Transactions of the VŠB – Technical University of Ostrava, Mechanical Series
No. 2, 2008, vol. LIV,
article No. 1636

Jolana ŠKUTOVÁ*

WEIGHTS INITIALIZATION METHODS FOR MLP NEURAL NETWORKS

METODY INICIALIZACE VAH PRO NEURONOVÉ SÍTĚ TYPU MLP

Abstract

This paper describes the weights initialization methods for two-layer feedforward neural networks. The choice of method influences of a convergence and a minimal final value of cost function for a neural network training process. The neural networks application in different domain to be expected, that their users will be acquired the applicable neural network models. Hence it is important prune away all kinds of an uncertainties while the choice of neural network structure, the learning algorithm in context of other adjustable parameters as well as preparing of suitable learning and testing data set for the neural network learning.

Abstrakt

Příspěvek popisuje inicializační metody vah spojení pro dvouvrstvé dopředné neuronové sítě, jejichž volba má vliv na rychlost a minimální výslednou hodnotu účelové funkce při vlastním učení neuronových sítí. Aplikace neuronových sítí v různých oborech předpokládá, že jejich uživatelé získají takové modely neuronových sítí, které budou použitelné. Proto je důležité odstranit všechny možné nejasnosti při volbě typu a struktury neuronové sítě, metodě učení v souvislosti s dalšími stavitelnými parametry i sestavení vhodných vzorů pro učení a testování neuronové sítě.

1 INTRODUCTION

The using of neural networks for different applications requires the acquirement such a quality model of neural network that cost function minimization heads for a global minimum. In case the learning algorithm attains to local minimum of cost function, the neural network model can be satisfactory model. In the other situation with different initial parameters is possible to acquire another local minimum, but this neural network model can be unacceptable and its application will be unavailable. The initial values of neural network weights influence searching a global minimum resulting in the acquirement of quality neural network model. If the neural network model is unavailable, the learning process is executes again, in some case at more unsuccessful attempt at the acquirement of the quality neural network model.

For a large training set, the time for finding of acceptable model should be reduced. The reduction in a repetition rate of learning algorithm as well as the reduction in a convergence speed of learning algorithm requires the minimal cost function in a short time. For a setting of initial parameters, the choice of initialization method can influence these requirements.

2 INITIALIZATION METHODS OF MLP NEURAL NETWORKS

In current time, the initialization methods attract an attention in a neural networks research in comparison with a learning algorithm development. This is concern primarily multi-layer perceptron

* Ing., Department of Control Systems and Instrumentation, Faculty of Mechanical Engineering, VSB – Technical University of Ostrava, 17. listopadu 15/2172, Ostrava-Poruba, tel. (+420) 59 732 4192, e-mail jolana.skutova@vsb.cz

(MLP) neural networks used quite often for an optimization and a control system considering other type of neural network, where their time point of view is a primary requirement. In most publications, there are often mentioned a uniform random initialization of weight centered about zero and Nguyen-Widrow initialization extended to an application programs. There are a few of method using deterministic values for the weight initialization, e.g. [4], [7]. In several recent years, a number of innovative initialize methods are developed and tested, see publications [9], [8] and genetic algorithms also are applied to parameters initialization [6].

2.1 Uniform random initialization method

This method is employed as the primary procedure for neural network weight initialization that randomly setting a initial values of weights in the selected range, generally in range of [-1, 1], but often in range of [-0.5, 0.5], see the publication [7].

2.2 Nguyen-Widrow initialization method

This method is very used for the weight initialization of the feedforward neural networks. The initial weights of the hidden neurons are distributed so that a learning of neural network progresses more effectively. The factor β is given by the equation for the feedforward neural network with n inputs and p hidden neurons

$$\beta = 0.7\sqrt[n]{p}, \tag{1}$$

and then the weights $w_{ij}^{(0)}$ are randomly chosen in the range of [-1, 1]. Next, the initial weight values are expressed using normalized vector and factor β as

$$w_{ij} = \frac{\beta \cdot w_{ij}^{(0)}}{\left\| w_j^{(0)} \right\|}, \tag{2}$$

where w_{ij} are initial parameters of learning algorithm. For the output layers, the initial weights are also random generated, but in range of [-0.5, 0.5].

