On Weight Initialization in Constructive Backpropagation Learning

Mikko Lehtokangas
Tampere University of Technology
Signal Processing Laboratory
P.O.Box 553, FIN-33101 Tampere, Finland
Phone: +358-3-3653881, Fax: +358-3-3653095
email: mikkol@cs.tut.fi

ABSTRACT: Constructive backpropagation learning is a recently proposed training method that similarly constructs feedforward neural networks as the well known cascade-correlation learning. In recent studies the constructive backpropagation has been shown to yield several benefits compared to the cascade-correlation. However, in difficult classification type of problems the constructive backpropagation may have difficulties to converge. In this study weight initialization is considered for solving the convergence problems of the constructive backpropagation learning. Two weight initialization approaches are investigated. The first one is so called candidate training approach which has been proposed in the context of cascade-correlation learning. The other approach is the so called maximum covariance initialization. Experimental results demonstrate that with proper network weight initialization, the constructive backpropagation can successfully handle also difficult classification problems. Especially the maximum covariance initialization method was found to yield good cost/performance ratio.

KEYWORDS: constructive backpropagation learning, weight initialization, classification, neural networks

INTRODUCTION

Feedforward neural networks are powerful models for solving nonlinear mapping problems, Haykin (1999). Their training is usually done with gradient descent based optimization routines, Fletcher (1990). The training can be viewed as a nonlinear optimization problem in which the goal is to find a set of network weights that minimize a cost function. The cost function which is usually a function of the network mapping errors describes a surface in the weight space, often referred to as the error surface. Training algorithms can be viewed as methods for searching minimum of this surface. The complexity of the search is governed by the nature of the surface. For example, error surfaces for multilayer perceptron networks can have many flat regions where learning is slow, and long narrow "canyons" that are flat in one direction and steep in the other directions. Thus, in realistic cases, the large number of very flat and very steep parts of the surface make it difficult to search the surface efficiently using gradient based training routines. In addition, the cost function is characterized by a large number of local minima with values in the vicinity of the best or global minimum.

Because of the complexity of the search space, the main drawbacks of gradient techniques are that they are slow and unreliable in convergence. Major reasons for poor training performance are the problem of determining optimal steps (i.e. size and direction in the weight space in consecutive iterations), the problem of network size, and weight initialization. The improved training algorithms (more optimal steps) and optimal network size do not alone guarantee adequate convergence because of the initialization problem. When the initial weight values are poor the training speed is bound to get slower even if improved training algorithms and optimal network size are used. In the worst case the network may converge to a poor local optimum.

Recently the constructive backpropagation (CBP) learning method was proposed for training feedforward neural networks, Lehtokangas (to appear). Similar to the cascade-correlation (CC) learning, Fahlman and Lebiere (1990), the main idea in constructive backpropagation is to start with a minimal network (no hidden units), and then train and add new hidden units one by one to the network until desired performance is reached (or until we give up). The main benefits of this type of learning are the fast convergence and automatic network size determination. In addition, constructive backpropagation have several benefits compared to the cascade-correlation learning, Lehtokangas (to appear). First of all, constructive backpropagation uses only one cost function which simplifies the implementation. Secondly, since the cost

function used is the sum of squared errors, stochastic optimization routines can be used for weight adjustment as well as batch ones. Stochastic routines have been found practical and useful especially in problems with a large amount of data. Thirdly, constructive backpropagation can be extended with two important features. A batch of hidden units can be added to a network instead of only one unit at a time. This feature can improve performance substantially since multiple units trained together can reduce modelling error more than multiple units trained independently. Continuous automatic structure adaptation can also be performed instead of just growing the network once and fixing it permanently. This feature can be very helpful in problems where the underlying dynamics change gradually as a function of time.

