[Sharkey et al, 1997] Amanda J.C. Sharkey, Noel J. Sharkey, Combining diverse neural networks. - The Knowledge Engineering Review. - 1997. - V. 12: 3. -P. 1-17.

[Giacinto et al, 2001] Giorgio Giacinto, Fabio Roli, Design of effective neural network ensembles for image classification purposes. - Image and Vision Computing. -Elsevier Science. - 2001. - V. 19. - P. 699-707.

[Crepet et al, 2000] Agnes Crepet, Helen Paugam-Moisy, Emanuelle Reynaud, Didier Puzenat, A modular neural model for binding several modalities/ In Proc. of IC-AI'2000, Int. Conf. of Artificial Intelligence. - Las Vegas, USA. - 2000. - P. 921 – 928.

[Tang et al] Bin Tang, Malcolm I. Heywood, Michael Shepherd, Input Partitioning to Mixture of Experts

[Happel et al, 1994] Bart L.M. Happel, Jakob M.J. Murre, The design and evolution of modular neural network architecture. - Neural Networks. – 1994. – V. 7. – P. 985-1004.

[Kussul et al, 2002] Kussul M., Riznyk A., Sadovaya E., Sitchov A., Chen T.Q. A visual solution to modular neural network system development // Proc. of the 2002 International Joint Conference on Neural Networks IJCNN'02. - Honolulu, HI, USA. - 2002. - 12-17 May. - Vol.1. - N749.

[Middleton, 1960] D. Middleton, An Introduction to Statistical Communication Theory, V. 2., 1960.

[Reed et al, 1999] R.D. Reed and R.J. Marks, Neural Smithing. MIT Press. - 1999. – P. 346.

Author information

Reznik A., Galinskaya A. - Department of neural technologies, Institute of Mathematical Machines and systems NAS Ukraine. neuro@immsp.kiev.ua

APPLICATION OF THE SUFFICIENCY PRINCIPLE IN ACCELERATION OF NEURAL NETWORKS TRAINING

Krissilov V.A., Krissilov A.D., Oleshko D.N.

Abstract: One of the problems in AI tasks solving by neurocomputing methods is a considerable training time. This problem especially appears when it is needed to reach high quality in forecast reliability or pattern recognition. Some formalised ways for increasing of networks' training speed without loosing of precision are proposed here. The offered approaches are based on the Sufficiency Principle, which is formal representation of the aim of a concrete task and conditions (limitations) of their solving [1]. This is development of the concept that includes the formal aims' description to the context of such AI tasks as classification, pattern recognition, estimation etc.

Keywords: neural networks

Introduction

Nowadays developers have a lot of different models of neural networks and algorithms of their training [2, 3] for disposal. Though the scientific researches are permanently carried on in this field, the theory of neural networks is still feebly formalised. However, even now two stages of creation of artificial neural systems could be defined: structural and parametric synthesis. At the first stage, developer has to do the following: choose the model for the network, define its structure and choose the algorithm for its training. The parametric synthesis includes training processes of the created network and verification of the obtained results. Then, depending on verification results, there can be a necessity of return to one of the stages of structural or parametric synthesis. Thus, becomes obvious that creation of the neural system is an iterative process.

Feeble formalisation of these stages results in necessity for the developer of the neural system to solve a number of problems. E.g., at the structural synthesis stage, in case of solving a non-standard task, it is necessary to spend a lot of time for choosing the corresponding model for the network, choosing its structure and training method. The problem of the parametrical synthesis is a considerable training time. If real tasks are being solved without any simplification, then duration of training process for created network could be too long. However, some tasks require spending as less training time as it is possible, e.g., real-time tasks.

The aim of the given article is to offer possible methods to reduce the training time for neural networks with back propagation training algorithm. As such methods are offered: control of procedures of modification and evaluation of weight coefficients, reorganisation of objects in recognition classes. Two possible ways for solving this problem were offered in [4]. The first one was based on choosing the particular functional base for the network. The second method controlled the value of the step of weights modification, considering it from the point of view of a centrifugal force and, adjusting it so that its vector was always directed on an optimum of the set of weights.

In this paper the given problem is considered from the point of view of overtraining the network. In most cases a neural network is trained, while its error will not become equal to zero. It can result in inadmissible spending of time. Though, for most tasks it is *enough* for this error not to exceed some defined value.

