Homeostasis and artificial neuron

MARTIN RUZEK, TOMAS BRANDEJSKY
Department of informatics and telecommunications
Czech Technical University in Prague
Konviktska 20, Praha 1, 100 00
CZECH REPUBLIC
ruzekmar@fd.cvut.cz, brandejsky@fd.cvut.cz

Abstract: Homeostasis is a property of a system that regulates its internal environment in order to maintain stable condition. This property is typical for biological systems and therefore also for neural cell. This article presents one possible use of the idea of homeostasis in the field of the artificial neural networks. The proposed neuron is a homeostat for which the state of equilibrium means a situation when the level of acceptance of its output reaches its maximum. The neuron is operating with two kinds of information: its input signal (as any artificial neuron), and the input weights of other neurons that are receiving its output. This idea is inspired by the fact that the biological neuron can know which part of its output energy is accepted by other neurons. Several methods of the learning are presented. The main feature of the proposed neuron is the independence of the learning mode; no teacher or higher structure are needed as for example in back-propagation algorithm. Several qualities of the homeostatic neuron, such as stability, speed of learning and independence, are discussed. The results of the first test are presented.

Key-Words: artificial neuron, homeostasis, learning, artificial neural network

1 Introduction

The neural networks are already a classical approach in computer science. Its main paradigm, the artificial neuron, is an idealization of biological neuron. The neural cells is known for more than 100 years, but there are still unsolved questions regarding the mechanism in which the neurons and the whole networks work, even though its applications are used in many original ways [1]. The exact method of learning of the neural networks is still one of the 'mysteries of the nature'. During the 20th century, several artificial neurons have been proposed, some of them proved to be suitable for practical tasks. The most commonly used model is MsCulloch-Pitts neuron. The main advantage of this model is its simplicity. On the other hand, it is clear that the real world neuron is much more complicated.

As there are several types of artificial neurons, there are also some ways of how to organize the neurons into a network and how to learn them. The learning methods can be divided into two types-supervised learning and unsupervised learning. The supervised learning requires a 'teacher', which is in fact a function that informs the neuron about the correctness of its setting. It seems that in nature only the unsupervised learning can exist, however, the reality is more complicated. In the real world, we can expect existence of some 'teacher', as there must be always at least this closed loop: sensors-neural network-effectors-real world-sensors. On the other hand, we can expect some level of independence of each particular neuron. In fact the neural cell is a very complex structure, whose internal complexity is even comparable to the complexity of the full human brain. Therefore we suppose that the neural cell can perform quite a complicated operation.

The design of this model was motivated by intents to simulate brain functions by neural networks. This work is a part of a more general project that is focused mostly on modeling the brain functions. Several artificial models of brain or its parts have already been realized [2]. Models of driver's behavior are important for the identification of dangerous states, such as micro sleep of the loss of the attention; however, the possible uses are wider. It can be used for example for the noise control as suggested in [3]. Evolutionary computing has proven to be a strong method in connection with neural networks [4, 5].

In order to build a neural network it is necessary to define its basic unit, an artificial neuron. Two basic requirements were respected for the construction of this neuron: the similarity to biological neuron (at least in its basic parameters) and the simplicity. The exact copy of biological neuron isn't achievable and is also not desirable, because we won't be able to analyze its functions

[6]. The biological neural network can do the same types of task as the artificial networks: prediction and classification [7, 8]. Another common characteristic of both artificial and biological neural networks is the ability to generalization and abstraction [9]. The possibility of solving never seen problems brings new opportunities to both information theory and everyday life problems [10, 11, 12].

