 inputs and 4 outputs) has been designed in CMOS 0.18 μm technology. Designed network is in 90% an analog circuit, which enables a low power and parallel data processing. The initialization mechanism used in this network has been designed in accordance with results obtained in the software model described above. The network is initially sequentially programmed using one analog input and four digital addressing lines. An internal decoder converts an address delivered as a BCD code into 12 separate lines, which directly control analog demultiplexer that programs particular memory cells.

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

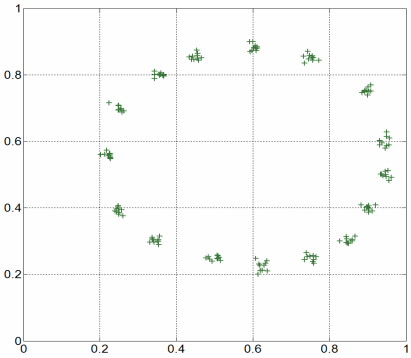
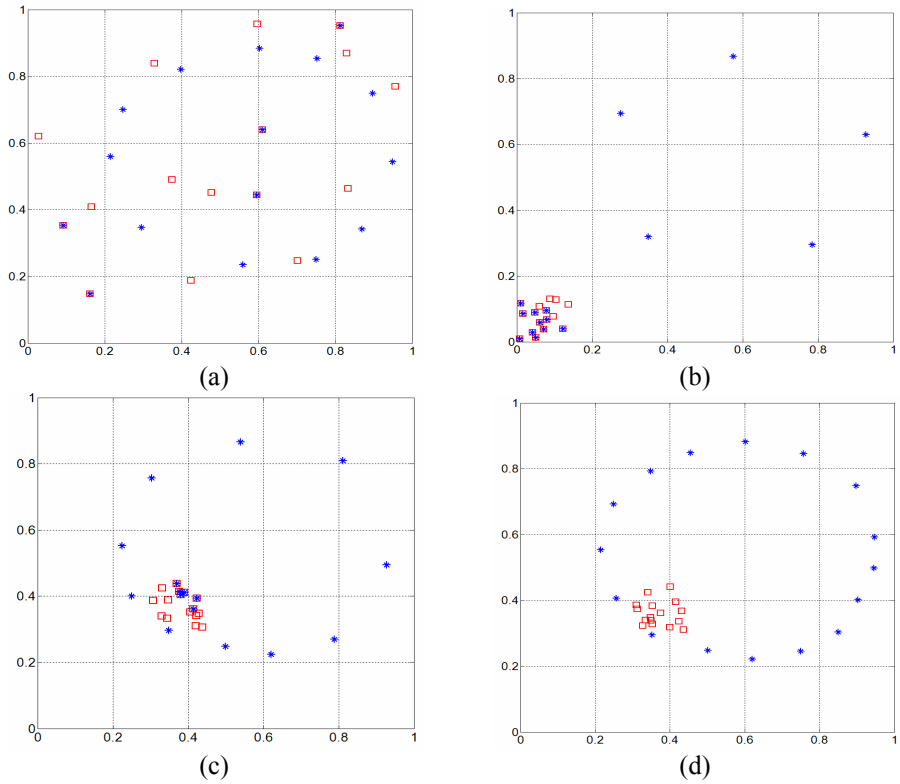


Fig. 1: Example training vector with 160 data clustered in 16 areas of data space.



(a) (b) (c) (d)

Fig.2: The network model's simulations when initial values of neurons' weights: (a) cover entire space; (b) do not cover the input data; (c) are within input data range, but are not evenly distributed; (d) like in (c) but when conscience mechanism is used.

Designed network works in the current mode. Although both the input signals and neuron weights are in calculations represented by currents, but these signals after calculations are stored as voltages across capacitors in current-mirror based sample & hold memory cells [12]. This solution has one important advantage. Input

transistors in current mirrors work in the diode configuration. In this case even when input signals vary in relatively wide range e.g. between 1 μA and 5 μA, voltages across storage capacitors vary only by about 10-20 %. As a result, writing errors that result from the charge injection effect are almost equal in wide range of the input signals. This allows for precise initial programming of the network and additionally the charge injection effect can be compensated with a constant correction factor.

