INFORMATICA, 2001, Vol. 12, No. 1, 101–108

Using the Cascade–Correlation Algorithm to Evaluate Investment Projects

Ilona MAGDISYUK
Technical University of Riga
Kalkyu Str. 1, LV-1658 Riga, Latvia
e-mail: ils.tm@mailbox.riga.lv

Received: November 2000

Abstract. This paper considers some aspects of using a cascade-correlation network in the investment task in which it is required to determine the most suitable project to invest money. This task is one of the most often met economical tasks. In various bibliographical sources on economics there are described different methods of choosing investment projects. However, they all use either one or a few criteria, i.e., out of the set of criteria there are chosen most valuable ones. With this, a lot of information contained in other choice criteria is omitted. A neural network enables one to avoid information losses. It accumulates information and helps to gain better results when choosing an investment project in comparison with classical methods. The cascade-correlation network architecture that is used in this paper has been developed by Scott E. Fahlman and Cristian Lebiere at Carnegie Mellon University.

Key words: neural network and criterion.

1. Introduction

When making a decision about investments, one should answer two questions:

(i) to accept or to reject the given project;

(ii) which of mutually excluding projects should be accepted.

There exist a lot of techniques of investment project evaluation which, in their turn, use different evaluation criteria. Below there are listed most popular ones:

(i) method of accounting rate of return normalisation;

(ii) method of recoupment period determination;

(iii) method of net profit evaluation;

(iv) method of internal rate of return normalisation.

To evaluate an investment project, the above methods are used either separately or in combination. However, if they are used simultaneously, a situation might occur when different methods would yield different contradictory conclusions. For example, one of the methods could indicate that the given project could be accepted, whereas some other method could conclude that this project is not significant enough for acceptance. When comparing projects, the above methods could also lead to the contradictory results. Using of a neural network enables one to avoid such inconsistency.

2. Neural Networks

An artificial neural network is a computing system that consists of a collection of artificial neurons connected with each other. An artificial neuron simulates performance of a biological neuron. Such artificial neuron was first introduced in (McCulloch and Pitts, 1943). However, the first practical development belongs to Widrow and Hoff (1960). In 1960, just these researchers proposed to use a simple non-linear element along with the adaptive algorithm. The essence of this algorithm is that various patterns are forwarded to the inputs of a simple neuron. The neural element transforms input signals into the output signal, the latter is compared with the expected results and if the real output does not coincide with the expected one, the algorithm is being corrected. The samples are forwarded to the outputs one by one until the result is satisfactory.

The main component of this network is a single processing element which receives input signals from the environment via the weighted channels, connections, and computes the output signal. Computation proceeds two phases that are described below.

(1) first, the weighted sum of input signals is computed, i.e., numerous external signals are converted to a single internal one

$$Y = \sum_i x_i w_i.$$

(2) an output signal is computed with the help of the activation function, that is internal signal Y is converted to external signal Z

$$Z = f(Y).$$

In neural networks there are used various activation functions. McCulloch and Pitts (1943) have used a linear separation function. However, in other algorithms applied in neural networks a sigmoid function

$$f(Y) = 1/(1 + \mathrm{e}^{-\lambda Y})$$

or hyperbolic tangent function

$$f(Y) = \bigl(2/(1 + \mathrm{e}^{-2\lambda Y})\bigr) - 1.$$

are used. These functions are more flexible for solving non-linear tasks. Moreover, parameter λ, which defines the inclination of the activation function, enables one to maximally adapt the function for solving a specific task.

3. Cascade-Correlation Architecture

The cascade-correlation neural network learning algorithm has been proposed in (Fahlman and Lebiere, 1990). This algorithm not only constructs a network but also trains the weights. With this, the number of hidden layers is not assigned in advance, but is determined during the process of learning. It means that the topology of a cascade-correlation neural network only depends on the task being solved and on the nature of data forwarded to the network inputs. Such a flexibility of the algorithm allows to construct models for solving very complex non-linear tasks, for example, *the task of two spirals*. Due to this reason this algorithm was used in the present paper to solve the investment task.

