ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 243-248

# Maximum Covariance Method for Weight Initialization of Multilayer Perceptron Networks

Mikko Lehtokangas[1], Petri Korpisaari[2] and Kimmo Kaski[2]

[1]Nokia Research Center
P.O. Box 100, FIN-33721 Tampere, Finland
lehtokan@research.nokia.com

[2]Tampere University of Technology, Electronics Laboratory
P.O.Box 692, FIN-33101 Tampere, Finland
{petrik, kaski}@ee.tut.fi

**Abstract.** The training of multilayer perceptron network starts by giving initial values to the weights. Commonly small random values are used for weight initialization. Then their adjustment is carried out by using some gradient descent based optimization routine such as backpropagation. If the initial weight values happen to be poor then it may take a long time to obtain adequate convergence, or in the worst case the network may get stuck to a poor local minimum. To improve the convergence in the training phase we introduce a maximum covariance method to initialize the weights. The simulation results show that the maximum covariance method is relatively fast to compute and it improves the convergence significantly over the random initialization.

## 1 Introduction

The multilayer perceptron (MLP) network is one of the most well known and widely used neural network model. Nowadays there exists numerous learning algorithms which can be used for training the network weights [1]. Regardless of many sophisticated learning algorithms the initial values given to the weights can affect critically the learning behaviour. Therefore, several initialization methods have been studied [2,3].

In this study a maximum covariance (MC) method is proposed for the weights initialization. This method can be divided to three phases. First a large number of candidate hidden units is created by initializing their weights with random values. Then the desired number of hidden units is selected amongst the candidates by using the MC criterion. Finally, the weights feeding the output units are calculated with linear regression. After the MC initialization the network is trained with some optimization routine. We chose to use the resilient backpropagation (RPROP) which has been shown to be an efficient adaptive training algorithm [4,5]. The performance of the MC initialization method is tested using two benchmark problems, and a comparison with random initialization is presented.

## 2 Maximum Covariance Method

The proposed MC initialization method can be used to initialize MLPs with one hidden layer as depicted in Fig. 1. Although in this study we use single-output network, the MC method can be directly expanded to multi-output case. The network we are considering can be written as

$$y = v_0 + \sum_{j=1}^{q} v_j \tanh\left( w_{0j} + \sum_{i=1}^{r} w_{ij} x_i \right). \tag{1}$$

The number of inputs is $r$, number of hidden units is $q$, weights are denoted with $v_j$ and $w_{ij}$ (including the biases $v_0$ and $w_{0j}$), and the activation function in the hidden units is *hyperbolic tangent* (*tanh*) function. It is noted that the output unit is linear.

The RPROP training method, which is used after the initialization, can be expressed with the following equations

$$\theta(t+1) = \theta(t) + \Delta\theta(t) \tag{2}$$

$$\Delta\theta(t) = \begin{cases} -\Delta(t) & \text{, if } \partial E^t / \partial\theta > 0 \\ +\Delta(t) & \text{, if } \partial E^t / \partial\theta < 0 \\ 0 & \text{, else} \end{cases} \tag{3}$$

$$\Delta(t) = \begin{cases} \eta^+ \Delta(t-1) & \text{, if } \left(\partial E^{t-1}/\partial\theta\right)\left(\partial E^t/\partial\theta\right) > 0 \\ \eta^- \Delta(t-1) & \text{, if } \left(\partial E^{t-1}/\partial\theta\right)\left(\partial E^t/\partial\theta\right) < 0 \\ \Delta(t-1) & \text{, else} \end{cases} \tag{4}$$

Parameter $\theta$ denotes a weight ($v_j$ or $w_{ij}$) and $E$ is the cost function i.e. the sum squared error. The RPROP method includes several parameters for which we used the following values: decrease factor $\eta^-$=0.5, increase factor $\eta^+$=1.2, initial update value $\Delta_0$=$10^{-5}$, maximum update value $\Delta_{max}$=1 and minimum update value $\Delta_{min}$=$10^{-10}$. More details about the RPROP method can be found in Refs [4,5].