2.3 Initial method based on sensitivity analysis

The method is based on using of a sensitivity of all parameter of each layer respect to their inputs and outputs and also on using the independent system of linear equations of each layer and the objective of method is optimised the weight values. This method can be used as the method for neural network learning, but also as the initial method for faster and more effective Levenberg/Marquardt learning method. For the application, the method expects exactly two-layered feedforward neural network.

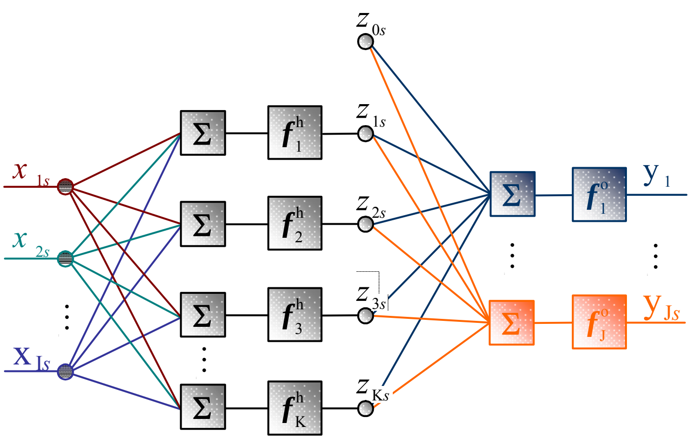

Fig. 1 Two-layer perceptron neural network

The two-layered neural network (Fig. 1) is split in two one-layered networks where first network have inputs, x_{is}, and outputs, z_{ks}. Further, I is the number of inputs, K is the number of outputs, and S is the number of training data. The second network has K+1 inputs, z_{ks}, and J outputs, y_{js}, and f_k^h is a transfer function of hidden layer and f_j^o is a transfer function of output layer of neural network [1]. On the assumption that output of hidden layer z_{ks} is known, the cost function is given by

$$Q(z) = Q^h(z) + Q^o(z) =$$
$$= \sum_{s=1}^{S}\left[\sum_{k=1}^{K}\left(\sum_{i=0}^{I} w_{ki}^h x_{is} - f_k^h\left(z_{ks}\right)^{-1}\right)^2 + \sum_{j=1}^{J}\left(\sum_{k=0}^{K} w_{jk}^o z_{ks} - f_j^o\left(y_{js}\right)^{-1}\right)^2\right]. \tag{3}$$

The initial values of weight have been randomly adjusted in range of [0.05, 0.95] at the beginning of initialization algorithm. In next step of the initialization algorithm, the system of equations will be compiled and solved in order to acquire the updating weight values of neural network through the cost function. The particular sensitivities of cost function are expressed as

$$\frac{\partial Q}{\partial z_{ks}} = \frac{\partial Q^h}{\partial z_{ks}} + \frac{\partial Q^o}{\partial z_{ks}} =$$
$$= -\frac{2\left(\sum_{i=0}^{I} w_{ki}^h x_{is} - f_k^h\left(z_{ks}\right)^{-1}\right)}{f_k^h\left(z_{ks}\right)'} + 2\sum_{j=1}^{J}\left(\sum_{m=0}^{K} w_{jm}^o z_{ms} - f_j^o\left(y_{js}\right)^{-1}\right) w_{jk}^o, \tag{4}$$

where:

k – number of hidden neurons, 1, 2, …, K,

z_{0s} – bias of hidden neurons, must be $z_{0s} = 0$ for all K.

By using Taylor series approximation, the cost function is given by

$$Q(z + \Delta z) = Q(z) + \sum_{k=1}^{K}\sum_{s=1}^{S}\frac{\partial Q}{\partial z_{ks}}\Delta z_{ks} \approx 0 . \tag{5}$$

Now, the change in the output vector of hidden layer can be re-written of the equation (5) as

$$\Delta z = -\rho\frac{Q(z)}{\left\|\nabla Q\right\|^2}\nabla Q, \tag{6}$$

where ρ denotes step size.

