The Cascade - Correlation Neural Network Architecture in Classification and Forecasting Tasks

Ilona Magdisyuk and Arkady Borisov
Decision Support Systems Group
Technical University of Riga
1 Kalkyu St., Riga LV-1658, Latvia
Phone: +371 7089 530, Fax: +371 782 00 94
e-mail:aborisov@egle.cs.rtu.lv

ABSTRACT: The paper deals with studying special features of the cascade correlation architecture of a neural network as well as possibilities of its adaptation while solving forecasting and classification tasks. The mechanism of the method of network output interpretation operation and the influence of various combinations of error correction methods on the convergence of the network are studied.

KEYWORDS: multi-layer neural network, gradient descent, time-series prediction

1. INTRODUCTION

Among the major problems that arise when neural networks are employed for solving real-world tasks, is the low convergence rate of classical network learning algorithms, e.g. back propagation method (Fahlman and Lebiere, 1990). One of the factors influencing the convergence rate decrease is the simultaneous change of all the weights in the network. Such a procedure is time-consuming. Moreover, it leads to the repeated adjustment of connection coefficients. Another key question when building a neural network is choosing a number of hidden neurons. A network with a small number of hidden units is unable to ensure the required quality of the stated task solution, whereas a network possessing too many layers converges very slowly. Furthermore, the size of a network might be over the size of the training set.

The cascade correlation architecture has been developed to solve the above problems. It combines two key ideas, namely the *cascade architecture* in which the hidden units are added to the network one by one and are not changed after addition, and the *learning algorithm* that creates hidden units (Fahlman and Lebiere, 1990).

2. THE CASCADE CORRELATION ARCHITECTURE

The architecture and the algorithm for cascade correlation neural network learning have been proposed by Fahlman and Lebiere, (1990). This algorithm is constructing a network and correcting the weights at the same time. With this, the number of hidden layers is not specified a priori but is determined during learning. Such a flexibility of the algorithm enables one to build models for solving complicated non-linear tasks, for example a *task of two spirals*.

The cascade correlation architecture algorithm exemplifies the supervised learning. Training is started for a network that only has an input unit and an output unit and no hidden units (Hoehfeld and Fahlman, 1991). The aim of training is to minimise the mean square error of the network output, E:

$$E = 1/2 \sum_{o,p} (y_{op} - t_{op})^2 \quad (1)$$

where y_{op} is the network output for pattern p but t_{op} is the desired output for the given pattern. The aim of hidden unit weights correction is to maximise the value of correlation between the output of a candidate unit and the output error of the main network, C:

$$C = \sum_o \left| \sum_p (y_p - \bar{y})(e_{op} - \bar{e}_o) \right| \quad (2)$$

where $\bar{y}$ and $\bar{e}_o$ are the mean values of outputs and output errors over all patterns of the training set.

After training is completed, a candidate unit is added to the main net. With this, the weights of the added unit are being *frozen*, that is, this unit is not trained during the further network training, and its weights remain unchanged (Squires and Shavlik, 1990). To train the network, the QuickPropagation weights correction algorithm is used. However, other algorithms, for example *delta learning rule* and *Widrow-Hoff rule (*Widrow and Hoff ,1960) can also be applied.

3. SPECIAL FEATURES OF THE CASCADE CORRELATION ARCHITECTURE APPLICATION IN FORECASTING TASKS

The above mentioned neural network architecture is a convenient tool for solving tasks of economical and technical processes forecasting. It can also be applied in sociology. The specifics of neural network application in these tasks is that the values of the parameters that influence proceeding of the process being predicted are not passed to the inputs of the network. Instead, the prehistory of this process development is used as a training set. The part of the training set that consists of several test points (x_1, …, x_n) is then passed to the network inputs. The actual output of the network is then computed and compared to the desired output. If both outputs coincide, the network is adjusted according to the above described algorithm. The next part of the training set is then passed to the network inputs, x_1 being correspondent to x_2 of the previous sample but x_n of this part of values being correspondent to the desired output for the previous part, that is the bias of a vector of a part of the training set to one step takes place.

For the cascade correlation neural network adaptation it is also proposed to apply various activation functions for a two-output unit network as shown in Fig. 1.

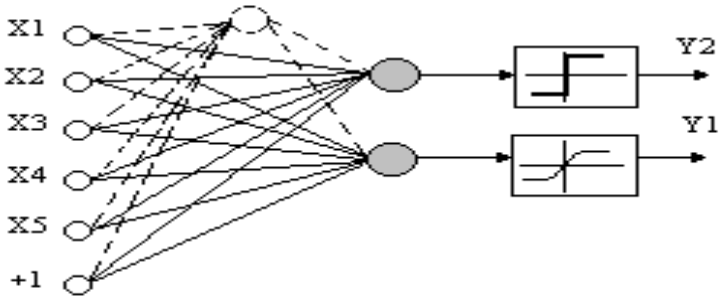

Figure 1: The initial cascade network with two outputs

For one output, Y_1, there is employed a conventional sigmoidal activation function. For the other output, Y_2, a linear activation function is used. This output is applied to determine the growth, fall or stability trend for the process with regard to the present time t. In other words, if the desired value of the network output at time *t+1* is beyond the limits of the interval [x_1, x_5], then the value of output Y_2 is Y_2= +1 when the upper limit is passed over and Y_2= -1 when the lower limit is passed over. Thus we can at once determine the trend of the process as well, not only the actual output of the network. Moreover, the additional output of the network will enable one to interpret output Y_1.

