ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 39-44

Simplified Cascade-Correlation Learning

Mikko Lehtokangas, Jukka Saarinen and Kimmo Kaski

Tampere University of Technology
Electronics Laboratory
P.O. Box 692, FIN-33101 Tampere, Finland
{mikkol, jukkas, kaski}@ee.tut.fi

Abstract. The original Cascade-Correlation architecture includes short-cut connections from inputs to outputs and the hidden units are high order feature detectors. The need of the short-cut connections and high order feature detectors are considered by empirical simulations. We also study reliability of the learning process in which new hidden units are added to the network according to the Cascade-Correlation algorithm. The benchmark problems we used are 8-bit parity problem and nonlinear time series (measured from far-infrared laser) modeling problem. According to our simulations the short-cut connections and high order feature detectors do not increase the performance of the Cascade-Correlation network. The results would suggest that the most important feature of the Cascade-Correlation method is the way in which new hidden units are added one by one to the network. In this way the Cascade-Correlation learning process was found to be very reliable.

1. Introduction

One of the most promising new neural network method is the Cascade-Correlation learning architecture [1]. It was developed to overcome certain problems and limitations which are encountered with the widely used backpropagation algorithm [2]. In this study we consider three features which were proposed in the original Cascade-Correlation method [1]. The original architecture includes short-cut connections from inputs to outputs and the hidden units are created to be high order feature detectors. The need of these two features is studied by making empirical simulations with four different network configurations. The reliability of the learning process in which new hidden units are added to the network one by one is also considered. The simulations are done with two benchmark problems. The first problem is the 8-bit parity problem which is a generalization of the infamous XOR problem. The other problem deals with real world time series data measured from a clean physics laboratory experiment. The time series represent the fluctuations in a far-infrared laser and it was recently used in the Santa Fe Time Series Prediction and Analysis Competition [3].

2 Original Cascade-Correlation Architecture

An example of the original Cascade-Correlation architecture is shown in Fig. 1. The training of this network starts with no hidden units. In other words, the direct input-output connections are trained by minimizing the output error. If the residual error in the output remains unacceptable we attempt to reduce it by adding a new hidden unit to the network. The new unit is trained by maximizing the correlation between its output and the residual error. Only after the training of the new hidden unit is complete its output is connected to the active network, and at the same time its input weights are frozen permanently. Then we train the output weights again by minimizing the output error. This cycle of adding new hidden units and training the output weights is repeated until the error is acceptably small (or until we give up) [1]. One main characteristic of this network configuration is that each new hidden unit receives a connection from each of the network's original inputs and also from every pre-existing hidden unit.

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 39-44

Therefore each new hidden unit forms a new layer to the network and the final network may include feature detectors (or hidden units) which are of relatively high order. In this study we used hyperbolic tangent function as the activation function in the hidden units. The output unit was set to be linear. The training of the hidden units was done with gradient ascent where we used a constant learning rate as is commonly used in the standard backpropagation. The output weights were optimized with linear regression in which no iterative procedures are needed.

3 Variations of Cascade-Correlation Architecture

In this work we study the need of the short-cuts and high order feature detectors in Cascade-Correlation approach by empirical simulations. One way to accomplish this is to make comparative simulations with network configurations which do not include these two features or include only one of them. The possible network variations are shown in Fig. 2. In the first variation (Fig. 2a) we dropped out the short-cut connec-