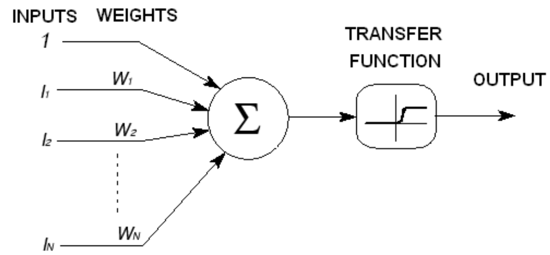
The basic idea is that the neuron is able to change its weights and by this change to increase its importance in the whole network. In further text, the term *input weight* w_i will denote the weight of the connection that is leading the signal into the reference neuron, and the term *output weight* w_o will denote the weight of the connection that transfers the output of the reference neuron to others neurons. *Utility* is real number that quantifies the importance of the reference neuron to other neurons. The utility is calculated as a function of output weights, meanwhile the output weights depend on the the neuron, and therefore on the input weights.

The biological inspiration is obvious, because the proposed neuron is in fact an *information* homeostat. There is no reason why the idea of homeostasis should be limited to energy or physical qualities, such as salinity, humidity or temperature. On the other hand, we can expect that the principle of homeostasis is common to all living entities.

The basic idea is that the neuron is willing to be useful to other neurons. If the other neurons are interested in its output, it will send them majority of its output energy; otherwise the output energy will return to the reference neuron. We can imagine that the neuron is programmed so that it tries to maximize the part of its output energy that is accepted by other neurons. If it is not able to do so because of some reason, it will produce less output or output energy, or it will die completely. This idea is in accordance with our knowledge about the neural cells, as the majority of the neurons die out quite soon after the birth (soon with respect to the whole life of the organism) and only some small part of the cells remains active during the whole life of the organism.

2 Methodology

The proposed neuron is based on McCulloch-Pitts model of artificial neuron that is illustrated on picture 1.



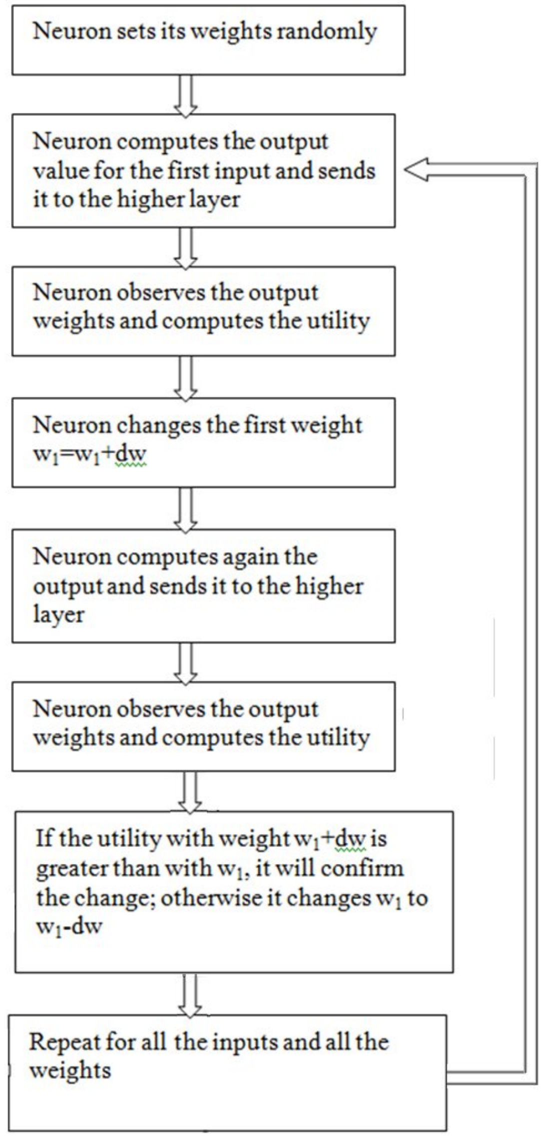
Picture 1: McCulloch-Pitts model of neuron