The cascade-correlation learning algorithm examplifies the supervised learning. While learning, it constructs the minimal network, that is a network with the minimal possible number of hidden layers (Fahlman and Liebiere, 1990). Learning starts when the network is minimal, i.e., when there is an input layer, an output layer and no hidden layers. For learning, an algorithm is used that minimises the value of the network output error, E:

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

where y_{op} is the network output for pattern p, but t_{op} is the expected output for this pattern. On the output of the network, the above described sigmoid activation function is used. Network learning is considered to be completed when the convergence of the network is achieved, that is the value of the error stops to change or if the value of the error is sufficiently small and does not exceed earlier set maximal error value.

In case if the error value does not meet the above requirements, learning should be continued. For this, a new hidden layer is added to the network. This node is called a candidate node and its output is not activated in the main network at this stage. After a new hidden layer is added, all patterns out of the training sample are then passed through this node. The candidate node learns, that is its weights are being revised. The aim of the candidate node weights' correction is to maximise the value of correlation between the output of the candidate node and network output error, C:

$$C = \sum_{o} \left| \sum_{p} (y_p - \bar{y})(e_{op} - \bar{e}_o) \right|,$$

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

After learning, the candidate-node is added to the main net. The weights of this added node are frozen. That means this node does not learn any more and its weights remain unchanged. Inputs of the main net are inputs for each added node, as well as the output of earlier added node. The output of this node, in its turn, can either be forwarded to the output of the main net or serve as one of inputs for added then hidden units. One by one added hidden nodes thus make a cascade architecture (see Fig. 1) which gave the algorithm its name.

During the process of the cascade-correlation network learning, the gradient descent and the gradient ascent, are used, respectively, to minimise the value of error E and to

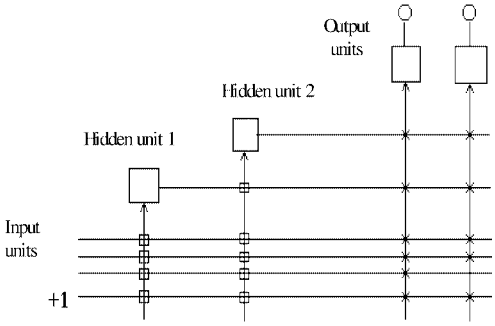

Fig. 1. The cascade-correlation network architecture.

maximise the value of correlation C. For both mentioned values, partial derivatives are computed. For error E, it is computed as

$$\frac{\partial E}{\partial w_{oi}} = \sum_p e_{op} X_{ip},$$

where

$$e_{op} = (y_{op} - t_{op}) f'_p.$$

For correlation C it will look as

$$\frac{\partial C}{\partial w_i} = \sum_p \delta_p X_{ip},$$

where

$$\delta_p = \sum_o \sigma_o (e_{op} - \bar{e}_o) f'_p.$$

If $\frac{\partial E}{\partial w}$ and $\frac{-\partial C}{\partial w}$ will be denoted as S, weights correction formula will look as follows.

$$w(t+1) = w(t) + \Delta w(t),$$

where
$\Delta w_t = \varepsilon S(t)$ if $\Delta w_{t-1} = 0$,
$\Delta w_t = \Delta w_{t-1} S(t) / \bigl(S(t-1) - S(t)\bigr)$ if $\Delta w_{t-1} <>0$ and $S(t)/\bigl(S(t-1)-S(t)\bigr) < \mu$
$\Delta w_t = \mu \Delta w(t-1)$ in all other cases.

Here ε is the error correction step, but μ is the minimal correction step.

The above described weights correction algorithm is known in literature as Quick-Propagation. It takes into consideration the value of weight correction at the previous learning step (Hoehfeld and Fahlman, 1991). Nevertheless, other algorithms might also be used to correct the weights in the cascade-correlation architecture, e.g., Delta rule or Widrow-Hoff rule.