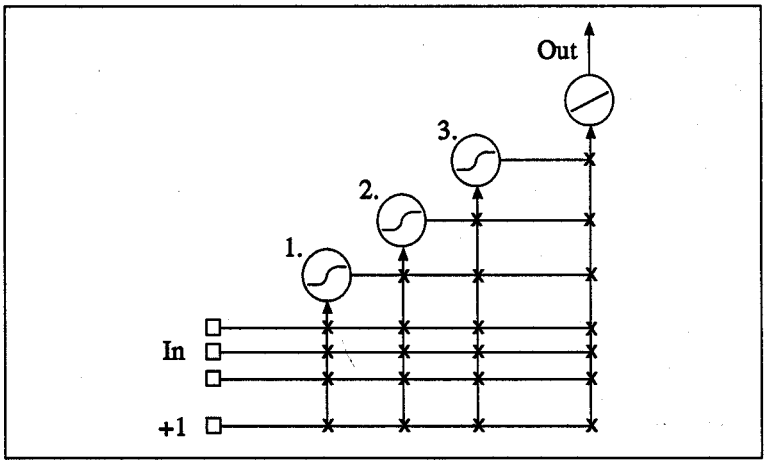

Fig. 1. An example of the original Cascade-Correlation architecture. The crosses represent the network connections.

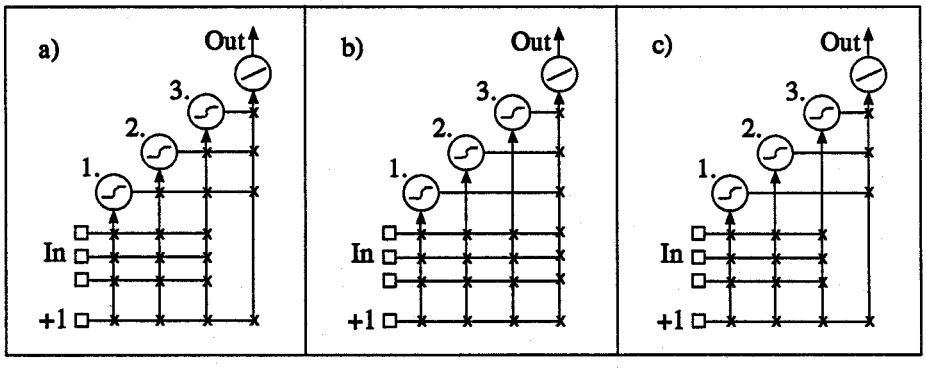

Fig. 2. Examples of the Cascade-Correlation variations. The crosses represent the network connections. a) Cascade-Correlation without short-cuts, b) three-layer perceptron with short-cuts, and c) three-layer perceptron without short-cuts.

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 39-44

tions. In the second one (Fig. 2b) we dropped out the connections between the hidden units. Thus, in this configuration all the hidden units are feature detectors with the lowest possible order. This network is actually a three-layer perceptron network with short-cut connections. The last possible variation in which both of the two features are omitted (Fig. 2c) is just an ordinary three-layer perceptron network. The training of these 'new' network configurations can be done in a very similar manner as the training of the original Cascade-Correlation configuration is done. The main difference is encountered with the networks in which the short-cuts are omitted. The original Cascade-Correlation algorithm starts by training the short-cut connections. In the case when we do not have the short-cuts the only trainable parameters are the bias terms of the output units. However, we can not really talk about training these biases. The best we can do is to set a bias weight to be such that the output is the mean of the desired output sequence. For instance, as we use linear output unit in this study its bias weight value is set to be the mean of the output sequence in the first training phase. After that the hidden units are added one by one according to the Cascade-Correlation algorithm.

4 Learning Results

The learning performance of the four different network configuration is studied by using visually representative learning curves. In other words we plotted the normalized mean square error (NMSE) as a function of the number of hidden units. The NMSE is defined as