In our recent experiments we have noticed that especially in difficult classification problems the constructive backpropagation learning suffers also from the weight initialization problem. As the training of a new batch of hidden units starts, its weights are initialized with small random values. Thus it is possible that the new batch converges to a poor solution with the result that a useless batch will be permanently installed to the network. To address this problem, in this study we investigate two weight initialization approaches to be used in the context of constructive backpropagation learning. The first is the so called candidate training approach which has been proposed in the context of cascade-correlation learning, Fahlman and Lebiere (1990). The drawback of this approach is that the computational load of training can increase significantly if the number of candidates is large. To provide computationally more efficient weight initialization scheme, we consider the maximum covariance method, Lehtokangas et al. (1996), as the second initialization approach. The feedforward architecture considered here is a network where batches of sigmoidal hidden units are added to a single hidden layer. This way we avoid the problems of large fan-in to hidden units and irregular network structure, Phatak and Koren (1994). Moreover, in Prechelt (1997) it was reported that in many problems placing hidden units in one hidden layer yields equal or better results than cascading the hidden units as proposed in Fahlman and Lebiere (1990).

CANDIDATE TRAINING INITIALIZATION

For the reader's convenience let us briefly recall the constructive backpropagation learning procedure as follows:

step1. *Initial configuration:* The network has no hidden units. Only bias weights and possible direct connections (i.e. short-cut connections) from inputs to outputs feed the output units. Train the weights of this initial configuration by minimizing the usual sum of squared errors (SSE).
step2. *Train new batch of hidden units:* Connect inputs to the hidden units in the new batch and connect their outputs to the output units. Adjust all the weights connected to the new batch of units (both input and output connections) by minimizing the SSE criterion.
step3. *Freeze new batch of hidden units:* Fix the weights connected to the units in the new batch permanently.
step4. *Test for convergence*: If the current number of batches yield an acceptable solution, then stop the training. Otherwise go back to step 2.

For exact details see Lehtokangas (to appear). Now, step 2 in the above procedure is the critical step considering the weight initialization. There, before gradient descent is applied for training the respective batch of units, the weights of the batch need to be initialized. The simplest initialization scheme is to use small random values like in standard multilayer perceptron network initialization. As our experimental results will demonstrate, this simple initialization may not be enough in e.g. complex classification problems. In cascade-correlation learning (which has similar steps as the above constructive backpropagation procedure) the weight initialization problem has been addressed by the use of candidate training, Fahlman and Lebiere (1990). There, several candidate units are first initialized with different small random numbers. Then all these candidates are trained for the same task, and the best candidate is finally chosen to be installed to the network. This same idea can be used also in constructive backpropagation. In above step 2, instead of initializing only one batch we initialize several batches with different small random numbers. Then all these batches are trained for the same task, and at the end the best batch is chosen to be installed to the network. Although candidate training is simple to implement it can be rather costly in terms of amount computation required. Obviously, training several candidate batches requires several times more computation compared to training only a single batch. To introduce computationally more viable approach we shall next consider the maximum covariance method.

MAXIMUM COVARIANCE INITIALIZATION

Maximum covariance (MC) method has been originally proposed for initialization of multilayer perceptron networks (MLP), Lehtokangas et al. (1996). The underlying idea of the MC-method is based on the stepwise regression which is

an old statistical technique for selecting the best regression equation, Draper and Smith (1981). In MLP initialization the MC-method has provided good cost/performance ratio in many difficult classification problems. Therefore, we believe that is can provide to be useful also in the context of constructive backpropagation learning. To incorporate the maximum covariance initialization method to the constructive backpropagation procedure described in the previous section, we must include the following initialization steps at the beginning of step 2 in the previous section:

Ini1. Create R candidate hidden units ($R>>r$; r denotes the desired number of hidden units in the new batch) by initializing the weights feeding them with random values.

Ini2. Do not connect the candidate units to the output unit yet. The only connections feeding the output unit at this time are the frozen connections from the previously added batches.

Ini3. Calculate the covariance for each of the candidate unit from the equation

$$C_k = \frac{1}{n}\sum_{l=1}^{n} (y_{k,l} - \bar{y}_k)\ (\varepsilon_l - \bar{\varepsilon})\ , \qquad (1)$$

k=1,...,R. In above $y_{k,l}$ is the output of the k:th candidate hidden unit for the l:th example. The parameter $\bar{y}_k$ is the mean of the k:th hidden unit outputs, ε_l is the output error at the network output and $\bar{\varepsilon}$ is the mean of the output errors.

