UDC 681.327.12 (045)

[1]D. S. Gerasimova,
[2]V. I. Serdakovsky

ALGORITHM OF CONSTRUCTING EXPERT SYSTEM, BASED ON ANN TECHNOLOGY

[1] National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute," Kyiv, Ukraine
[2] University "Ukraine", Kyiv, Ukraine
E-mails: [1]daniela.gerasimova@ukr.net, [2]witdba@gmail.com

Abstract—*The algorithm for constructing expert systems through training multilayer artificial neural network. The algorithm optimization scales artificial network method Leuven Berg-Marquardt. Efficiency studies of ANN shown on the instrument classification of EEG.*

Index Terms—Expert system; artificial neural network; Levenberg-Marquardt method; radial basic function, electroencephalogram (EEG).

I. INTRODUCTION

Despite the fame approaches of expert systems in practice, them inherent drawbacks are that in most cases the person who decides on the basis of the findings of experts to be sure the appropriate professional competence of experts, which is not always true.

Expert opinion must be based on more perfect knowledge of the subject area under uncertainty.

One of the possible solutions to this problem is to use for analyze the problem domain system of artificial neural networks that may contribute to an objective judgment of experts on the subject.

II. PROBLEM STATEMENT

It is proposed to include a multilayer ANN hidden layers as experts on the subject area, which should provide objective expert judgment on the issue under study. Examples include experts ANN shown in Fig. 1 [1].

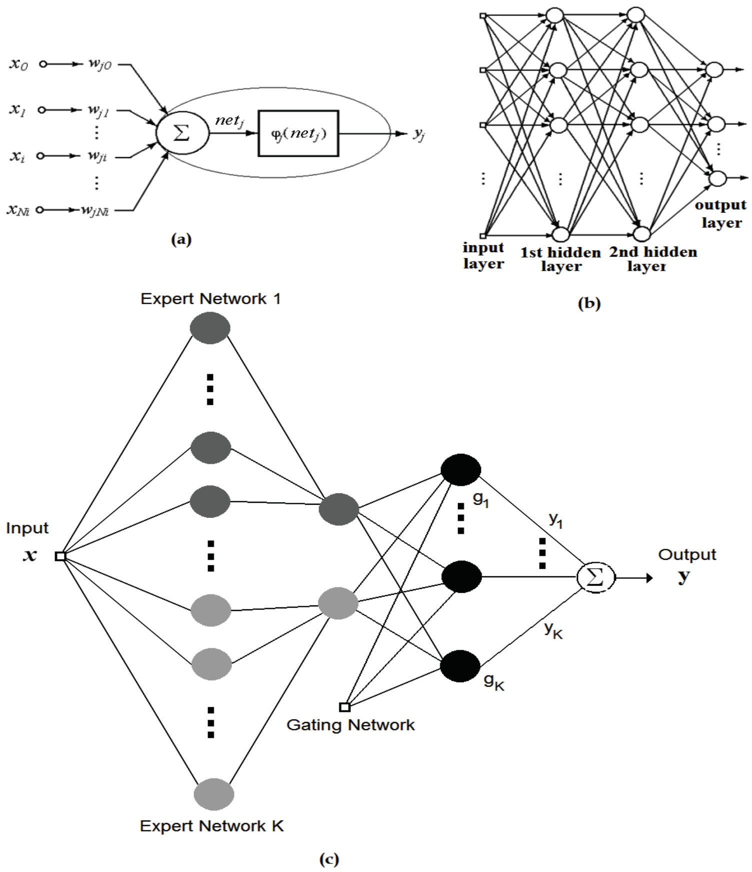

Fig. 1. (a) Nonlinear model of an artificial neuron; (b) FNN configuration with two hidden layers; (c) Extended modular neural network configuration with *K* experts [6]

Notable among neural expert systems occupy predictive models are used, for example, to predict disease outcomes.

Forward neural network model can be used in Demographic and Health Organization [2], [3].

Another application of Newtonian optimization strategy is the Levenberg-Marquardt algorithm. When using the exact value of Hessian replaced approksymuyuchym value that is calculated based on information contained in the gradient of regularization, considering some factors [4].