Mathematically, its function is described as $f(x)= (\sum_{i=1}^{n} x_i w_i + \delta)$. In this paper, a new improvement of this model is presented. The proposed neuron is able to improve automatically its function in a manner that is similar to biological neuron. The similarity to biology was one of the basic requirements; therefore the back propagation algorithm [13] can't be used. The reason is that in this model the backward connections are identical to forward connections. There are no special communication channels for the back propagation of the error. The neuron doesn't calculate the difference between desired and real value, it only observes the weights of the output connections. The basic presumption is that the neuron seeks to be as useful as possible, which means that it intends to make the other neurons to set their input weights to the greatest possible value. The weights are limited to interval $\langle -1;1 \rangle$ because of practical reasons. With the respect to the negative values, it isn't possible to determine the importance of the connection directly by its weight. For instance, a connection with weight –0.8 is more important than weight 0.3. Therefore, the absolute values or the squares of the weights are used instead of the the values directly.

The process of the learning of the neuron can be described by the following algorithm:

1. neuron sets its input weights to random values
2. neuron performs the forward phase (the computation of the output values)
3. neuron evaluates the output weights
4. neuron adds *dw* to the weight of the first connection
5. neuron repeats the forward phase; computes the output of the same input with the changed weight
6. if the previous change made the output weights greater, the neuron will confirm the change. In the contrary, it will take off *dw* from the first connection
7. neuron repeats steps 3 to 6 with all connections and all inputs

This algorithm can be used for neural networks with one layer. In networks with more layers [14], there will be different delays of the signal with the information about the utility. These delays will cause instabilities that will make impossible the direct use of this method because in these multilayered systems the change of the input doesn't influence the output immediately. For neural networks with more layers the algorithm must be modified. This work will deal only the learning of a neuron as a part of one layered neural network. However, some solutions for multilayered networks are suggested.



Diag. 1: Process of learning of the homeostatic neuron

The more detailed algorithm is shown on diag. 1, from which is obvious the relative simplicity of the code. To calculate the importance of the neuron, several methods based on different theories can be used. The first extreme case is searching such setting of the input weights that maximizes the sum of absolute values of the output weights. In other words, the neuron is trying to be interesting to all of the neurons in the higher layer. The other extreme case is neuron that is trying to be interesting for only one neuron in the higher layer. This neuron is searching a setting for which the maximum of the output weights is maximal. Between these extremes there are many compromise variants. For example, the neuron may try to increase its importance to some given number of output neurons. During the process of learning, the neuron can set the weights, the slopes and the thresholds. The process of setting of the slopes and the thresholds is analogical to the weights. In this article, only the weights adjustment will be discussed.

2.1 Learning of the neuron by the sum of the output weights

This method is finding such vector of input weights $A=\{w_1, w_2, .. w_n\}$ for which the sum of the absolute values of the output weights is maximal. This model corresponds well to the biological neuron, because there is only one axon leaving the biological neuron and therefore the neuron can only know the total amount of the signal that is accepted by the others, not the particular weights. In the case of artificial neuron, we expect also the negative weights. Because of that, the neuron will sum the absolute values. The utility *q* is:

$$q = \sum_{j=1}^{n} \left| w_j^o \right| \qquad (1)$$

Another option is to use the square values:

$$q = \sum_{j=1}^{n} \left(w_j^o \right)^2 \qquad (2)$$

Method according to (2) will lead to different values than (1), as it prefers the changes of high absolute values. The neuron based on (2) won't be interested in improving the output weights that are close to 0, as the improvement of these weights will have smaller impact on its overall utility. For example, the combination of the output weights [0.5, 0.6] is better than [0.1] when considered according to (1), otherwise the second combination is better.

The neuron must be equipped with a memory and a function that enables the computation of its utility. The memory makes possible the comparison of the current utility with at least one of the past values. The neuron doesn't know the number of the outputs and therefore doesn't know the maximum of the utility *q*. Therefore it will never be aware of reaching the optimal homeostatic position. The process of the learning will last for the whole time of the existence of the neuron.