Sometimes the level of sufficiency is determined by conditions of the task and required result. However, in most cases this process flows past at an intuitive level and the guided principle is not sufficiently fixed by us. Actually this moment is one of most important in solving similar problems, and optimal value of the varied parameter can depend on many basic values and limitations of the task. Thus, there is a necessity for formalising the given principle, in further – the Sufficiency Principle (SP).

Using SP for training neural networks

Let's consider training of the multilayer back propagation neural network within the frames of solving the classification problem.

Three kinds of errors can be picked out in the training process. Let's name them Elementary Error, Local Error and Global Error. The Elementary Error is the error of a single neuron of the network, for neurons from the output layer it can be evaluated as follows:

$$e_i = Y_i - A_i \qquad (1)$$

where Y_i – standard value, A_i – neuron activation level.

The Local Error is the common average error for all neurons of output layer on a single iteration.

$$E_{Li} = \sqrt{\frac{\sum_{k=1}^{m} e_k^2}{m}} \qquad (2)$$

where m – number of neurons in the output layer of the network, i – number of training process iteration.

The Global Error is obtained as it is shown below:

$$E = \sqrt{\frac{\sum_{i=1}^{n} E_{Li}^2}{n}} \qquad (3)$$

where n – number of training sets in the training sample.

The neural network is considered to be ideally trained if its Global Error is equal to zero [5]. However, usually it is difficult to train the net to such level and sometimes it is even impossible. These hardships are connected with presence in the training sample of similar training sets. Thus, the more of such sets are in the sample, the harder will be to train the net.

The essence of the SP is the rejection from attempt to reach the Ideal in solving the concrete task. Considering the training process from the point of view of SP and Local Error, it is possible to say, that complete recognition ($E_L = 0$) is not always necessary. Usually, in order to refer some object to a specific class, the Local Error just shouldn't exceed some defined δ.

Thus, in the frames of errors considered above, three kinds of applying the SP are represented below. The first one offers to accept the error of a single neuron equal to zero if its Elementary Error lies within some boundaries ($e_i \le \delta_e$; δ_e – elementary sufficiency parameter). The second considers the Local Error of the neural network. If E_{Li} is less or equal to δ_{EL} (δ_{EL} – local sufficiency parameter), then procedure of recounting the weights won't be applied for this training iteration. And the last one offers to stop the training process after the Global Error of the network will reach value of some δ_E (global sufficiency parameter).

The minimal value of each δ depends on kind of the training sample. Let's consider the following characteristics of the sample: its completeness heterogeneity, and contradictoriness. The completeness is characterised by provision of classes with training sets. The number of training sets for each class should be in 3 – 5 times more, than number of its *features* used in the set [6]. Let's evaluate the value of completeness as follows:

$$F_{TS} = \frac{N_F}{N} * 100 \ \% \qquad (4)$$

where N_F – number of classes satisfying to the condition mentioned above; N – number of all classes.

The heterogeneity shows how uniformly the sets are distributed among classes. In order to obtain its value let's take the number of training sets for the i-th class $[C_i]$. Then the mean deviation of this value on sample for the given class is:

$$\overline{\Delta}_{C_i} = \sqrt{\frac{\sum_{k=1}^{N} \left([C_i] - [C_k]\right)^2}{N-1}} ; \quad k \neq i \qquad (5)$$

Let's evaluate the average of distribution for $\overline{\Delta}_{C_i}$ and $[C_i]$, on condition that values are equiprobable:

$$R_\Delta = \frac{\sum_{k=1}^{N} \overline{\Delta}_{C_i}}{N} ; \qquad R_C = \frac{\sum_{k=1}^{N} [C_k]}{N} \qquad (6)$$

Then heterogeneity can be evaluated as following:

$$H_{TS} = \frac{R_\Delta}{R_C} \qquad (7)$$

Contradictoriness is a rate of conflicting sets in the training sample. Conflicting sets have the same features, but distributed to different classes. Thus, contradictoriness can be obtained as following:

$$I_{TS} = \frac{N_I}{N} \qquad (8)$$

where N_I – number of conflicting sets.