Depending on the nature of the process, network outputs can assume meanings in the interval [0, +∞) but in some cases also in the interval (-∞, +∞), whereas output Y_1 changes within the interval [0, 1]. That is, to use this output in the forecasting task, one should somehow interpret it. Here, two options are possible: (1) to reduce the forecasting task to a classification task with multiple classes and (2) to find out a method that would allow to reduce value Y_1 from the interval [0,1] to the value in the required interval in dependence on the nature of the task. The first way does not require

any transformations of the network output, however it needs addition of the necessary quantity of output elements to the network according to the number of classes. With this, a number of classes is preliminary fixed thus restricting dynamic self-formation abilities of the network. Moreover, by restricting the number of classes, a possibility of network training to recognise the patterns that do not enter the predetermined classes is in advance excluded. Thus, the second way seems to be more reasonable.

By using the value of network output Y_2, it is possible to at once determine the trend of the process and, depending on it, to apply one of the following formulas:

$$x_{t+1} = (x_{5t} - x_{1t})\, Y_1 \tag{3}$$

if $Y_2 = 0$, i.e. the process is stable;

$$x_{t+1} = x_{5t} + (x_{5t} - x_{1t})\, Y_1 \tag{4}$$

if $Y_2 = +1$, i.e. the upper limit is exceeded;

$$x_{t+1} = x_{1t} - (x_{5t} - x_{1t})\, Y_1 \tag{5}$$

if $Y_2 = -1$, i.e. the lower limit is exceeded. The described method of network output interpretation was employed to forecast the behavior of the process shown in Fig. 2.

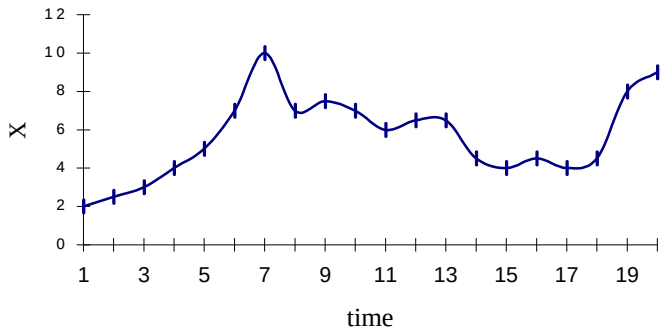

Figure 2: An experimental curve

To perform an experiment, the values of this curve have been fixed for 20 points, namely 15 training points and 5 testing patterns. The process of interpretation is illustrated in Table 1. This table shows three sets of values, for each of which network output Y_2 is different. Depending on the value of Y_2, network output Y_1 was interpreted in accordance with the above described algorithm.

t_1	t_5	Y_1	Y_2	x_{t+1}	O	Error
1	5	0.67	+1	7.01	7	0.01
5	10	0.5	0	6	6	0.00
9	13	0.89	-1	5.61	4.5	1.11

Table 1: Network output interpretation

Without doubt, this method has some extent of error which is difficult to avoid when solving forecasting tasks. The experiments performed have shown that a network gives better results when the process is stable. If the value exceeds the upper limit of the considered interval, the network error is insignificant. The network worst of all responds to sharp '*falls*' of the process. It is possible to avoid appearance of a significant forecasting error by increasing the size of the input vector. That is, the size of the input vector should be selected so that possible fluctuations of the process values are maximally expressed.

4. SPECIAL FEATURES OF USING THE CASCADE CORRELATION NETWORK IN CLASSIFICATION TASKS

Consider some aspects of cascade correlation network application by using an investment task, i.e. a task of determination of the most profitable project for money investment, as an example. Traditional methods usually exploit one or several criteria to select a project. That is, out of the set of criteria they reveal the most significant ones. With this, the information contained in other criteria might be lost. Application of neural networks enables one to avoid both contradictions and information losses.

The advantages of the cascade network application for choosing the most significant criteria have been studied experimentally. The experimental results are given in Table 2.

Criterion	Experiment 1	Experiment 2	Experiment 3	Experiment 4
Investment amount	X	X	X	X
Final profit	X	X	X	X
Average profit per year	------	X	X	X
Liquidity amount	------	------	X	X
Investment period	------	------	------	X
Liquidity period	------	------	------	X
Error	50%	25%	8%	40-60%

Table 2: Experiments aimed to determine the significance of criteria

To start the experiments, two evaluation criteria, *the investment amount* and *the final profit*, have only been used since they seem to be most significant for choosing a project. When these criteria were employed in the cascade correlation network, it was found that the minimum possible classification error was 0.5, i.e. 50%. Such an error is too large. Thus one can conclude that input information is not sufficient for the algorithm's convergence and it is not possible to select an investment project by those two criteria only. Adding of one more criterion, *the average profit per year*, resulted in the minimum classification error of 25%. This means that using the above criteria still does not enable correct selection of a project. After the next criterion, *the liquidity amount*, was added to the network, the minimum error was 0.08, i.e. the error was only 8% over the whole training set. This error value seems to be satisfactory to solve the task. Note that the convergence of the algorithm was already reached at the 108^{th} iteration. From this it follows that the above four criteria, *the investment amount*, *the final profit*, *the average profit per year* and *the liquidity amount* are most important for choosing an investment project.

