Bio-inspired Artificial Intelligence: A Generalized Net Model of the Regularization Process in MLP

Stanimir Surchev[1], Sotir Sotirov[1], Wojciech Korneta[2]

[1] *"Prof. Dr. Assen Zlatarov" University*
Burgas, Bulgaria
E-mails: ssurchev@gmail.com, ssotirov@btu.bg

[2]*College of Computer Science and Business Administration in Łomża*
14, Akademicka, 18-400 Łomża, Poland
E-mail: wkorneta@pwsip.edu.pl

Received: July 3, 2013 **Accepted: October 04, 2013**

Published: October 15, 2013

Abstract: *Many objects and processes inspired by the nature have been recreated by the scientists. The inspiration to create a Multilayer Neural Network came from human brain as member of the group. It possesses complicated structure and it is difficult to recreate, because of the existence of too many processes that require different solving methods. The aim of the following paper is to describe one of the methods that improve learning process of Artificial Neural Network. The proposed generalized net method presents Regularization process in Multilayer Neural Network. The purpose of verification is to protect the neural network from overfitting. The regularization is commonly used in neural network training process. Many methods of verification are present, the subject of interest is the one known as Regularization. It contains function in order to set weights and biases with smaller values to protect from overfitting.*

Keywords: *Neural network, Generalized net, Supervised learning, Overfitting, Regularization.*

Introduction

Neural Networks are an abstract representation, bio-inspired from human brain neural system [4]. All elements of the artificial neuron are taken from the biological neuron (Fig. 1).

An electrical signal has been received in a biological neuron from other neurons, and conveyed to the cell body by dendrites. Resultant electrical signals are sent along the axon in order to be distributed to other neurons. The artificial neuron operates by analogy. Activation signals from other neurons are summed in the neuron and passed through an activation function and the resulting value is further sent to the other neurons [8].

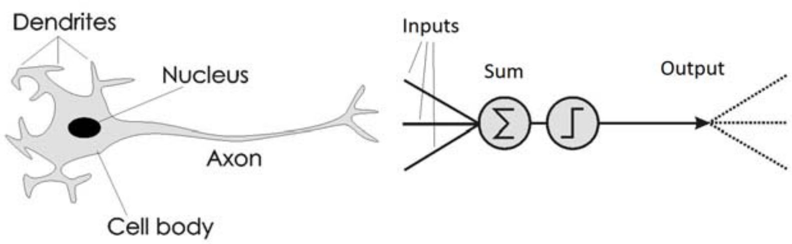

Fig. 1 Comparison between biological and artificial neuron

There are many types of neural networks and each of them represents a different aspect of recognition and prediction. In general, there are two types of learning of the neural networks –

unsupervised and supervised [7]. The problem of overfitting is frequently met in the learning process [3]. It appears in different situations, affects the trained parameters and worsens the output results. There are different methods that can help avoiding the overfitting, namely Early Stopping and Regularization. In [6], we described the process of Early Stopping. The process of Regularization is described below.

The general idea of the Regularization method is to improve the performance function and this is the reason why the described neural network with generalized net exhibits the moment where the training process obtains the coefficients of weights and biases.

The first step is to prepare the necessary data of neural network. The data contains input and target matrix, which must have an equal number of vectors:

$$p_{i,j} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,j} \\ \vdots & \vdots & \vdots \\ p_{i,1} & \cdots & p_{i,j} \end{bmatrix}$$

where i is the maximum number of vectors in input data; and j is the maximum number of elements of input vector; and

$$t_{m,n} = \begin{bmatrix} t_{1,1} & \cdots & t_{1,n} \\ \vdots & \vdots & \vdots \\ t_{m,1} & \cdots & t_{m,n} \end{bmatrix}$$

where m is the maximum number of vectors in target data; and n is the maximum number of elements of target vector.