Ini4. Find the maximum absolute covariance $|C_k|$ and connect the corresponding hidden unit to the output unit. Decrement the number of candidate hidden units R by one.

Ini5. Optimize currently existing weights that connect candidate units to the output unit with linear regression. Note that the number of these weights is increased by one every time a new candidate unit is connected to the output unit, and due to the optimization the output error changes each time. Note also, that the connections from the previously added batches are not adjusted since they are frozen.

Ini6. If r candidate units (that form the new batch) have been connected to the output unit then continue with the actual batch training. Otherwise go back to Ini3.

The above maximum covariance method can be seen as a deterministic method for choosing the best initializations (according to covariance criterion) from a large set of random initializations. Compared to the candidate training initialization in the previous section, the maximum covariance initialization is obviously more computationally economical since only one batch need to be actually trained. This can be very important feature in problems where fast learning is required.

EXPERIMENTS

In this section the performances of the above described candidate training and maximum covariance initialization methods are empirically investigated in the context of constructive backpropagation learning. The performance of standard cascade-correlation learning is also presented for comparison purposes. We have used the channel equalization problem described in Kantsila et al. (to appear) as a benchmark problem. There the goal is to equalize a burst of bits transmitted through a fixed communication channel having 20db signal-to-noise ratio. The optimization procedure we used for actual training was the RPROP algorithm, Riedmiller and Braun (1993). In constructive backpropagation batches of one or two hidden units were used. Also, in candidate initialization we used 5 candidates, and in maximum covariance initialization we used R=100. Each of the simulations were repeated ten times.

The results for the cascade-correlation learning with no candidates and candidate initialization are depicted in Fig. 1. Figs. 2-3 present results for the constructive backpropagation with no candidates, candidate initialization and MC-initialization, respectively. Obviously, in cascade-correlation learning the usage of candidates has significant effects on the results. The effective number of hidden units required reduces from 8 to 5. Considering constructive backpropagation learning, using batches of one hidden unit with no candidates (Fig. 2a) and with candidates (Fig. 3a), there are clearly problems in convergence. The usage of batches of two hidden units (Figs. 2b and 3b) drastically improve the situation. This is yet another demonstration about the fact that multiple units trained together can reduce modelling error more than multiple units trained independently. However, the best results are obtained by the use of MC-initialization (Fig. 4).

Considering computational costs, the best constructive backpropagation scheme is about 5 times faster than the best cascade-correlation scheme. In addition, maximum covariance initialization is about 4-6 times faster than the candidate initialization. As a result, these experiments demonstrate that the maximum covariance initialization can provide very good cost/performance ratio in the constructive backpropagation learning.

CONCLUSIONS

Weight initialization in the constructive backpropagation learning was considered. First we described the problem of weight initialization, and mentioned about our recent findings that in difficult classification problems the constructive backpropagation learning may have difficulties to converge. Then we described two approaches for solving the weight initialization problem. These were the candidate training and maximum covariance initialization. After that, we empirically investigated the performances of different weight initializations by the use of channel equalization problem. Performance of cascade-correlation learning was also presented for comparison purposes. The obtained results demonstrated that without proper initialization the constructive backpropagation can fail to converge. However, with proper initialization the constructive backpropagation can yield much better results compared to the cascade-correlation learning. In terms of cost/performance ratio, especially the maximum covariance initialization was found to provide the greatest potential for the initialization problem.

ACKNOWLEDGEMENTS

This work has been supported by the Academy of Finland.