The determined criteria were then used to solve a task of the best investment project selection. This task is hard solving since the input data that are used during training do not allow to sharply separate objects into two classes. For example, a project with the expected profit of 1000 monetary units may turn to be more profitable than the project promising the profit of 100,000 monetary units. That is why, to improve the network adaptation to the present task, various combinations of weight coefficient correction algorithms have been used.

The original algorithm outlined in (Fahlman and Lebiere, 1990), uses the Quickprop algorithm to correct the weights. The experiments performed with network training, in which the present algorithm is used both in the main network and to train the hidden units, have indicated that the initial value of the mean-squared error is not large and is gradually decreasing. However, at a certain learning step the error sharply increases as shown in Fig. 3. Later on, the error constantly increases. Such a behaviour of the process can be explained by *skipping* the minimum possible error value as well as by the nature of initial data.

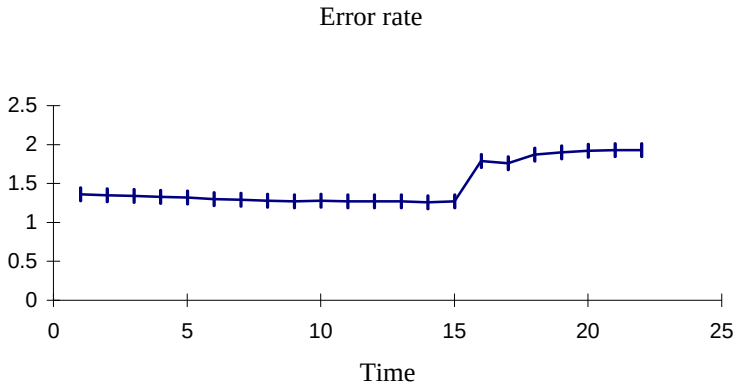

Figure 3: Error changes under the standard algorithm

A number of experiments were also performed that used delta rule to correct the weights in the main network and the QuickProp algorithm to correct the weights in the hidden units. The initial value of the error is also not big provided such combination of algorithms. The value of the error constantly decreases as shown in Fig. 4. However, the algorithm converges very slowly. Several thousand of iterations are required for the algorithm to converge.

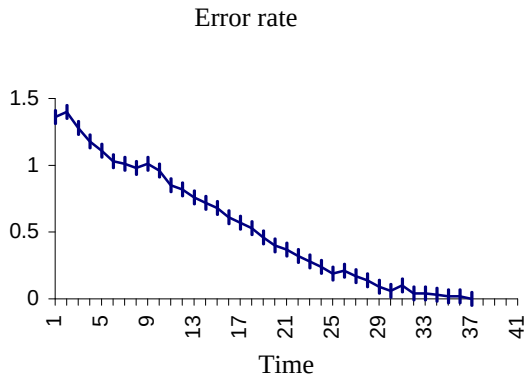

Figure 4: Error changes when delta rule and the QuickProp algorithm are used

The largest convergence rate has been reached when delta rule was used both in the main network and in the hidden units. This process is shown in Fig. 5. As can be seen from the picture, the initial error value is rather big and the process sharply changes its values. However, the algorithm requires only several hundred of iterations to converge.

Figure 5: Error changes when delta rule is used

Thus, one can conclude that a combination of delta rule and QuickProp method is more suitable for solving tasks with multiple classes, whereas application of delta rule both in the main network and in the hidden units is more reasonable in classification tasks where classification into two classes takes place.

5. CONCLUSIONS

The outlined methods of the cascade correlation neural network adaptation enable one to efficiently use this network in classification tasks and time-series prediction, whereas application of the cascade correlation network allows to decrease the size of computations when solving these tasks as well as to increase the precision of classification and prediction what was validated during the experiments performed.

REFERENCES

Fahlman, S.E.; Lebiere, C, 1990, "The Cascade - Correlation Learning Architecture", Advances in Neural Information Processing Systems, Morgan Kaufmann, San Mateo, CA, Vol. II, pp.524-532.

Hoehfeld, M.; Fahlman S.E., 1991, "Learning with limited numerical precision using the Cascade - Correlation algorithm", CMU-CS-91-130.

Squires, Charles S.; Shavlik, Jude W., 1990, "Experimental Analysis of Aspects of the Cascade-Correlation Learning Architecture", Computer Sciences Department, University of Wisconsin-Madison, Machine Learning Research Group Working Paper 91-1.

Widrow, G.; Hoff, M.E, 1960, "Adaptive switching circuits", Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record, Part 4, pp. 96-104.