Using this algorithm as the initialization process presents a few of loops of the weights determination and the modifications of hidden layer output by the cost function. Should, this algorithm continuous until acquire a desired cost function minimal value, then ones is one of the possible neural network learning methods. The results in comparison with efficient Levenberg-Marquardt algorithm are much slower in term of its speed. Hence, the proposed method is used for optimal initialization of neural network parameters and then continues by the standard learning method of neural networks.

3 APPLICATION OF INITIALIZATION METHODS

For a neural network realization and a comparison of initialization method, Matlab® programme was used. This programme uses for neural network initialization especially most widespread Nguyen-Widrow initialize method. The method based on sensitivity analysis was realized as individual m-file link to existing Neural Network toolbox within the consistent name of variables.

The testing of initialization method is suitable carries out on different well-known testing problems. Presently, the program was tested with XOR logical problem. The neural network structure was composed of two layers and 8 hidden neurons with different activation functions (tansig, logsis, tadbas, satlin, satlins). Training data was extended about an intermediate values given by range of logical values 0 and 1 in real application (converted in range of [0, 1]).

x_1=[0 0.087 0.087 0 1 1 0.478 0.478 0 0 0.087 0.087 1 0.478 0.478 1]

x_2=[1 1 0.478 0.478 0 0.087 0.087 0 0 0.087 0.087 0 1 1 0.478 0.478]

y=[1 0.478 0.478 0.478 1 0.478 0.478 0.478 0 0.087 0.087 0.087 0 0.087 0.087 0.087]

For the initialization method based on sensitivity analysis, a parameter ρ was chose as 0.02 and the initial weight values was chosen in range of [-0.002, 0.002].

In the simulation, the output values for all combination input parameters x_1 and x_2 were determined with step size 0.01 and in range of [0, 1]. For better visualization of acquired neural network model, the results of XOR problem were depicted by 3D graphs for particular activation functions. The neural network models acquired by Nguyen-Widrow initialization for different activation functions are shown in Fig. 3 – Fig. 7 and the neural network models acquired by the initialization method based on sensitivity analysis are depicted in Fig. 8 – Fig. 12.

The comparison of randomly selected neural models initialized above mentioned the initialization methods is expressed by graphs with percentage values in Fig. 2, where the particular columns express if the values of neural network model exist in given range.

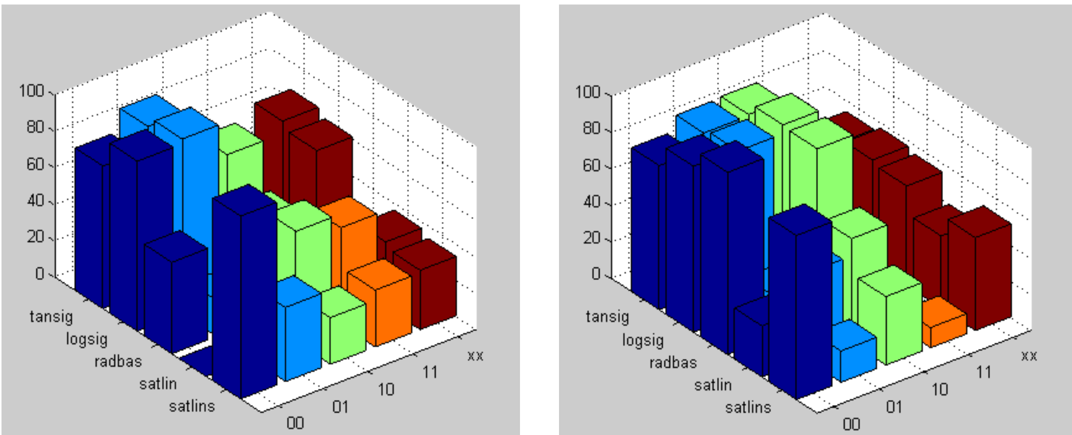

Fig. 2 The percentage evaluation of neural network model with Nguyen-Widrow initialization method (left) and the sensitivity analysis initialize method (right)

4 CONCLUSIONS

This paper introduces the initialization methods of neural networks. The method based on sensitivity analysis is described in more detail. Within the realization this initialization method, the source code was designed in Matlab programme in sequence on current Neural Network Toolbox and its own structured variable *net* for neural network.