REFERENCES

Draper N. and Smith H., 1981, Applied regression analysis, 2nd ed., John Wiley & Sons Inc.
Fahlman S. and Lebiere C., 1990, "The cascade-correlation learning architecture," In D. Touretzky (Ed.), Advances in Neural Information Processing Systems 2, San Mateo, CA, Morgan Kaufman, pp. 524-532.
Fletcher R., 1990, Practical methods of optimization, 2nd ed., Wiley, Chichester.
Haykin S., 1999, Neural networks: a comprehensive foundation, 2nd ed., Prentice Hall, New Jersey.
Kantsila A., Lehtokangas M. and Saarinen J., to appear, “Burst adaptive equalization of binary data,” Journal of Intelligent Systems.
Lehtokangas M., to appear, “Modelling with constructive backpropagation,” Neural Networks.
Lehtokangas M., Korpisaari P. and Kaski K., 1996, "Maximum covariance method for weight initialization of multilayer perceptron network," Proceedings of the European Symposium on Artificial Neural Networks, ESANN'96, pp. 243-248.
Phatak D. and Koren I., 1994, “Connectivity and performance tradeoffs in the cascade correlation learning architecture,” IEEE Transactions on Neural Networks, vol. 5, no. 6, pp. 930-935.
Prechelt L., 1997, “Investigation of the CasCor family of learning algorithms,” Neural Networks, vol. 10, no. 5, pp. 885-896.
Riedmiller M. and Braun H., 1993, “A direct adaptive method for faster backpropagation learning: the RPROP algorithm,” *Proceedings of IEEE International Conference on Neural Networks*, pp. 586-591.

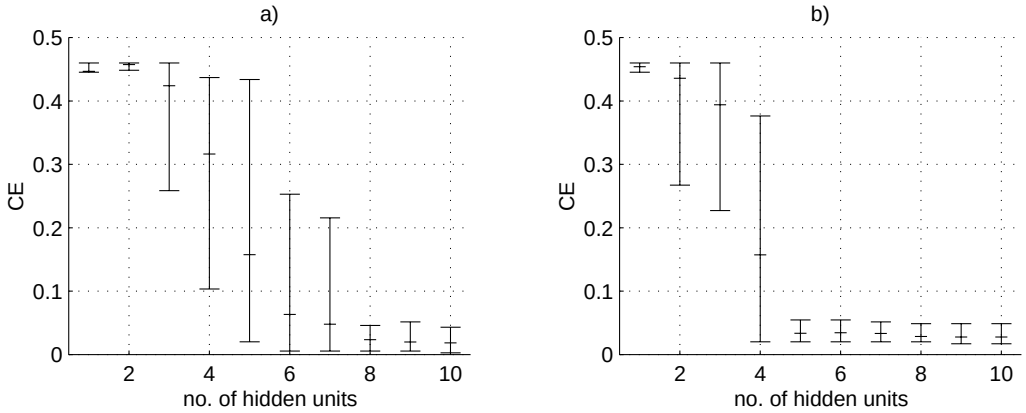

Figure 1: Classification error as a function of number of hidden units for the *independent test set* by the use of cascade-correlation learning; a) no candidates, and b) candidate initialization. The smaller horizontal line in the middle is located at the average of the repetitions, while the whiskers show the total range of values.