To describe this method present objective function as corresponding to the existence of a single training set

$$E(w)=\frac{1}{2}\sum_{i=1}^{M}[e_i(w)]^2,$$

$$e_i=[y_i(w)-d_i], \tag{1}$$

$$J(w)=\begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \frac{\partial e_1}{\partial w_2} & \cdots & \frac{\partial e_1}{\partial w_n} \\ \frac{\partial e_2}{\partial w_1} & \frac{\partial e_2}{\partial w_2} & \cdots & \frac{\partial e_2}{\partial w_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial e_M}{\partial w_1} & \frac{\partial e_M}{\partial w_2} & \cdots & \frac{\partial e_M}{\partial w_n} \end{bmatrix},$$

gradient vector and approcsimated Hessian matrix, corresponding objective function,

$$g(x)=\exp\left[-\frac{\|x-c\|^2}{\sigma^2}\right],$$

defined as

$$g(w)=[J(w)]^{\mathrm{T}}e(w), \tag{2}$$

$$G(w)=[J(w)]^{\mathrm{T}}J(w)+R(w), \tag{3}$$

where $R(w)$ — components of Hessian, that contain relatively higher derivatives. The essence of the approach Levenberg–Marquardt is approximation of $R(w)$, using factor of regulariztion νI, which variable parameter ν, called Levenberg–Marquardt parameter, is a scalar quantity, that varies during optimization. Thus, Hessian matrix at k th step of the algorithm becomes:

$$G(w_k)=[J(w_k)]^{\mathrm{T}}J(w_k)+\nu_k I. \tag{4}$$

At the beginning of training,when the actual value w_k is still far from the desired solution, using the value ν_k, that far exceeds the actual value of the matrix $[J(w_k)]^{\mathrm{T}}J(w_k)$. In this case, the Hessian actually replaced to factor: $G(w_k)\equiv\nu_k I$, and the direction chosen by minimizing gradient descent:

$$p_k\equiv-\frac{g(w_k)}{\nu_k}.$$

As reducing errors and closer to the desired solution decreases the parameter ν_k and the first term in (3) begins to play an increasingly important role.

The effectiveness of the algorithm affects educated choice of ν_k. Too much importance as the initial optimization should decrease until reaching zero at the actual decision to close pursuit. There are different ways of selecting this value, but we limit ourselves to just one original method proposed by D. Marquardt. Let the objective function value on k th and $(k-1)$-th iteration steps are denote as E_k and E_{k-1}, and parameter ν values – ν_k, ν_{k-1}. Reduction of coefficient ν values denote $r>1$. According to the classical algorithm LM value changes as follows:

– if $E(\nu_{k-1}/r)\le E_k$ then $\nu_k=\nu_{k-1}/r$ take;

– if $E(\nu_{k-1}/r)>E_k$ and $E(\nu_{k-1})<E_k$, then accept $\nu_k=\nu_{k-1}$;

– if $E(\nu_{k-1}/r)>E_k$ and $E(\nu_{k-1})>E_k$, then gradually increase m to reach ν just mentioned $E(\nu_{k-1}r^m)\le E_k$, while taking $\nu_k=\nu_{k-1}r^m$.

This procedure is performed to change the value of the moment in which the so-called reflection coefficient fidelity q, calculated by the equation

$$q=\frac{E_k-E_{k-1}}{[\Delta w_k]^T g_k+1/2[\Delta w_k]^T G_k\Delta w_k},$$

reaches a value close to one. This quadratic approximation of the objective function has a high degree of coincidence with the true value, which indicates the proximity of the optimal solution. In this situation factor of regularization may be omitted ($\nu_k=0$), the process of Hessian reduced to direct first-order approximation, and LM algorithm turns into Newton algorithm, characterized quadratic convergence to the optimal solution.

III. PROBLEM SOLUTION

Modified modular ANN is trained, using algorithms, adapted in accordance with model of gauss shift [5], as well as back propagation algorithm errors, including calculation of gradient descent.