$$NMSE = \frac{1}{n\sigma^2} \sum_{i=1}^{n} (d_i - o_i)^2,$$

in which σ^2 is the variance of the desired outputs d_i, and o_i are the network outputs. These plots show how the error goes down as new hidden units are added to the network. Since new hidden units are initialized with small random values in the learning phase each benchmark problem was trained 100 times with different initializations on each trial. The plotted curves are the averages of 100 repetitions. Also, we plotted upper and lower deviation curves on the same picture in order to see the variations between the worst and best training trials. If the variation is small we can consider the learning algorithm to be reliable. The upper deviation curve was obtained as an average of those values which were greater than the average curve and the lower one is the average of those error values which were smaller than the average curve.
When we study the need of the short-cuts and high order feature detectors a comparison based on the above described plots is not sufficient nor fair. When all the four configurations have the same number of hidden units they do not have the same number of weights in them. As the learned information is stored to the weights, a network which has many additional weights may have better chances of extracting more information from the given problem. Thus in this study it would be reasonable to expect that the original Cascade-Correlation architecture is able to learn the problems with fewer hidden units than the three-layer perceptron network. Therefore we plotted also learning curves in which the NMSE is represented as a function of the number of weights in the network. It should also be reminded that a network with more weights requires more computation in the training phase.
8-bit parity problem: In a n-bit parity problem the output is to be on if an odd number of the n inputs are on. The learning results for the 8-bit parity problem are shown in Figs. 3 and 4. From Fig. 3 we can see that the variations between the best and worst trials are relatively small for all the four network configurations. This would suggest that for this problem the method in which new hidden units are added one by one works very well. From Fig. 4 we can see that the Cascade-Correlation architecture without the short-cuts gives slightly better learning performance than the other configurations.

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 39-44

However, the results are so similar that one might prefer to use the three-layer perceptron architecture which has the simplest network structure. In this example the short-cuts and high order feature detectors do not seem to play an important role.
Laser time series data: In this problem we used three input units which represent the previous values of the time series. The aim is to predict the next value of the time series. The learning results for this problem are shown in Figs. 5 and 6. The variations between the best and worst trials are very small with all four network configuration. According to Fig. 6 the three-layer perceptron with short-cuts gives slightly better result compared to other schemes. Otherwise the results are very similar to those which were obtained for the 8-bit parity problem. The short-cuts and high order feature detectors do not seem to increase the learning performance significantly in this experiment either.

5 Discussion

The learning results obtained with the two benchmark problems support those theoretical results in which it has been proven that three-layer perceptron network with sigmoidal hidden units can learn any mapping as well as the more complex feedforward networks. According to the simulations the number of weights seems to be one factor which determines whether a network can learn the given mapping reasonable well or not. In other words, the short-cut connections and the usage of higher order feature detectors did not give any significant advantage since we still needed the same amount of weights in the network as with the three-layer perceptron network. Furthermore, a simple three-layer structure is probably more convenient in practical usage. For instance, after Cascade-Correlation training one can fine-tune the three-layer network with backpropagation. It should be noted that both benchmark problems were nonlinear. It is apparent that if we have a linear problem then the short-cut connections would be quite useful. Also, it is quite possible that in some problems the higher order feature detectors would be useful. One solution to this is that in the training phase we could train two new hidden unit candidates in parallel. One of them would be a low order feature detector and the other a higher order feature detector. After training we would connect that unit to the network which learned the problem better. It is our intention to study this method in the future. In the tested problems we did not encounter any poor local minima on the different trials. Also for all the network configurations the variation between the best and worst trials was relatively small. This indicates clearly, that the Cascade-Correlation learning algorithm in which new hidden units are added one by one is very reliably. This seems to be the most important feature of the Cascade-Correlation method. The presented results do provide new insights on Cascade-Correlation learning and inspire further theoretical and empirical studies.

References

1. S. Fahlman and C. Lebiere, "The Cascade-Correlation learning architecture," Research Report CMU-CS-90-100, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1991.
2. S. Fahlman, "An empirical study of learning speed in back-propagation networks," Research Report CMU-CS-88-162, Carnegie Mellon University, Pittsburgh, PA, 1988.
3. A. Weigend and N. Gershenfeld (Eds.), Time Series Prediction: Forecasting the Future and Understanding the Past, Addison-Wesley Publishing Company, Reading, Massachusetts, 1994.