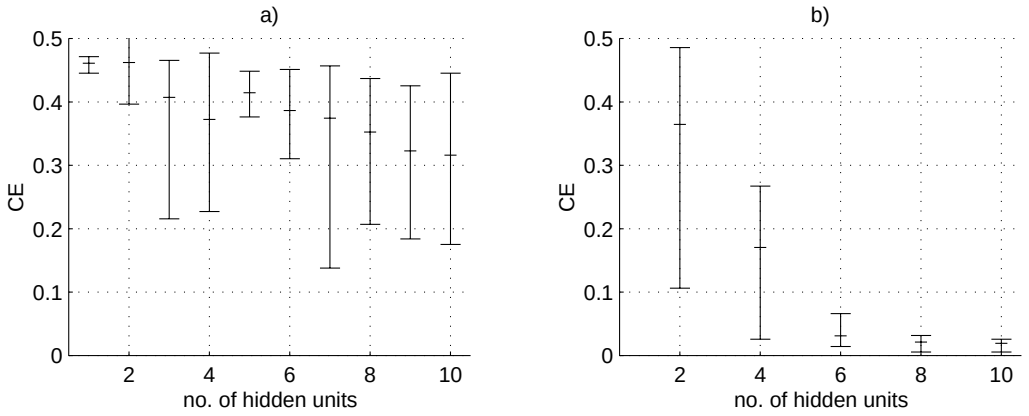

Figure 2: Classification error as a function of number of hidden units for the *independent test set* by the use of constructive backpropagation learning with no candidates; a) batches of one hidden unit, and b) batches of two hidden units. The smaller horizontal line in the middle is located at the average of the repetitions, while the whiskers show the total range of values.

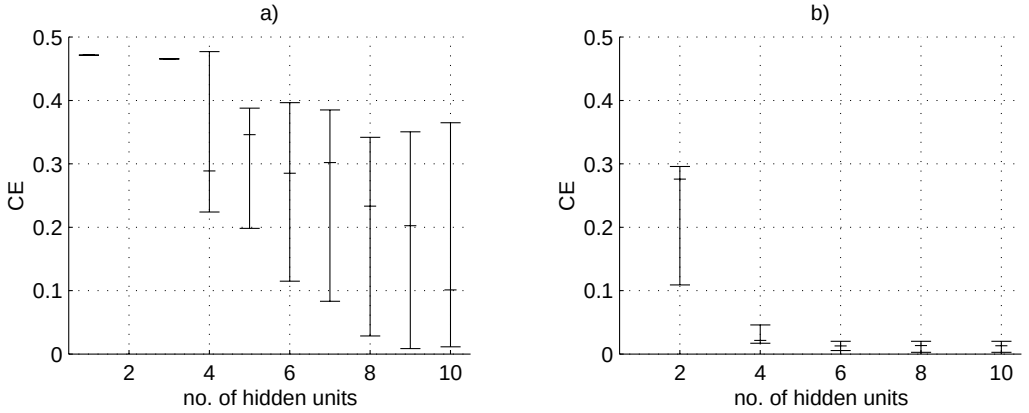

Figure 3: Classification error as a function of number of hidden units for the *independent test set* by the use of constructive backpropagation learning with candidate initialization; a) batches of one hidden unit, and b) batches of two hidden units. The smaller horizontal line in the middle is located at the average of the repetitions, while the whiskers show the total range of values.

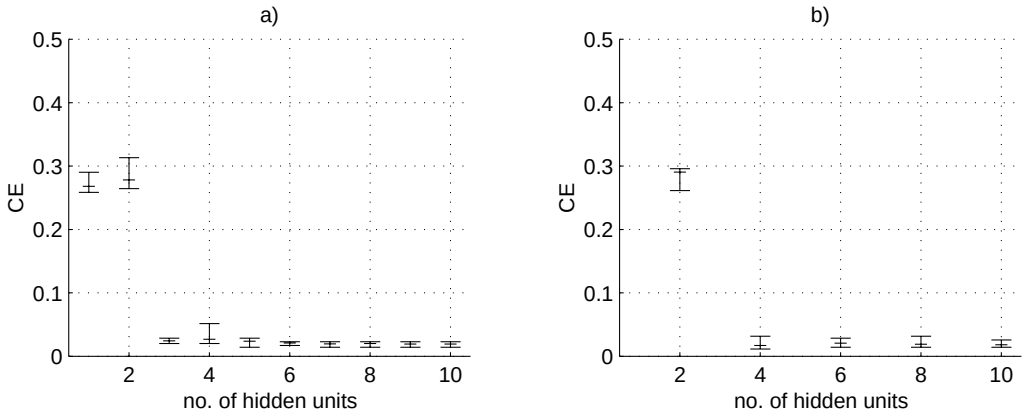

Figure 4: Classification error as a function of number of hidden units for the *independent test set* by the use of constructive backpropagation learning with maximum covariance initialization; a) batches of one hidden unit, and b) batches of two hidden units. The smaller horizontal line in the middle is located at the average of the repetitions, while the whiskers show the total range of values.