When the matrices are obtained, the next step is to choose the type of neural network and necessary parameters and after that the training process can begin. The program source code for creation of the neural network has the following requirements:

$$net = \left(\left[\min\left(p_{i,1}\right)\ \max\left(p_{i,1}\right); \ldots ; \min\left(p_{i,j}\right)\ \max\left(p_{i,j}\right)\right], \left[G\ ; h\right], \left\{\ 'F_i\ ', \ 'F_0\ '\right\}, \ 'F_t\ '\right) \quad (1)$$

where net is the set of elements for creating the feedforward neural network; $\min\left(p_{i,j}\right), \max\left(p_{i,j}\right)$ are the minimal and maximum elements value of column; G is the number of hidden neurons; h is the number of outputs neurons; F_i is the performance function of input data; F_0 is the performance function of output data; F_t is the training function. It updates values of weight and bias matrixes, where depending of training method. The following formulae can be changed:

$$w^g\left(k+1\right) = w^g\left(k\right) - \alpha \frac{\partial \hat{F}}{\partial w^g} \quad (2)$$

$$b^g\left(k+1\right) = b^g\left(k\right) - \alpha \frac{\partial \hat{F}}{\partial b^g} \quad (3)$$

where g is the current neuron; k is the current number of iteration; $\alpha \frac{\partial \hat{F}}{\partial w^g}$ is the modification function.

When the input parameters are prepared, the next step is to calculate and train the neural network. The basic parameter that is necessary for training process is the performance function, and the algorithm of mean squares error (MSE) is basically used:

$$mse = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 \quad (4)$$

where a is the output data;

$$a = f\left(\sum_{i=1}^{N}(p_i w_i) + b\right) \quad (5)$$

and t is the target data; N is the maximum number of training pairs (p, t); e is the error between output and target data.

The Regularization method changes the formula of MSE:

$$msereg = \gamma * mse + (1-\gamma) * msw \quad (6)$$

where γ is the performance parameter with $\gamma \in [0 \div 1]$. Its value is very important, because the line between good and overfit output data is very small.

$$msw = \frac{1}{n}\sum_{j=1}^{n} w_j^2 \quad (7)$$

where msw – mean square weights; n – maximum number of weights.

The modification of the performance function reduces the weight and bias values. This influences the results, making them smoother and decreasing the chance of overfitting.

GN-model

Initially, the following tokens enter in the GN [1]:

- In place L_1 – α token with initial characteristic "Input and target vectors"

$$x_0^{\alpha} = \text{“} p_{i,j},\ t_{m,n} \text{”};$$

- In place L_5 – β token with initial characteristic "Structure of neural network"

$$x_0^{\beta} = \text{“net”};$$

- In place L_6 – δ token with initial characteristic "Maximum number of iteration and minimum value of performance function"

$$x_0^{\delta} = \text{“} I_{max}, M_{min} \text{”};$$

- In place $L_7 - \gamma$ token with initial characteristic "Performance parameter"

$$x_0^{\gamma} = \text{“}\gamma\text{”};$$

- In place $L_{19} - \varepsilon$ token with initial characteristic "Initial and maximum corrections"

$$x_0^{\varepsilon} = \text{“}K_s = 0, K_{max}\text{”}.$$

The GN presented on Fig. 2 by the following set of transitions: $A = \{Z_1, Z_2, Z_3, Z_4\}$. These transitions describe the following processes:

Z_1 = "Preparing of data";
Z_2 = "Training of neural network";
Z_3 = "Computing mean square error with regularization";
Z_4 = "Correction, if overfit".