We use some symbols: the structure has experts MLP (indexed to $i=\overline{1,K}$) with hidden layers (indexed to $i=\overline{1,L}$) of neurons. In addition, there is a network of MLP LPas layers (indexed to $\overline{l=1,q^{(l)}}$) of neurons in each layer (indexed $\overline{j=1,q^{(l)P}}$).

The main point of this algorithm is the increment of synaptic weights network, running on several levels. Synaptic weight updated by experts according to:

$$w^{(l)}_{e_i^{(K)}},(n+1)=w^{(l)}_{e_i^{(K)}},(n)+\eta\delta^{(l)}_{e_i^{(K)}}(n)y_j^{(l-1)}(n),$$

where η is the learning speed and gradient $\delta^{(l)}_{e_i^{(K)}}$ for the neurons in the output layer:

$$\delta^{(l)}_{e_i^{(K)}}(n)=h_i(n)e_i(n)^{(l)}\varphi^{(l)}_{e_i^{(K)}}\left[v^{(l)}_{e_i^{(K)}}(n)\right],$$

where $e_i=y_e(K)-d_i$. The gradient for the neurons of hidden layers is calculated as:

$$\delta^{(l)}_{e_i^{(K)}}(n)=\varphi^{(l)}_{e_i^{(K)}}\left[v^{(l)}_{e_i^{(K)}}(n)\right]\sum_{m=1}^{q^{(l+1)}}\delta_m^{(l+1)}(n)w^{(l+1)}_{me_i^{(K)}}(n).$$

Updates synaptic weight of network is performed by

$$a^{(l)}_{p,j}(n+1)=a^{(l)}_{p,j}(n)+\eta\delta^{(l)}_{p_i}(n)y_j^{(l-1)}(n),$$

where the original gradient layer:

$$\delta^{(l)}_{pi}(n)=\left[h_i(n)-g_i(n)\right]\varphi^{(l)}_{pi}\left[v^{(l)}_{pi}(n)\right].$$

The error is the difference between h_i and g_i. Gradient hidden layers by the equation

$$\delta^{(l)}_{pi}(n)=\varphi^{(l)}_{pi}\left[v^{(l)}\sum_{m=1}^{q^{(l+1)}}\delta_m^{(l+1)}(n)a^{(l+1)}_{mpi}(n)\right].$$

Thus, the error is returned to its network of hidden layers.

This algorithm is used for processing EEG fragment using RBF algorithms and optimization descent (the distance from the point of extremum) and Levenberg–Marquardt approaches the fixed points.

The result of simulation of multilayer ANN with expert in hidden layer for fragment of EEG is presented on Fig. 2.

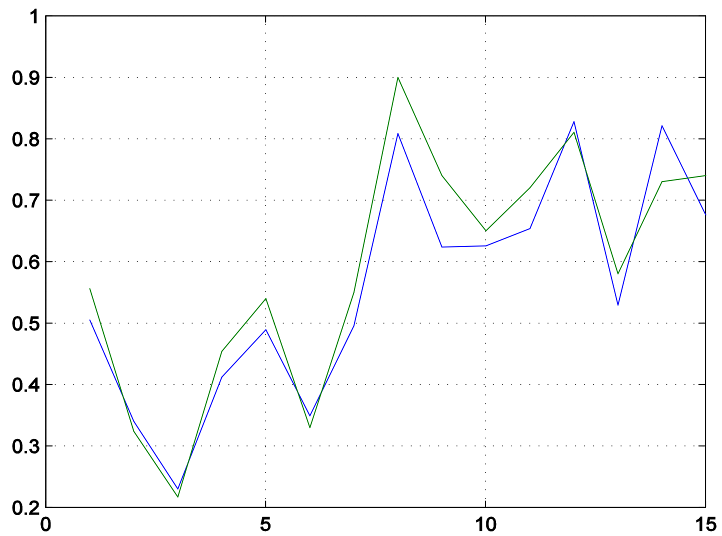

Fig. 2. Graphics of etalon and result of leaning of ANN for fragment of EEG

IV. Conclusions

The algorithm for solving the problem of optimizing the parameters of multilayer neural networks and constructing the set of expert reports.

Developed appropriate software implementation language *C*.