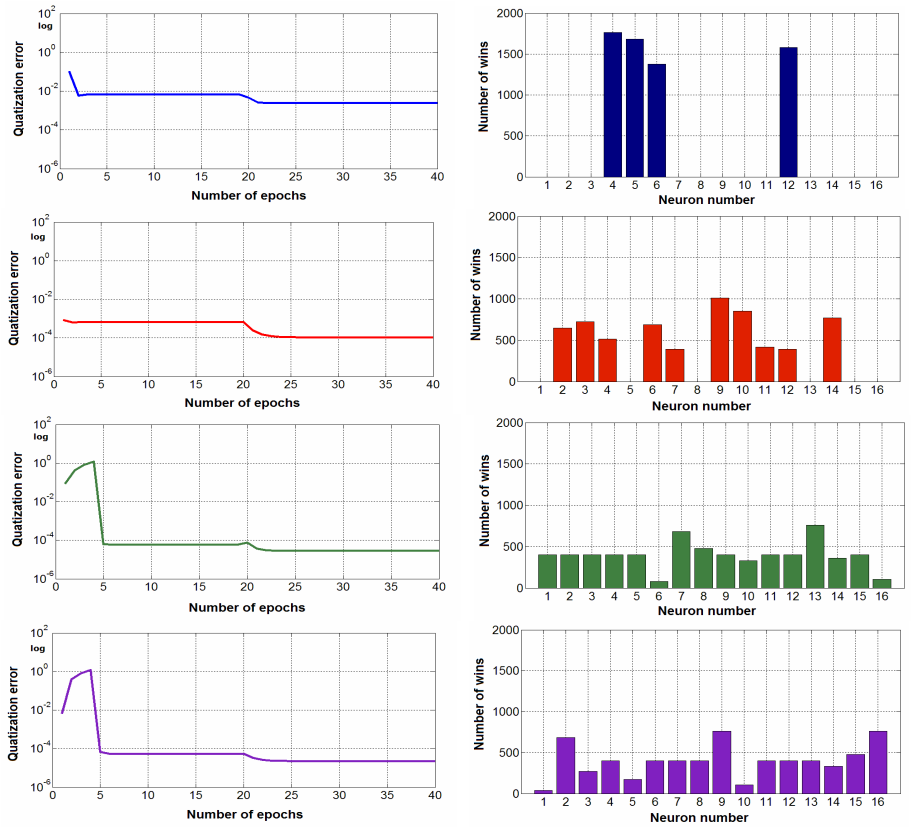


Fig. 3: Quantization error and histogram of number of wins of particular neurons for particular cases presented in Fig.2 for network model simulations.

An example initialization process in designed KNN is illustrated in Fig. 4 (a). This Figure shows currents that represent selected neuron weights. Initialization is a quick process that in this case takes 10 μs per a single weight. The network operation in case when CONS mechanism is not working and without initialization is shown in Fig. 4 (b). Only third neuron is active in this case. Fig. 5 illustrates behavior of the network in case when CONS mechanism is used and when the initialization has been applied and when this initial phase has been omitted. A proper initialization activated all neurons, while in case without initialization first neuron remained inactive, while activity of the second neuron was small. Figs 4(b) and 5 illustrate activity of the WTA block, which directly indicates, which neuron has won in particular time instant.

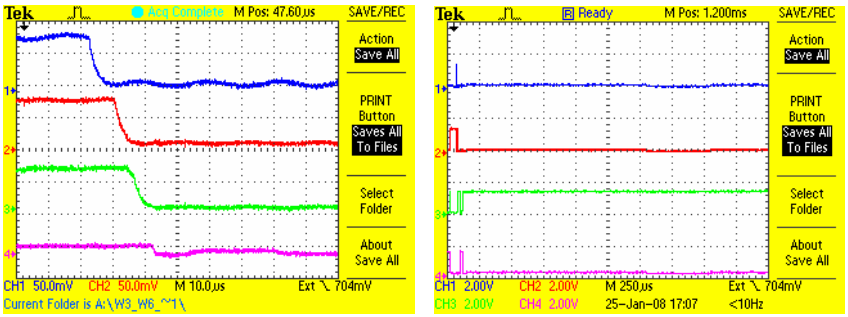


Fig. 4: Measurement results: (a) Initialization of neuron weights (b) operation of the network without initialization and without conscience mechanism

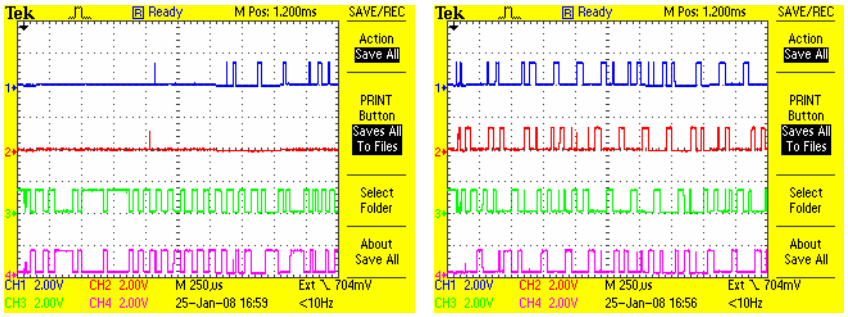


Fig. 5: Measurement results: Operation of the network with conscience mechanism and (a) without and (b) with initial polarization of neurons weights