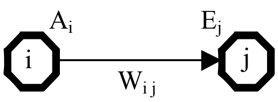
It is obvious, that the lower contradictoriness and heterogeneity, the more narrow can be intervals δ.

Proposed procedures allow to reduce the number of idle changes of weight coefficients. Thus they speed up an approximation of the weights' set to its optimum.

Adjusting the step of weights modification

In original the expression for changing weights between neurons i and j is as following [7]:

$$W_{ij}^{t+1} = W_{ij}^{t} + \alpha * E_j * A_i^t \qquad (9)$$



Pic. 1

where E_j – the error of the j-th neuron;
A_i – the activation level of the i-th neuron;
α – the step of weights modification.

In (9) α is a constant value. However, it is obviously, that if α will be too small, then training will last too long. On the other hand, if α is big, then when the network comes near the minimum point of the error function $E = f(W)$ (E – the Global Error; W – the set of weights) (Pic. 2), it won't be able to reach it. The network will continuously oscillate around this point re-counting its weights and only making worse its characteristics.

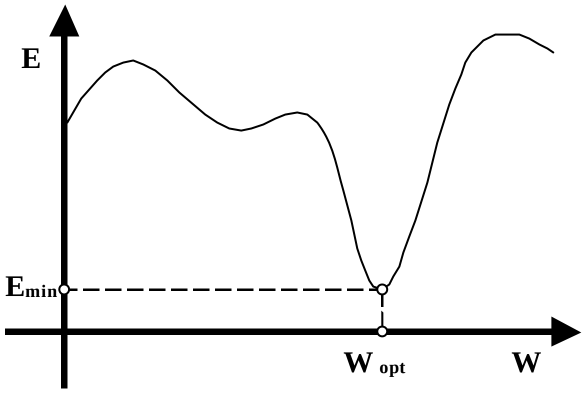
Thus it is necessary to manage the value of α. It is obvious, that if W^{opt} should be reached for the minimal number of iterations, then some average value of α is not acceptable.

Then, at the beginning of the training process some maximum value for α should be set. It will provide a quick approximation to the area of W^{opt}. During the approximation the value of α should be gradually decreased.

$$\alpha_0 = \alpha_{\max} ; \quad \alpha_{t+1} = \alpha_t - \partial\alpha \qquad (10)$$

where $\partial\alpha$ – is decrement of the α.

The offered method of dynamical adjusting the step of weights modification allows keeping the speed of error's decreasing on a sufficient and satisfactory level.



Pic. 2

Reorganization of recognition classes

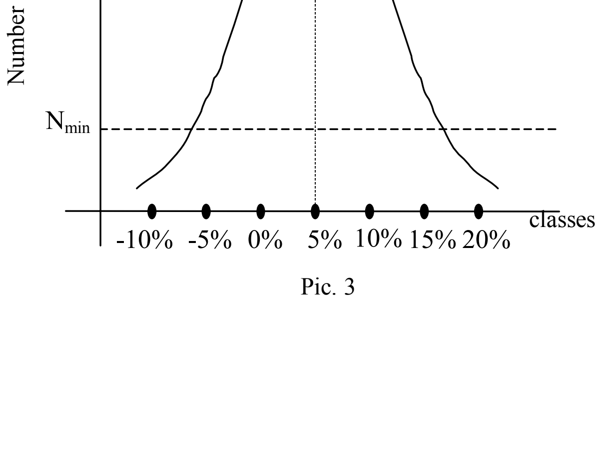
There are number of AI tasks which suppose of possibility of reorganising objects between classes and classes themselves, e.g. creating the forecast based on analysis of time series. This provides two ways for acceleration of the training process in this case. The aim for these ways is to perfect the training sample's characteristics.

There are number of AI tasks which suppose of possibility of reorganizing objects between classes and classes themselves, e.g. creating the forecast based on analysis of time series. This provides two ways for acceleration of the training process in this case. The aim for these ways is to perfect the training sample's characteristics.

Let's consider reducing the number of recognition classes. It is known that the smaller a neural network is, the quicker will be its training. For back propagation neural network its structure is defined by created training sample: the number of recognition classes uniquely defines the number of neurons in the output layer. Thus, reducing the number of classes results in decreasing the size of the network.

However, there is a big number of real tasks, where such losses in precision of classification are inadmissible. Thus, this method can be applied only for tasks without tight restrictions on precision.