2.2 Learning of the neuron by the maximum of the output weights

Another possible type of training is based on the idea that the neuron is willing to increase its importance to only one neuron in the output layer. In this case the optimal setting maximizes function

$$q = \max \left| w_j^o \right|;\ j \in \{0,1,..n\} \qquad (3)$$

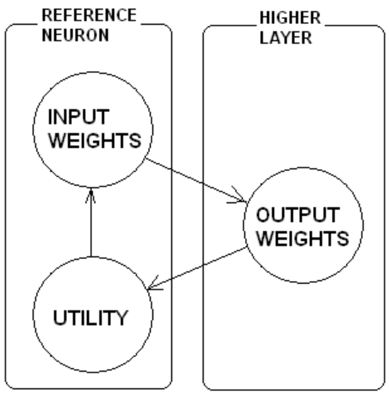
In this situation, the neuron can see whether its setting is ideal or not by comparing its utility q to maximum value, that is 1. When max $|w_o|=1$, no further improvement is possible. The problem is that in situation with many output neurons, the probability that at least one of them is close to 1 is high. Then there will be no learning since the initialization of the process because the neuron will be close to its optimal position. This problem can be solved by adding another criterion that will take into account more neurons.

There are many other 'compromise' solutions that use advantages of both methods. One example is a neuron that is trying to maximize its importance to some given number of neurons in the output layer. This variant seems to be quite promising for future networks because it is most realistic. In the real-world network on can expect that the neurons are trying to be important for the others, but also they are not able to be important for all of them. Therefore this compromise solution (sum of some number of highest absolute values) seems as a model.

3 Results

The neuron was programmed in MATLAB R2008b. The function of learning is based on methods described in Part 2. The input is a vector of any length composed by ones and zeros. The output is a real number from (0, 1) interval. This number determines the values of the output weights, which are the 'second input' of the neuron. The external parameter sets which function will be used for evaluation of the quality of the setting (sum of absolute values, square, maximum or other).

An important step in the design phase of this model was the definition of the output layer. This layer is necessary because the previously described neuron can be tested and improved only as a part of functional virtual environment. The model of the output layer simulates the vector of the connections between the reference neuron and the higher layer. The realization of this layer was the major difficulty of the whole model. The neuron's functions are well defined and therefore its code can exactly fulfill its functions, but the output layer has many dubiousness and ambiguities.



Picture 2: Scheme of the neuron and its environment

Picture 2 illustrates the realization of the test loop, where generation of the inputs is not included. The neuron is receiving input weights and is calculating the output. The higher (output) layer is receiving the output of the neuron and is changing its input weights according to how satisfied is with the work of the neuron. Here we can see one obvious difference with the biology, as one can suppose that the changes of the input weights will take a longer time than a single step. Also, the whole model should be run in continuous time because the biological neural networks do not work in discrete time. However, these tasks are rather out of scope of this research, whose main target is just to develop a working model of homeostatic neuron that can be later adapted to more realistic tasks.

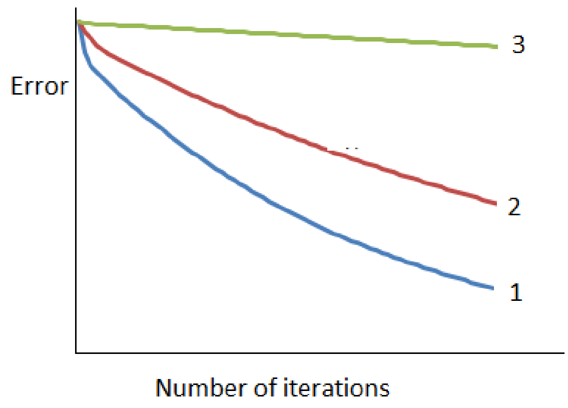
The main difficulty is that in the output layer there may be many different neurons with diverse functions. There is no practical limitation of the number neither of the output neurons nor of their functions. Therefore it turned out that the design of the functional environment is the main obstruction in the training mode. However, it is indispensable to program this layer, as it is not possible to check the feasibility of the homeostatic neuron.