4. Cascade–Correlation Learning Algorithm

The process of network learning consists of six steps. They are listed below.

Step 1. Train the initial net until the mean square error E reaches a minimum.

Step 2. Install a hidden candidate node. Initialise weights and learning constants.

Step 3. Train the hidden candidate node. Stop if the correlation between its output and the network output error is maximised.

Step 4. Add the hidden candidate unit to the main net, i.e., freeze its weights, connect it to the other hidden units, if available, and connect to the network outputs.

Step 5. Train the main net that includes a hidden unit. Stop if the minimum mean square error is reached.

Step 6. Add another hidden unit. Do Steps 2–5 until the mean square error value is acceptable for solving the given task.

5. Experiments

This section describes various experiments performed with the cascade-correlation algorithm in order to determine criteria for investment project evaluation that are most important for the choice of the best investment project. In the experiments used the following data base (see Table 1).

When starting the experiments, only two evaluation criteria were used: *the amount of investments* and *the final profit* (Table 2). These criteria were used as the basis since they appeared to be most valuable for the choice of a project and also as most often met

Table 1

The amoun of investments, t.USD	The final profit, t.USD	The average profit per year, t.USD	The amount of liquidation, t.USD	Investment term, year	Liquidation term, year
100	40	35	10	4	3
100	20	30	20	4	4
100	20	30	10	4	3
60	35	19	5	5	3
150	50	40	20	5	4
60	140	40	30	5	3
60	85	17	10	5	2
100	110	52.5	40	4	2
90	40	32.5	40	4	4
10.2	3.8	4.67	0.8	3	2
35.25	12.75	16	12.75	3	3
40	10	15	5	4	3

Table 2

Criterion	Experiment 1	Experiment 2	Experiment 3	Experiment 4
the amount of investments	X	X	X	X
the final profit	X	X	X	X
the average profit per year	–	X	X	X
the amount of liquidation	–	–	X	X
investment term	–	–	–	X
liquidation term	–	–	–	X
Error value	50%	25%	8%	40–60%

in various classical project evaluation methods. Using these methods within the cascade-correlation network has shown that the minimal possible classification error is equal to 0.5, that is 50%. The great value of this error enables one to conclude that input information is not sufficient for the convergence of the algorithm and it is not possible to choose an investment project according to those two criteria only.

In later experiments, one more criterion, *the average profit per year*, was added as the input signal. Also this criterion can often be met in various investment project evaluation methods. Application of all three above mentioned criteria allowed to come to 25% minimal classification error what was not sufficient yet to make the correct choice of the project. That is why, one more choice criterion, *the amount of liquidation*, was then added. This criterion shows what will be the sum of profit at the moment when the amount of profit will exceed the amount of investments. Shortly speaking, what will be the first profit. This criterion is very important since it is connected with the time factor. It is clear to everybody that money devaluates as time goes. That is why it is very important to take into account the money devaluation factor when choosing a project in order to gain the maximal possible liquidation amount. After this criterion was added to the network, there was obtained the minimal error equal to 0.08, that is the error was only 8% over the whole sample. Such an error is satisfactory to solve the task, the convergence of the algorithm being obtained at iteration 108 yet. From this it follows that the four above mentioned criteria are of great importance while choosing an investment project.

However, a number of experiments were additionally performed with using time criteria: investment term and liquidation term. After these criteria were added, the minimal network error fluctuated between 0.4 and 0.6, the weights not being stabilised.

6. Using Various Weights Correction Algorithms

When performing experiments with algorithms correction weights changing, it was found out that the cascade-correlation algorithm is very sensitive both to changing the algorithms and to changing the activation function. For example, when the QuickPropagation

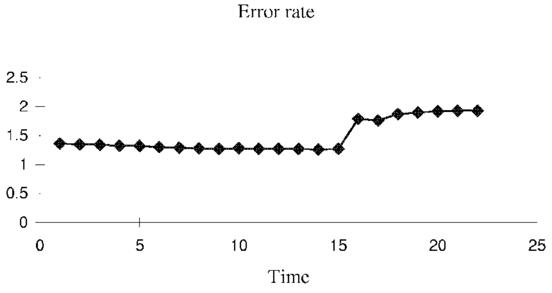

Fig. 2. Error rate changes by using QuickPropagation rule.