It is offered to reduce the number of classes by their combining. In order to find classes for combining, it is necessary to analyse the completeness and heterogeneity of the training sample. If the number of training sets for some class doesn't satisfy the completeness condition, or it is greatly less than in other classes, then recognition of this class by the network will be difficult. For example, results obtained after analysis of the training sample can be the classical normal distribution looking as it is shown on Pic. 3. In order to decrease the heterogeneity of the training sample, classes with number of sets lower than some $N_{\min}$ should be taken and then neighbouring classes should be combined. Then the number of training sets will get over the barrier of $N_{\min}$ and network will be able to train qualitatively and quickly. However, it will also results in reducing the precision in solving the given task. Thus, it is necessary to adjust, using SP, the number of classes recognised by the network with its size.



Pic. 3

Further, let's consider the contradictory training sample. In such a sample classes have both:
objects with low dispersion and located close to the standard of this class – Rules, and objects remote from the standard and located somewhere near the class' boundaries –Eliminations.
Also in the sample there can be classes with high dispersion inside, for which it is impossible to find the standard. Eliminations and Fuzzy classes increase the contradictoriness of the training sample, essentially slow down the training and sometime make it ever impossible. Presence of such elements in the sample can indicates that subsetting of the objects' space on classes was wrong. The solving of this problem is moving Eliminations to other classes and/or forming new classes with lower dispersion.
Thus, the training speed of the network can be increased either by reducing the number of recognition classes, or by moving objects among classes and by forming new classes. The second way increases the training speed by perfecting the training sample, and the first one also by reducing the size of the network.

Conclusions

Thus, three ways of accelerating of the training process for back propagation neural network were considered in this paper.
The first way is based on the analysis of networks' errors. Three levels of errors were described: Elementary Error, Local Error and Global Error. Depending on the kind of the analysed error, different algorithms and software procedures of their implementation were created for obtaining values of the network's weights.
The second way consists in dynamic adjusting the step for changing values of the network's weights. The aim of this method is a minimisation of number of training iterations by reducing the inconsistent adjustments of weights.
The third way considers the reorganization of objects in recognition classes as the way of perfecting characteristics of the training sample: completeness, contradictoriness and heterogeneity.
All proposed ways were applied in forecast and pattern recognition tasks and have brought positive results. They have shown ability to decrease the number of iterations of the training process.
As the test case the task of forecasting the residuals on the bank accounts was solved. The training time was about 30 – 40 hours that was two times less in comparison with original methods.
Applying of them has allowed creating the forecast (for two weeks horizon) with mean-root-square error not greater than 4%.

Bibliography

[1] Krissilov, V.A., Krissilov, A.D. "High-Quality Decision Making by Aim-Oriented Modeling", Proc. of 19-th International Conference of NAFIPS, Atlanta, GA, 2000, pp.241-245
[2] Krissilov, V.A., Oleshko, D.N., Trootnev, A.V. "Applying of neural networks in tasks of intellectual analysis of information", Review of Odessa State Polytechnic University, Vol.2 (8), 1999, pp. 134-139
[3] Dayhoff J. Neural network architectures, New-York: Van Nostrand reinhold, 1991.
[4] Patrick, P. "Minimisation methods for training feed forward Neural Networks", NEUARAL NETWORKS, 1994, Volume 7, Number 1, pp. 1-11
[5] Fausett, L. Fundamentals of Neural Networks. New York: Prentice Hall, 1994.
[6] Bishop C. Neural Networks for Pattern Recognition, Oxford: University Press, 1995.
[7] Patterson D. Artificial Neural Networks, Singapore: Prentice Hall, 1996.

Author information

Victor A. Krissilov – Ph. D., Head of Chair "System Software Design" of Odessa Polytechnic University, App.36., 20 Deribasovskaya Str, Odessa-26, 65026, Ukraine; E-mail: VictorK@OL405.paco.net
Anatoly D. Krissilov – Ph. D., Senior Researcher, Institute for Market Problems and Econo-ecological Research of Ukrainian National Academy of Science 29 Francuzskij Blvrd., Odessa-44, 65044, Ukraine.
Dmitry N. Oleshko – post graduate student of Odessa Polytechnic University; E-mail: boss@ic.ospu.odessa.ua