During the first phase of the tests, simple and homogenous layers were considered. Neurons of the output layer were more or less identical. The great majority of the neurons was interested in only one input of the reference neuron. In this case, the reference neuron tends to set one weight (the desired connection) to 1 and all the others to 0. This means that there is one dendrite with useful signal and all the other dendrites transmit useless noise (from the point of view of the higher layer). The speed of convergence for different number of dendrites shows Table 1. In this experiment, it was assumed that the number of the inputs is equal to the number of the output units. Especially in the experiments with few dendrites the random character of the initial setting has a great influence on the number of iterations. To reduce the importance of the random initial setting, the final result was calculated as an average of ten experiments with the same conditions.

Number of dendrites	3	4	5	6
Number of iterations	137	274	698	2825
Number of dendrites	7	8	9	10
Number of iterations	4891	2818	3714	4528

Table 1: Number of necessary iteration as function of number of the inputs to the neuron. The neuron was trained to a simple function of transmitting one input

In the next step, more complex functions were desired by the output layer. First, the output neurons were divided into two groups, each of which was interested in another input signal of the reference neuron. In this case, the convergence process was significantly slower. In this test, the neuron is not considered to be trained immediately after reaching the desired level of importance, but must be able to fulfill the desired function for some period of time. In the following phase, the neuron was tested in an environment with diverse and complex desired functions. The output layer was interested in a group of functions that can't be realized simultaneously at the same time. In this case, the convergence was very slow and sometimes the neuron didn't read the homeostatic position at all.



Picture 3: Dependence of the speed of convergence on the complexity of the output layer. Curve 1 is for homogenous output layer, 2 for output layer with 2 types of neurons and 3 for complex output layer.

The question of the speed of the convergence is illustrated on picture 3. The vertical axis shows the inverse value of the utility *q*. The utility is measured by the weights of the dendrites of the output layer. Therefore, the greater the utility is, the better is the function of the neuron. In the more complicated cases with complex output layer we do not know what is the desired function of the neuron. Therefore, the name of the axis 'error' is a bit misleading in this case, as if there is no exactly defined function, it is difficult to define the error. The level of acceptance or utility is always a relative number fixed to initial initial value. The graph is an idealization obtained from real data. According to first-look observation, the error decreases as a negative exponential function

$$e = a \cdot e^{-bn} + c \qquad (4)$$

where n is a number of the iteration and *a, b* are real constant that are different in each particular situation.

3.1 Discussion

The main advantage of the proposed neuron is its ability of self-learning in way that can be expected in the biological neuron. The learning is indirect; there is no channel for the back propagation of the error. There is also no external function that describes the desired work of the neuron. Instead of this, the neuron is approving itself in order to increase its importance to other neurons. This fact implies that it can be trained incorrectly. If the other neurons are interested in incorrect data, the neuron will try to provide them. The learning is slower than with backpropagation algorithm.

The basic disadvantage of this type of learning is the delay. The proposed neuron changes its weights and expects that the change will be immediately reflected in the output layer. This presumption will be true only for two layered networks. In the case of multi-layered network it will take several steps until the change appear again in the neuron. One of the possible solutions of this problem is setting the dynamics of the inputs to enough low level, so that the change of the input signal will be significantly slower than the communication between the neurons.

3.2 Network of homeostatic neurons

The final neural network shall be composed only of neurons of this type. In other worlds, the neuron proposed in this paper should be also a part of the higher layer. This means that it should be at same time 'teacher' and 'student'. In case of multilayered networks[15, 16, 17], each layer will be learned from the higher layer. The neurons in the layer that is in the actual step working as a 'teacher' must know which function is desired be the higher layer. The neurons in the higher (teaching) layer will assign greater weights to neurons of the lower layer that are producing inputs that are acceptable for them. Therefore, these neurons in the lower layer will be more interested in improving their function only for some neurons in the higher layer. However, this is true only when the principle of measuring the utility from the maximum or sum some number of maximal values is used (3). This fact will lead to some kind of columned structure: some group of neurons will 'work for' another group in the higher layer. On the other hand, some neurons in the lower layer can have strong connections to two or more groups of neurons in the higher layer. This fact corresponds well to our knowledge of the organization of the neural cells in the brain, where at least the cortex has a columned structure. However, this structure is not completely created during the learning process, but is 'ready' from the birth. This situation cannot be simulated exactly, as we obviously don't know the correct organization of the neural network. The optimization of the neural network that is based on homeostatic neurons is a quite complicated task, as there are too many unknowns (the 'traditional' questions of the neural network as topology and neurons' parameters, here also the learning method (sum of output weights, maximum…). The inspiration from the biology here faces its limitations, as the homeostatic neuron is quite far away from the real organization of the biological neuron.