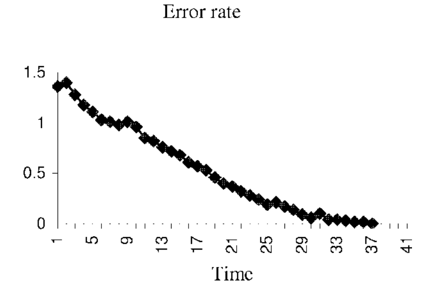

Fig. 3. Error rate changes by using Quickpropagation and Delta rule.

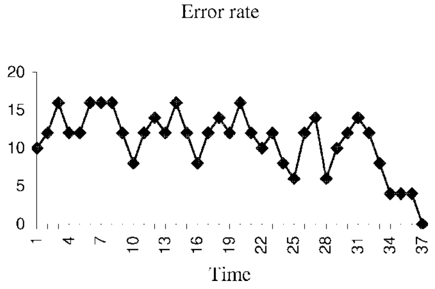

Fig. 4. Error rate changes by using only Delta rule.

algorithm was used both in the main net and for training the hidden layers, the cumulative error was monotonously decreasing but the network output was tending 1 (Fig. 2). However, the situation changed heavily when Delta learning rule was applied to change the weights in the main net, the QuickPropagation algorithm being used for correcting the weights of the hidden layers. Irrespective of the number of hidden layers, the value of the cumulative error monotonously decreases and only one hidden layer is needed to gain the convergence of the algorithm (Fig. 3). 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. 4. 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.

7. Conclusion

The performed experiments have shown an extreme flexibility and sensitivity of the cascade-correlation algorithm. It is obvious that these are the features an algorithm for solving complex non-linear economical tasks should possess. The proposed method is not universal, however it is able to unite separate supporters of various economical methods to evaluate investment projects.

References

Fahlman, S.E., C. Liebiere (1990). The cascade–correlation learning architecture. In: *Advances in Neural Information Processing Systems*, Vol. II, Morgan Kaufmann, San Mateo, CA, pp. 524–532.

Hoehfeld, M., S.E. Fahlman (1991). *Learning with Limited Numerical Precision Using the Cascade–Correlation Algorithm*, CMU–CS–91–130.

McCulloch, W.S., W.A. Pitts (1943). Logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, **5**, 115–133.

Rumelhart, D.E., G.E. Hinton, J.B. Williams (1986). Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland (Eds.), *Parallel Distributed Processing*, Vol. 1, MIT Press, Cambridge, MA.

Widrow, G., M.E. Hoff (1960). Adaptive switching circuits. In *Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record*, Part **4**, pp. 96–104.

I. Magdisyuk is working in the Decision Support Group at Technical University of Riga. She is preparing her doctor work. The main idea of her research is to investigate possibilities of neural network adaptation for solving other tasks. The main type of tasks are classification, time series prediction and technical object reliability definition.

Kaskadinio-koreliacinio algoritmo panaudojimas investiciniams projektams įvertinti

Ilona MAGDISYUK

Straipsnyje nagrinėjami kai kurie kaskadinio-koreliacinio tinklo pritaikymai geriausiems investiciniams projektams parinkti. Yra žinoma daug metodų investiciniams projektams atrinkti, besiremiančių vienu arba keliais kriterijais. Neurotinklas yra priemonė informacijai neprarasti, atsisakius nagrinėti kurį nors kriterijų. Tinklas kaupia informaciją ir todėl galima gauti geresnių rezultatų, palyginus su klasikiniais investicijų planavimo metodais. Šiame straipsnyje nagrinėjama kaskadinio-koreliacinio tinklo architektūra, sudaryta Scott E. Fahlman ir Cristian Lebiere iš Carnegie Mellon universiteto.