The common known problem XOR was used for testing of initialization methods. In next research, the testing of initialization method based on sensitivity analysis is supposed for different testing problems.

The research work was obtained during the completion of Specific Research, supported by the Czech Ministry of Education, Young and Sports.

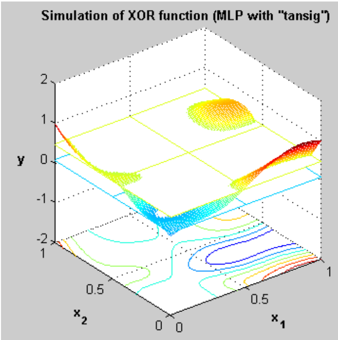

Fig. 3 Nguyen-Widrow initialization ("tansig")

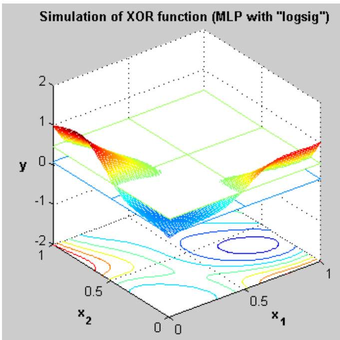

Fig. 4 Nguyen-Widrow initialization ("logsig")

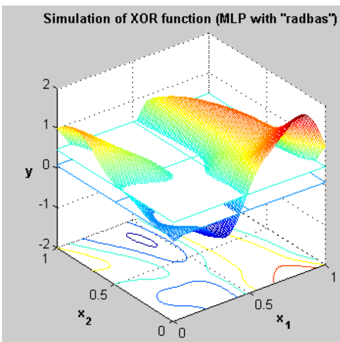

Fig. 5 Nguyen-Widrow initialization ("radbas")

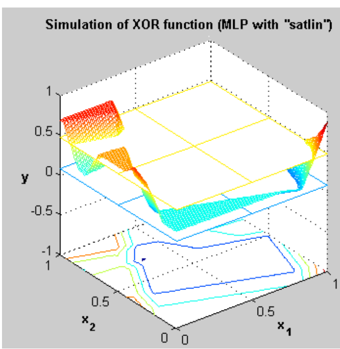

Fig. 6 Nguyen-Widrow initialization ("satlin")

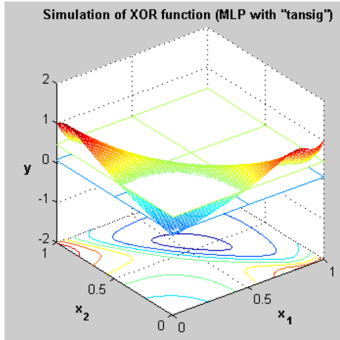

Fig. 8 Sensitivity analysis ("tansig")

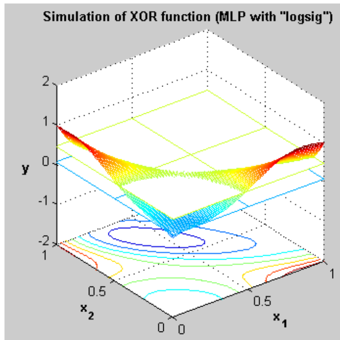

Fig. 9 Sensitivity analysis ("logsig")

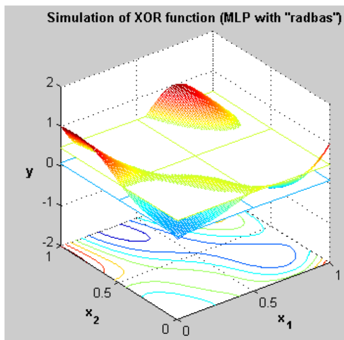

Fig. 10 Sensitivity analysis ("radbas")