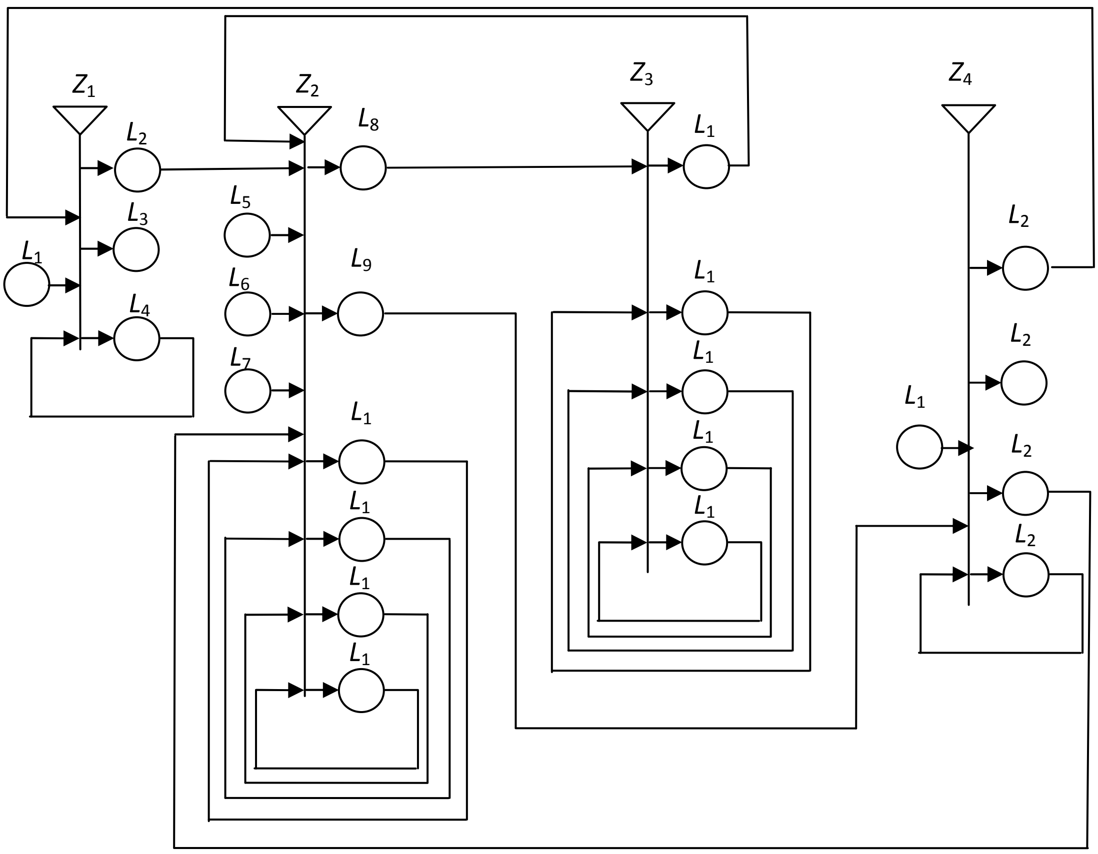

Fig. 2

$$Z_1 = \langle \{L_1, L_4, L_{20}\}, \{L_2, L_3, L_4\}, R_1, \square(L_1, L_4, L_{20})\rangle,$$

$$R_1 = \begin{array}{c|ccc} & L_2 & L_3 & L_4 \\ \hline L_1 & False & False & True \\ L_4 & W_{4,2} & W_{4,3} & True \\ L_{20} & False & False & True \end{array},$$

where $W_{4,2}$ = "$i = m$"; $W_{4,3} = \neg W_{4,2}$.

The token α enters place L_4 from place L_1 and does not obtain any new characteristic. It stays in place L_4 during the whole functioning of the GN. The token α enters place L_2 from place L_4

and does not obtain any new characteristic. The token α enters place L_3 from place L_4 and does not obtain any new characteristic.

$$Z_2 = \langle\{L_2, L_5, L_6, L_7, L_{10}, L_{11}, L_{12}, L_{13}, L_{14}, L_{22}\}, \{L_8, L_9, L_{10}, L_{11}, L_{12}, L_{13}\}, R_2, \square(\square(L_2, L_5, L_7), L_6, L_{10}, L_{11}, L_{12}, L_{13}, L_{14}, L_{22}))\rangle,$$

$$R_2 = \begin{array}{c|cccccc} & L_8 & L_9 & L_{10} & L_{11} & L_{12} & L_{13} \\ \hline L_2 & False & False & False & False & False & True \\ L_5 & False & False & False & False & False & True \\ L_6 & False & False & True & True & False & False \\ L_7 & False & False & False & False & False & False \\ L_{10} & False & W_{10,9} & True & False & W_{10,12} & False \\ L_{11} & False & W_{11,9} & W_{11,10} & True & False & False \\ L_{12} & W_{12,8} & False & False & False & True & False \\ L_{13} & False & False & False & False & W_{13,12} & True \\ L_{14} & False & False & False & True & False & False \\ L_{22} & False & False & True & True & False & True \end{array},$$

where

$W_{10,9}$ = "$I_{cu} = I_{max}$"; $W_{10,12} = \neg W_{10,9}$;
$W_{11,9}$ = "$F_{msereg} \leq M_{min}$"; $W_{11,10} = \neg W_{11,9}$;
$W_{12,8}$ = "Computed weights and biases";
$W_{13,12}$ = "Necessary data for training are obtained".