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 39-44

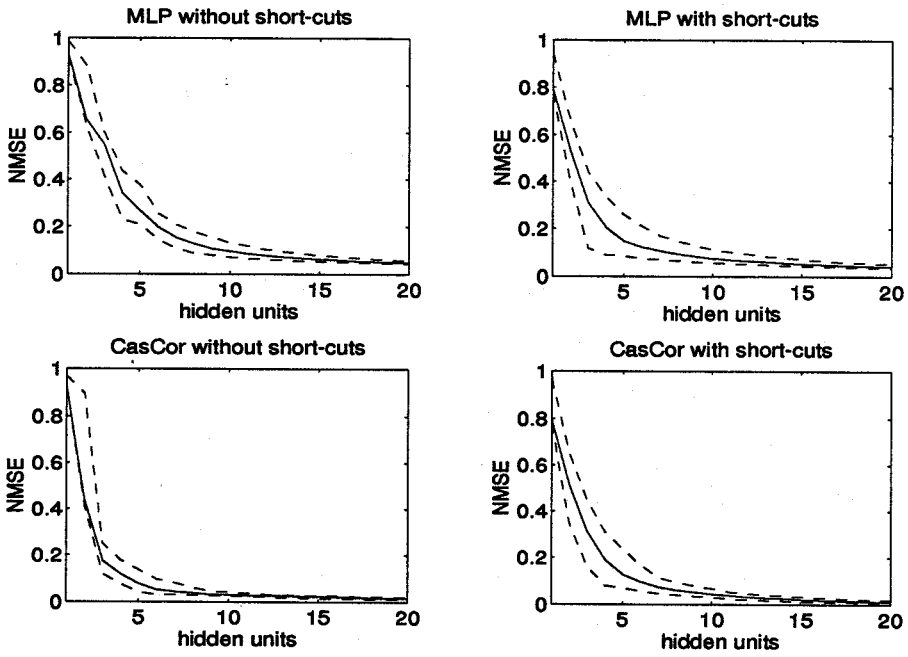

Fig. 3. Training curves for the 8-bit parity problem. Solid lines are the average curves and dashed lines are upper and lower deviation curves, respectively.

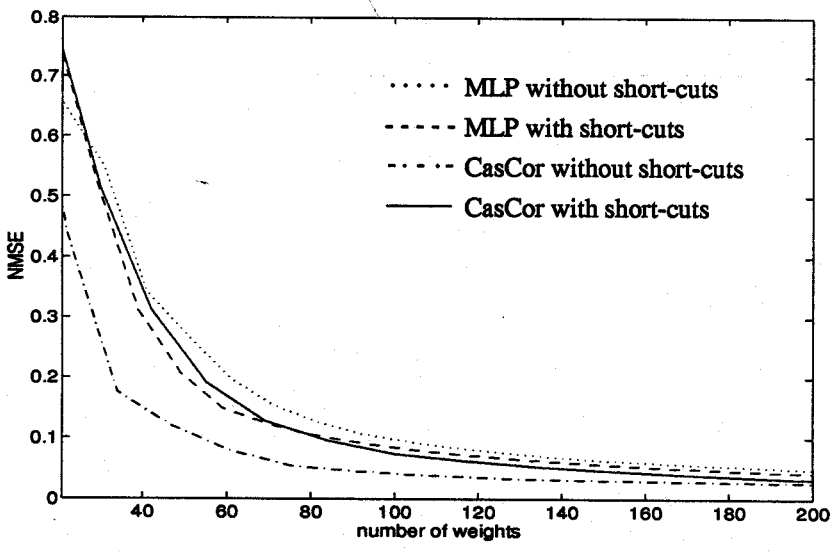

Fig. 4. Additional training curves for the 8-bit parity problem. Dotted line is the average curve for three-layer perceptron without short-cuts, dashed line is for three-layer perceptron with short-cuts, dashdot line is for Cascade-Correlation without short-cuts and solid line is for Cascade-Correlation with short-cuts.