References

[1] Y. H. Cheng and C. S. Lin, “Learning algorithm for radial basis function network with the capability of adding and pruning neurons.” *Proc. 1994 Conf. ICNN.* Orlando, 1994, pp. 797–801.

[2] S. Osowski and K. Siwek, “Selforganizing neural networks for short term load forecasting in power system.” *Engineering Applications of Neural Networks (EANN)*, pp. 253–256, 1998.

[3] T. Desell, S. Clachar, J. Higgins, and B. Wild, “Evolving Deep Recurrent Neural Networks Using Ant Colony Optimization.” *Evolutionary Computation in Combinatorial Optimization.* vol. 9026, pp. 86–98, 2015.

[4] M. S. Popova and V. V. Strijov, “Building superposition of deep learning neural networks for solving the problem of time series classification.” *Systems and Means of Informatics*, vol. 25, no. 3, pp. 60–77, 2015.

[5] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." [E-resource] arXiv.org. 2015. http://arxiv.org/abs/1502.03167.

[6] N.Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal of Machine Learning Research*, 15: 1929–1958. 2014.

[7] Z. Li, C. Chang, F. Liang and others, "Learning Locally-Adaptive Decision Functions for Person Verification." *Computer Vision and Pattern Recognition: Proceedings of the 2013 IEEE Conference*. 2013, pp. 3610–3617.

Received January 27, 2016

Gerasimova Dana. Master.
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
Education: National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine (2017)
E-mail: daniela.gerasimova@ukr.net

Serdakovsky Vitaliy. Graduate student.
University "Ukraine", Kyiv, Ukraine.
Education: National technical university of Ukraine "KPI", Kyiv, Ukraine (2003).
Research interests: data mining, ANN.
Publications: 4.
E-mail: witdba@gmail.com

Д. С. Герасімова, В. І. Сердаковський. Алгоритм побудови експертних систем на ґрунті штучних нейронних мереж

Розглянуто алгоритм побудови експертних систем за допомогою навчання багатошарової штучної нейронної мережі. Запропоновано алгоритм оптимізації вагів штучної мережі за методом Левенберга—Марквардта. Ефективність навчання штучної нейронної мережі продемонстровано на прикладі класифікації електроенцефалограм..

Ключові слова: експертна система, штучна нейронна мережа, метод Левенберга—Марквардта, радіальна базисна функція, електроенцефалограми.

Герасімова Дана Сергіївна. Магістр.
Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», Київ, Україна
Освіта: Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», Київ, Україна, (2017).
E-mail: daniela.gerasimova@ukr.net

Сердаковський Віталій Ігорович. Аспірант.
Університет «Україна», Київ, Україна.
Освіта: Національний технічний університет України «КПІ», Київ, Україна, (2003).
Напрямок наукової діяльності: Створення розподілених баз даних, штучні нейронні мережі.
Публікації: 4.
E-mail: witdba@gmail.com

Д. С. Герасимова, В.И. Сердаковский. Алгоритм построения экспертных систем на базе искусственных нейронных сетей

Рассмотрен алгоритм построения экспертных систем на базе искусственных нейронных сетей. Предложен алгоритм оптимизации весов искусственной нейронной сети на основе алгоритма оптимизации Левенберга–Марквардта. Эффективность обучения сети проиллюстрирована на примере классификации электроэнцефалограм.

Ключевые слова: экспертная система; искусственная нейронная сеть; радиальная базисная функция; метод Левенберга–Марквардта, электроэнцефалограмы.

Герасимова Дана Сергеевна. Магистр.
Национальный технический университет Украины «Киевский политехнический институт им. Игоря Сикорского», Киев, Украина.
Образование: Национальный технический университет Украины «Киевский политехнический институт им. Игоря Сикорского», Киев, Украина, (2017).
E-mail: daniela.gerasimova@ukr.net

Сердаковский Виталий Игоревич. Аспирант.
Образование: Нацональный технический университет «КПИ» (2003).
Направление научной деятельности: разработка и администрирование распределенных баз данных, искусственные нейронные сети.
Публикации: 4.
E-mail: witdba@gmail.com