4 Conclusions

The neuron proposed in this article is able to learn independently without any direct use of the teacher. The principle of independent or autonomous neuron is not identical to unsupervised learning; it simply means that each particular neuron is an independent unit. The possible uses of this principle can be increased when used in connection with another theories or systems [18]. In comparison with the back-propagation algorithm the proposed neuron has a slower learning, however, this is not necessarily a disadvantage as the speed of the convergence is not the only one of the criterions of the neural networks and many other characteristics are evaluated [19]. Except of adapting the neuron for the whole neural network, other improvements such as mode changes [20] will be in focus during the future research. Several ideas [21, 22, 23] can make this network viable in real-world problems. During the process of learning, it is searching its homeostatic position. The neuron is based on McCulloch-Pitts model with some modifications. The homeostatic position of the neuron is a situation when the acceptance of its output is maximal. The neuron is trying to increase its significance by changing its input weights. This process can be adapted also for other parameters of the neuron, such as the slope and the threshold; however, in this article only the weight adjustment is discussed, as it is the most important stage in the process of learning. The utility of the neuron is measured by the weights that other neurons are using to multiply the output of the reference neuron. The main advantage of this method is that the process of learning is autonomous; no external learning function is needed. Therefore it is closer to the original biological inspiration, the neural cell organized in brain structure. The experiments confirmed that this neuron converges to the homeostatic position. The simpler the desired function is, the faster is the convergence. In case of difficult and diverse desired function the neuron doesn't converge. This is one of the limitations of this model as the neural networks are in general useful for solutions of complex tasks. The main disadvantage of this model is that is applicable to systems with first order delay. In the following research, this model will be adapted to networks with two or more layers. This model can be a foundation stone of a new kind of neural network. The network composed of independent homeostatic neurons may be used for simulation of brain functions, as this neuron was inspired by biological neuron and in principle meets the concept of homeostat that is proper to all living cells. The use of fuzzy models in combination with the homeostatic neuron seems to be perspective in this area. However, the possible use of this network may be much wider, as it can be used in many different areas and tasks, such as prediction [24], classification [25] or control. In connection to mental models, unexpected applications arise in the field of the transport [26, 27]. In future research, we will focus on adapting this neuron to be able to work as a part of a whole network [28] as suggested in part 3.2. To do so, it is necessary to do two things. First, the question of signal delay must be solved out. In the case of multilayered network [29, 30] the neuron must work in an environment with bigger difference between the input and the information about the utility. Second improvement is that the neuron must be adapted to work as a part of the higher layer. In other words, it must be able to decide which inputs are relevant for its function and which should be suppressed. Although there are still no practically usable results with real-world data, the idea of homeostatic neuron seems to be able to introduce new possibilities to both the mental model of the driver, as it is its original motivation, as to the investigation of the neural networks in general.

References

[1] Chiang, W. K. and Zhang, D., Zhou, L., Predicting and Explaining Patronage Behavior Toward Web and Traditional Stores Using Neural Networks: a Comparative Analysis with Logistic Regression, *Decision Support Systems*, Vol. 41, 2006, pp. 514-531.