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 39-44

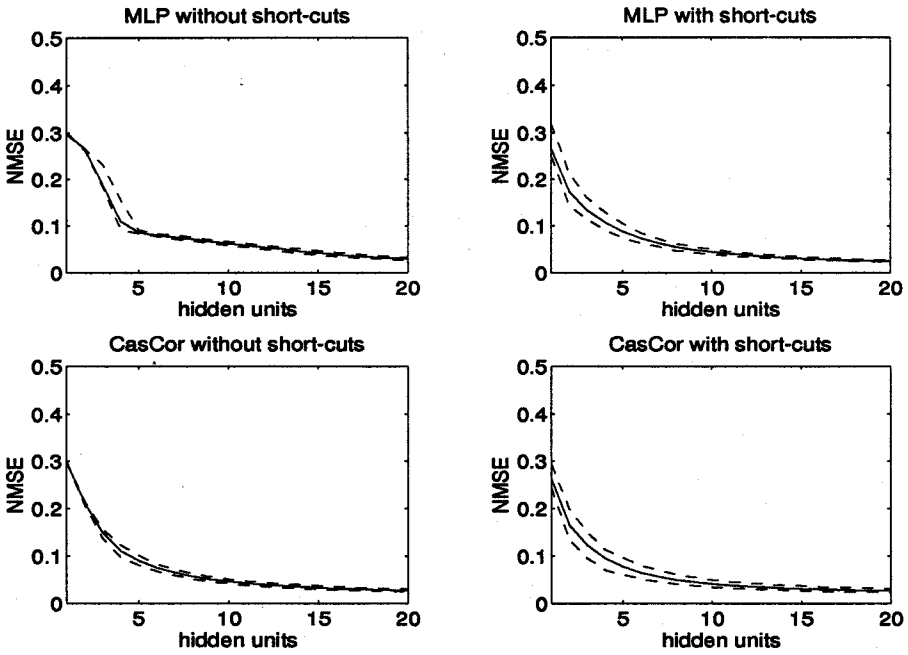

Fig. 5. Training curves for the time series problem. Solid lines are the average curves and dashed lines are upper and lower deviation curves, respectively.

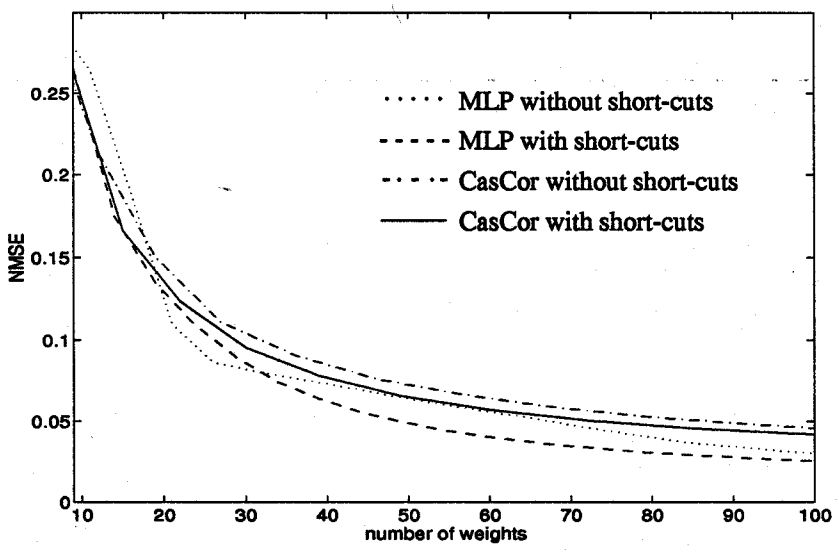

Fig. 6. Additional training curves for the time series modeling problem. Dotted line is the average curve for three-layer perceptron without short-cuts, dashed line is for three-layer perceptron with short-cuts, dashdot line is for Cascade-Correlation without short-cuts and solid line is for Cascade-Correlation with short-cuts.