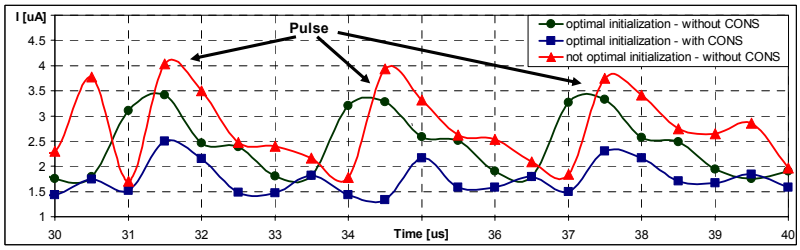


Fig. 6: Quantization error in hardware KNN for different initialization parameters.

Influence of initial values of neurons' weights on quantization error in designed KNN has been confirmed in postlayout simulations for different training vectors X and for different initial values of neurons' weights, as shown in Fig. 6 (after training). The network was trained using example ECG signals, which have been delivered to the network with frequency increased by 500 times to verify also dynamic parameters of designed network. Obtained results are convergent with earlier model simulations, although since number of neurons is smaller in this case, therefore quantization error varies by a smaller factor i.e. by about 50 - 100 %. Conscience mechanism decreases the error by further 20 - 60% (see "pulses" in Fig. 6), increasing number of active neurons. In most experiments the conscience mechanism activated all neurons in the network, while without this mechanism one to three neurons remained dead.

4 Conclusions

This paper presents both the software and the hardware realization of the initialization mechanism for experimental Kohonen neural network implemented in CMOS 0.18 μm technology, which contains 3 inputs and 4 outputs i.e. 12 neuron's weights.

The simulation and measurement results show that proper initial polarization of the network can significantly improve the training process, minimizing number of dead neurons. In the software model of the network with 16 neurons the quantization error for different initial values can vary even by 2 orders of magnitude, while in the hardware implemented network with 4 neurons by 50-250%.

Various initialization methods have been verified using the software model. More complex methods e.g. the hybrid method are difficult to implement in hardware and therefore an experimental conscience mechanism has been used. This mechanism additionally improves the learning process. In comparison to the best case of the initialization without the conscience mechanism, this mechanism decreases the quantization error even 8 times in the software model and by 60% in our experimental hardware network.

References

[1] D. Nguyen, B. Widrow, Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. *International Joint Conference on Neural Networks* (IJCNN 1990), San Diego, USA, 1990

[2] K. Kenni, K. Nakayama, H. Shimodaira, Estimation of initial weight and hidden units for fast learning of multi-layer neural network for pattern classification. *International Joint Conference on Neural Networks* (IJCNN 1990), Washington, USA,1999

[3] Y.K. Kim, J.B. Ra, Weight Value Initialization for Improving Training Speed in the Backpropagation Network. *Intern. Joint Conference on Neural Networks* (IJCNN 1991), Seattle, USA,1991

[4] G. Thimm, E.Fiesler, Neural network initialization In "From neural to artificial neural computation", Editors: J. Mira, F. Sandoval *International Workshop on Artificial Neural Networks*, Malaga, 1995

[5] Y. Chen, F. Bastani, ANN with two-dendrite neurons and its weight initialization, *International Joint Conference on Neural Networks* (IJCNN 1992), Baltimore, USA, 1992

[6] T. Kohonen, *Self-Organizing Maps*, Springer Verlag, Berlin, 2001

[7] H. Nielsen, Counterpropagation networks, *Applied Optics*, Vol. 26, 1987

[8] S.C. Ahalt, A.K. Krishnamurthy, P. Chen, D.E. Melton, Competitive learning algorithms for vector quantization. *Neural Networks*, Vol. 3, 1990

[9] D. E. Rumelhart, D. Zipser, Feature discovery by competitive learning. *Cognitive Sciences*, Vol. 9, 1985

[10] D. DeSieno, Adding a conscience to competitive learning. *International Conference on Neural Network*, San Diego, USA, 1988

[11] T. Talaśka, R. Wojtyna, R. Długosz, K. Iniewski, W. Pedrycz, Analog-Counter-Based conscience mechanism in Kohonen's neural network implemented in CMOS 0.18 μm technology. *The IEEE Workshop on Signal Processing Systems* (SIPS), Banff, Canada, 2006.

[12] T. Talaśka, R. Długosz, W. Pedrycz, Adaptive Weight Change Mechanism for Kohonen's Neural Network Implemented in CMOS 0.18μm Technology, *European Symposium on Artificial Neural networks* (ESANN 2006), Bruge, Belgium, 2007