[2] Omari Abdallah, Functional learning & Introduction to the Electronic Brain on El-dorra neural network technology, *Proceeding of the 11th WSEAS Int. Conference on Neural Networks (NN'10)*, Iasi, Rumania, June 13-15, 2010, pp. 37-44

[3] Guarnaccia C., Acoustical Noise Analysis in Road Intersections: a Case Study, *Proceedings of the 11th WSEAS Int. Conf. on Acoustics & Music: Theory & Applications (AMTA '10)*, Iasi, Romania, June 13-15, 2010. ISBN: 978-960-474-192-2/ ISSN: 1790-5095, pp. 208-215.

[4] Mousa AL-Akhras, An Evolutionary-Optimised Artificial Neural Network Approach for Automatic Appraisal of Jordanian Lands and Real

Properties, *Proceeding of the 11th WSEAS Int. Conference on Neural Networks (NN'10)*, Iasi, Rumania, 13-15 June, 2010, pp. 203-208

[5] Steri S., Quartieri J., Volzone G., Guarnaccia C., Evolutionary Processes Solved with Lie Series and by Picard Iteration Approach, *International Journal of Mathematical Models and Methods in Applied Sciences*, Issue 2, Vol. 2, pp 262-268, 2008, ISSN: 1998-0140

[6] M. Novák a kol, *Umělé neuronové sítě, teorie a aplikace*, Praha, C.H. BECK, 1998

[7] Hamid Yazdani, Prediction of Chaotic Time Series using Neural Network*, Proceeding of the 10th WSEAS International Conference on NEURAL NETWORKS (NN'09)*, Prague, Czech republic, March 23-25, 2009, pp. 63-69

[8] Chiang, W. K. and Zhang, D., Zhou, L., Predicting and Explaining Patronage Behavior Toward Web and Traditional Stores Using Neural Networks: a Comparative Analysis with Logistic Regression, *Decision Support Systems*, Vol. 41, 2006, pp. 514-531.

[9] Arif Selcuk Ogrenci, Abstraction in FPGA Implementation of Neural Networks, *Proceeding of the 9th WSEAS International Conference on NEURAL NETWORKS (NN'08)*, Sofia, Bulgaria, May 2-4, 2008, pp 215-221

[10] Nidal F. Shilbayeh, Gaith A. Al-Qudah, Face Detection System Based on MLP Neural Network, *Proceeding of the 11th WSEAS Int. Conference on Neural Networks (NN'10),* Iasi, Rumania, June 13-15, 2010, pp 238-243

[11] S. Steri, J. Quartieri, G. Volzone, C. Guarnaccia, On Picard Iteration and Lie Series in Analytic Evolutionary Problems of Physics, *Proceedings of the 9th WSEAS International Conference on Mathematics & Computers in Biology & Chemistry (MCBC '08)*, Bucharest, Romania, June 24-26, 2008, pp 157-163.

[12] Leonardo Ivirma, Mary Vergara, Sebastian Provenzano, Francklin Rivas, Anna Perez, Francisco Fuenmayor, Stress Smoothing in Hexaedrons using Artificial Neural Networks*, Proceeding of the 10th WSEAS International Conference on NEURAL NETWORKS (NN'09),* Prague, Czech republic, March 23-25, 2009, pp. 41-47

[13] H.S. Wang, Z.H. Che, Y.N. Wang, Cost estimation of plastic injection products through back-propagation network, *Proceeding of the 8th WSEAS International Conference on Neural Networks*, Vancouver, Canada, June 19-21, 2007, pp. 54-60

[14] Saeid Iranmanesh, A Differential Adaptive Learning Rate Method for Back-Propagation Neural Networks*, Proceeding of the 10th WSEAS International Conference on NEURAL NETWORKS (NN'09),* Prague, Czech republic, March 23-25, 2009, pp. 30-35

[15] S. Sureettanan, H. N. Phien, N. Sureettanan, N. Mastorakis, The Optimal Multi-layer Structure of Backpropagation Networks, *Proceedings of the 7th WSEAS International Conference on Neural Networks,* Croatia, June 12-14, 2006, pp 108-113