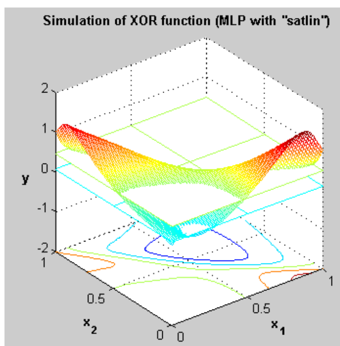

Fig. 11 Sensitivity analysis ("satlin")

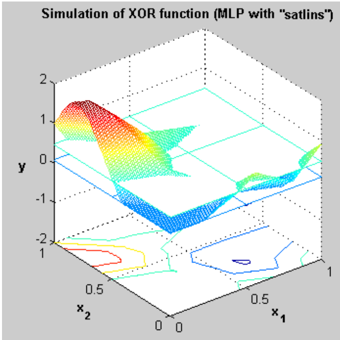

Fig. 7 Nguyen-Widrow initialization ("satlins")

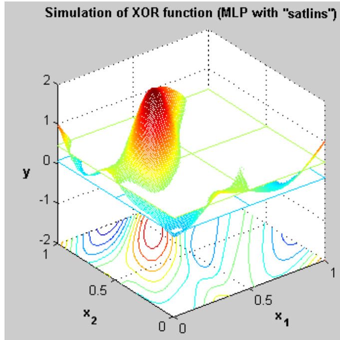

Fig. 12 Sensitivity analysis ("satlins")

REFERENCES

[1] BERDINAS B. et al. A new initialization method for neural networks using sensitivity analysis. *International Conference on Mathematical and Statistical Modeling in Honor of Enrique Castillo.* San Diego, June 28-30, 2006.

[2] CASTILLO E., BERDINAS B. G., ROMERO O. F. & BETANZOS A. A. A Very Fast Learning Method for Neural Networks Based on Sensitivity Analysis. *Journal of Machine Learning Research.* 2006, Vol. 7, pp. 1159-1182. Available from www: <http://jmlr.csail.mit.edu/papers/volume7/castillo06a/castillo06a.pdf>. ISSN 1533-7928.

[3] DOSTÁL Z. An optimal algorithm for bound and equality constrained quadratic programming problems with bounded spectrum. *Computing.* 2006, Vol. 78, No. 4, pp. 311-328. Available from www: <http://www.ingentaconnect.com/>. ISSN 1436-5057.

[4] KIM Y. K. & RA J. B. Weight value initialization for improving training speed in the backpropagation network. *International Joint Conference on Neural Networks (IJCNN'91).* 1991, Vol. 3, pp. 2396-2401.

[5] NGUYEN D. & WIDROW B. Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. *Proceedings of the International Joint Conference on Neural Networks.* San Diego, June 1990, Vol. 3, pp. 21-26.

[6] PAPAKOSTAS, G. A. et al. Two-stage hybrid tuning algorithm for training neural networks in image vision applications. *Signal and Imaging Systems Engineering.* 2008, Vol. 1, No. 1, pp. 58-67. Available from www: <http://www. inderscience.com/storage/ f122571398101146.pdf>. ISSN (online): 1748-0701.

[7] RUSSO, A. P. Constrained neural networks for recognition of passive sonar signals using shape. *IEEE Conference on Neural Network for Ocean Engineering.* Washington DC. 1991, pp. 69-76.

[8] ŘEPKA, M. *Využití umělých neuronových sítí pro modelování horninových procesů. Disertační práce*. Ostrava : Institut ekonomiky a systémů řízení, Hornicko-geologická fakulta VŠB-TU Ostrava, 2008. 113 s.

[9] YAM, J. Y. F., CHOW, T. W. S. & LEUNG C. T. A new method in determining the initial weights of feedforward neural networks. *Neurocomputing*. 1997, Vol. 16, No. 1, pp. 23-32. ISSN 0925-2312.

Reviewers:

prof. Ing. Ivan Taufer, DrSc., University of Pardubice

doc. RNDr. Ing. Miloš Šeda, Ph.D., Brno University of Technology