The tokens α, β, γ from places L_2, L_5, L_7 unite in one token ζ in place L_{13} and this new token obtains the new characteristic "Necessary data for training of neural network". The token ζ splits in two equal tokens, accordingly: the first one enters place L_{12} and obtains a new characteristic "Weight and bias data", while the second one stays in place L_{13} for the whole functioning of the GN.

$$x_{cu}^{\zeta} = \text{“}\, x_0^{\alpha}, x_0^{\beta}, x_0^{\gamma}, W, b\,\text{”}$$

The token δ from place L_6 splits in two tokens δ' and δ'', in places L_{10} and L_{11} accordingly, and these obtain the respective new characteristics: "Maximum number of iterations" and "Minimum number of performance function":

$$x_{cu}^{\delta'} = \text{“}\, I_{max}\,\text{”};$$
$$x_{cu}^{\delta''} = \text{“}\, M_{min}\,\text{”}.$$

The token ζ from place L_{12} splits in two tokens ζ' and ζ'', where the first token enters place L_8, and the second one enters place L_{12}, and they accordingly obtain the following new characteristics "Data for computing of performance function" and "Necessary data for training":

$$x_{cu}^{\zeta'} = \text{“}\, p, t, F_t, W, b, x_0^{\gamma}\,\text{”};$$
$$x_{cu}^{\zeta''} = \text{“}\left[\min\left(p_{i,1}\right) \max\left(p_{i,1}\right); \ldots; \min\left(p_{i,j}\right) \max\left(p_{i,j}\right)\right], \left[G; h\right], \{\, 'F_i', 'F_0' \}, W, b\,\text{”}.$$

The token η from place L_{11} enters place L_9, where it does not obtain any new characteristic. The token η from place L_{11} enters place L_{10}, where it obtains the new characteristic "Current number of iterations":

$$x_{cu}^{\eta} = \text{"} F_{msereg}, I_{cu} \text{"};$$
$$I_{cu+1} = I_{cu} + 1.$$

The token η from place L_{10} enters place L_9, and does not obtain any new characteristic. The token η from place L_{10} enters place L_{12}, and again does not obtain any new characteristic.

$$Z_3 = \langle \{L_8, L_{15}, L_{16}, L_{17}, L_{18}\}, \{L_{14}, L_{15}, L_{16}, L_{17}, L_{18}\}, R_3, \square(\square(L_{16}, L_{17}), L_8, L_{15}, L_{18})\rangle,$$

$$R_3 = \begin{array}{c|ccccc} & L_{14} & L_{15} & L_{16} & L_{17} & L_{18} \\ \hline L_8 & False & False & True & False & True \\ L_{15} & W_{15,14} & True & False & False & False \\ L_{16} & False & W_{16,15} & True & False & False \\ L_{17} & False & W_{17,15} & False & True & False \\ L_{18} & False & False & False & W_{18,17} & True \end{array},$$

where
$W_{15,14}$ = "Computed mean square error with regularization (*msereg*)";
$W_{16,15}$ = "Computed mean square weights (*msw*)";
$W_{17,15}$ = "Computed mean square error (*mse*)";
$W_{18,17}$ = "Computed output data (*a*)".

The token ζ from place L_8 splits in two tokens: ζ_1' and ζ_2' that enter places L_{16} and L_{18}. They obtain there the following new characteristics, accordingly; "Mean square weights" and "Output data":

$$x_{cu}^{\zeta_1'} = \text{"} x_0^{\gamma}, W, b, F_{msw} \text{"};$$
$$x_{cu}^{\zeta_2'} = \text{"} t, p, W, b, F_t, a \text{"}.$$

The token ζ_2' from place L_{18} splits in two tokens: the first token ζ_{21}' enters place L_{17} and obtain the new characteristic "Mean square error", while the second token ζ_{22}' stays in place L_{18} for the whole functioning of the GN:

$$x_{cu}^{\zeta_{21}'} = \text{"} t, a, F_{mse} \text{"};$$
$$x_{cu}^{\zeta_{22}'} = \text{"} p, W, b, F_t \text{"}.$$

The token ζ_1' from place L_{16} splits in two tokens: the first token ζ_{11}' enters place L_{15}, and the second token ζ_{12}' remains looping in place L_{16} for the whole functioning of the GN:

$$x_{cu}^{\zeta_{11}'} = \text{"} x_0^{\gamma}, F_{msw} \text{"};$$
$$x_{cu}^{\zeta_{12}'} = \text{"} W, b \text{"}.$$