[16] P. Radonja, S. Stankovic, D. Drazic, B. Matovic, Generalized Models Based on Neural networks and Multiple Linear Regression, *Proceedings of the 5th WSEAS Int. Conf. on CIRCUITS, SYSTEMS, ELECTRONICS, CONTROL & SIGNAL PROCESSING*, Dallas, USA, November 1-3, 2006 279-284

[17] *S. D'Ambrosio, C. Guarnaccia, M. Guida, J. Quartieri, S. Steri; A biomathematical study of a controlled birth and death process describing malignancy*; Proceedings of the 4th WSEAS International Conference on MATHEMATICAL BIOLOGY and ECOLOGY (MABE'08)*, Acapulco, Mexico, January 25-27, 2008*

[18] P. Bonissone, Y.-T. Chen, K. Goebel, P. Khedkar, Hybrid soft computing systems: industrial and commercial applications*, Proceedings of the IEEE 87 (9),* 1999, pp. 1641 – 1667.

[19] Iftikhar Ahmad, M. A. Ansari, Sajjad Mohsin, Performance Comparison between Backpropagation Algorithms Applied to Intrusion Detection in Computer Network Systems. *Proceeding of the 9th WSEAS International Conference on NEURAL NETWORKS (NN'08)*, Sofia, Bulgaria, May 2-4, 2008, pp 231-237

[20] Dominic Palmer-Brown, Miao Kang, Sin Wee Lee, Meta-Adaptation: Neurons that Change their Mode*, Proceeding of the 9th WSEAS International Conference on NEURAL NETWORKS (NN'08),* Sofia, Bulgaria, May 2-4, 2008, pp 155-167

[21] Elad, M.; Hel-Or, Y.; Keshet, R., Pattern detection using a maximal rejection classifier, *Pattern recognition letters*, Vol.23, 2001, pp. 1459-1471.

[22] Novoa, D. Pérez, A. Rivas, F., Fault Detection scheme using Neo-fuzzy Neurons, *IASTED International Conference on Intelligent Systems and Control*, Honolulu, Hawaii, USA, 2000

[23] Obreque C., Donoso M., Gutierrez G., Marianov V., A branch and cut algorithm for the hierarchical network design problem, *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH*, Vol. 200 Issue 1, Jan. 2010, pp. 28-35

[24] Samaria Munoz-Bravo, Anna Perez-Mendez, Francklin Rivas-Echeverria, Model for Estimating Venezuelan Population with Working Age using Artificial Neural Networks*, Proceeding of the 10th*

WSEAS International Conference on NEURAL NETWORKS (NN'09), Prague, Czech republic, March 23-25, 2009, pp. 35-41

[25] Zarita Zainuddin & Ong Pauline, Function Approximation Using Artificial Neural Networks, *INTERNATIONAL JOURNAL OF SYSTEMS APPLICATIONS, ENGINEERING & DEVELOPMENT,* Issue 4, Volume 1, 2007, pp. 173-178

[26] Quartieri J., Mastorakis N. E., Iannone G., Guarnaccia C., Cellular Automata Application to Traffic Noise Control*, Proceedings of the 12th WSEAS Int. Conf. on Automatic Control, Modelling & Simulation (ACMOS '10),* Catania, Italy, May 29-31, 2010. ISBN: 978-954-92600-1-4 / ISSN: 1790-5117, pp. 299-304

[27] Martin, Q. Santana, Y., Aplicación de las Redes Neuronales Artificiales a la Regresión, *Editorial La Muralla*, Spain, 2007

[28] D. E. Rumelhart, J. McClelland, *Parallel Distributed Processing*, MIT Press, 1986

[29] Wei Xia, Danny Ho, L. F. Capretz, A Neuro-Fuzzy Model for Function Point Calibration, *WSEAS TRANSACTIONS on INFORMATION SCIENCE & APPLICATIONS,* Vol. 5, January 2008, pp. 22-30

[30] A. E. Bryson, Yu-Chi Ho, *Applied optimal control, optimization, estimation and control,* Hemisphere Publishing Corporation, 1975