The token ζ_{21}' from place L_{17} splits in two: the first token ζ_{211}' enters place L_{15}, while the second token ζ_{212}' stays in place L_{17} for the whole functioning of the GN:

$$x_{cu}^{\zeta'_{211}} = \text{“}F_{mse}\text{”};$$
$$x_{cu}^{\zeta'_{212}} = \text{“}t, a\text{”}.$$

The tokens ζ'_{11} and ζ'_{211} from places L_{16} and L_{17} unite in one token θ in place L_{15} and that new token obtains the characteristic “Mean square error with regularization”. The token θ splits in two: the first token η enters place L_{14}, while the second token θ remains to loop in place L_{15} for the whole functioning of the GN:

$$x_{cu}^{\theta} = \text{“}F_{mse}, x_0^{\gamma}, F_{msw}\text{”};$$
$$x_{cu}^{\eta} = \text{“}F_{msereg}\text{”}.$$

The token θ from place L_{15} enters place L_{14} where it does not obtain any new characteristic.

$$Z_4 = \langle \{L_9, L_{19}, L_{23}\}, \{L_{20}, L_{21}, L_{22}, L_{23}\}, R_4, \square(L_9, L_{19}, L_{23})\rangle,$$

$$R_4 = \begin{array}{c|cccc} & L_{20} & L_{21} & L_{22} & L_{23} \\ \hline L_9 & False & False & False & True \\ L_{19} & False & False & False & True \\ L_{23} & W_{23,20} & W_{23,21} & W_{23,22} & True \end{array},$$

where $W_{23,21}$ = “Network is not overfit”; $W_{23,20} = W_{23,22} = \neg W_{23.21}$.

The token η from place L_9 enters place L_{23} and obtains there a new characteristic “Current number corrections”:

$$K_{cu+1} = K_{cu} + 1.$$

The token η obtains the following characteristics:

- When transferring from place L_{23} to place L_{20}, it obtains the new characteristic “Correction of input and target data”;
- When transferring from place L_{23} to place L_{21}, it obtains the new characteristic “Correct trained neural network”;
- When transferring from place L_{23} to place L_{22}, it obtains the new characteristic “Correction of necessary training data”.

Conclusion

In the present paper, one of the methods of verification of neural networks, namely Regularization, has been described by the apparatus of generalized nets. The main function of this method is to decrease weight and bias values by modification of the performance function. This modification reduces the percentage of overfitting.

References

1. Atanassov K. (1991). Generalized nets, World Scientific, Singapore, New Jersey, London.
2. Hudson B. M., M. T. Hagan, H. B. Demuth (2012). Neural Network Toolbox User’s Guide R2012a 1992-2012.
3. Geman S., E. Bienenstock, R. Doursat (1992). Neural Networks and the Bias/Variance Dilemma, Neural Computation, 4, 1-58.
4. McCulloch W., W. Pitts (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity, Bulletin of Mathematical Biophysics, 5, 115-133.

5. Morgan N., H. Bourlard (1990). Generalization and Parameter Estimation in Feedforward Nets, 630-637.
6. Tikhonov A. N. (1963). On Solving Incorrectly Posed Problems and Method of Regularization, Doklady Akademii Nauk USSR, 151, 501-504.
7. Widrow B., M. Hoff (1960). Adaptive Switching Circuits, IRE WESCON Convention Record, 96-104.
8. www.webpages.ttu.edu/dleverin/neural_network/neural_networks.html (access date July 3, 2013).

Stanimir Surchev, M.Sc.
E-mail: ssurchev@gmail.com

Stanimir Surchev graduated from "Prof. Dr. Assen Zlatarov" University, Burgas, Department of Computers, Systems and Technologies in 2012. His research is in the field of chaotic signals using neural network. His current research interests are on neural network, chaos and computers.

Assoc. Prof. Sotir Sotirov, Ph.D.
E-mail: ssotirov@btu.bg

Sotir Sotirov was graduated from Technical University in Sofia, Bulgaria, Department of Electronics. He received his Ph.D. degree from "Prof. Dr. Assen Zlatarov" University, Burgas in 2005. His current research interests are in the field of modeling of neural networks.

Wojciech Korneta, Ph.D., Dr. Hab.
E-mail: wkorneta@pwsip.edu.pl

Wojciech Korneta was graduated from Technical University in Warsaw, Poland, Department of Technical Physics and Applied Mathematics. He received his Ph.D. degree in Łódź University, Poland in 1984 and Dr. Hab. degree in 2002. His current research interests are in the field of signal processing, stochastic resonance and chaos.