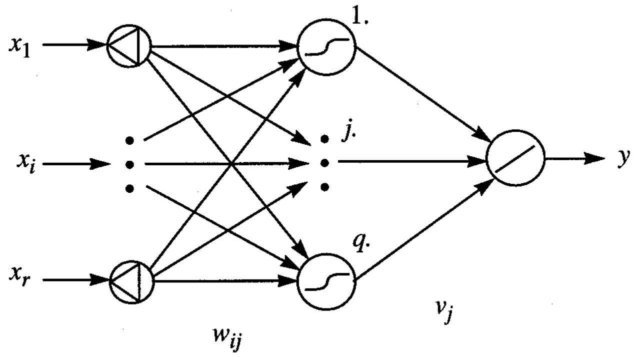

**Figure 1.** MLP with one hidden layer and single output.

The maximum covariance initialization algorithm can be described by the following steps:

1. Choose the desired number of hidden units $q$ by using some appropriate model selection method. Different model selection methods have been represented for example in Ref [6].
2. Create $M$ candidate hidden units ($M>>q$) by initializing their weights $w_{ij}$ with random values. We used $M=10q$ and the candidate units were initialized with uniformly distributed random numbers from the interval [-4;4].
3. Do not connect the candidate units to the output unit yet. Only parameter feeding the output unit at this time is the bias weight $v_0$. Set the bias weight value to be such that the network output is the mean of the desired output sequence.
4. Calculate the covariance for each of the candidate unit from equation

$$C_j = \frac{1}{n} \sum_{p=1}^{n} (o_{j,p} - \bar{o}_j)(e_p - \bar{e}) \qquad , j = 1, \ldots, M \tag{5}$$

   in which $o_{j,p}$ is the output of the $j$th hidden unit for $p$th pattern. Parameter $\bar{o}_j$ is the mean of the $j$th hidden unit's outputs, $e_p$ is the output error at the network output and $\bar{e}$ is the mean of the output errors.
5. Find the maximum absolute covariance $|C_j|$ and connect the corresponding hidden unit to the output unit. Set $M=M-1$.
6. Optimize the currently existing output weights $v_j$ 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.
7. If $q$ candidate units have been connected to the output unit then quit the initialization phase; otherwise repeat the steps 3-5 for the remaining candidate units.

The idea behind the MC initialization method is to one by one select those hidden units amongst the candidates which have the maximum absolute covariance with the current output error. In this way those candidate hidden units are selected which can efficiently 'cancel' the output error.

## 3 Experiments

Next we present two examples of the usefulness of the proposed MC initialization scheme. The first benchmark is the 4x4 chessboard problem, which is depicted in Fig. 2a. In an *nxn* chessboard problem the network has two inputs which are the coordinates of the squares in the *nxn* sized chessboard. For white squares the output is 'off' and for black squares the output is 'on'. The other problem is the well known two-spirals problem which has been studied for example in Ref [7]. There are two inputs which correspond the X-Y coordinates. Half of the input patterns produce 'on' and another half 'off' to the output. The training points are arranged in two interlocking spirals as shown in Fig. 2b.

The training performance is studied by using the misclassification percentage metric which indicates the proportion of incorrectly classified output items. The 40-20-40 scheme is used which means that if the total range of the desired outputs is 0.0 to 1.0 then any value below 0.4 is considered to be a zero ('off') and any value above 0.6 is considered to be a one ('on'). Values between 0.4 and 0.6 are automatically classified as incorrect. The training was repeated 100 times for each scheme. In the random initialization scheme uniformly distributed random numbers from the interval [-0.5;0.5] were used. For the 4x4 chessboard problem we used 6 hidden units and for two-spirals problem 42 hidden units in the network. In the two-spirals problem the MLP with 42 hidden units has about the same number of parameters as the 15 unit Cascade-Correlation network which have been shown to give good results in Ref [7].

The average learning curves for the benchmark problems are shown in Figs. 3 and 4. Evidently, with maximum covariance initialization the convergence is significantly better compared to the case where random initialization is used; much lower misclassification value is obtained on average. From Table 1 it can be seen that the computational cost of the MC method is acceptably low. For example, in the chess problem the MC initialization corresponds 20 epochs of training with RPROP. Certainly it is profitable to do the initialization because it improves the convergence considerably. The same conclusion applies for the two-spiral problem. More simulations of the MC method and comparisions with other initialization methods can be found in Ref [6].

**Table 1.** Computational costs of the initialization methods.

| Data | Method | $n$ | $M$ | $q$ | Cost in epochs |
|---|---|---|---|---|---|
| chess | random | 16 | - | 6 | ~ 0 epoch |
| chess | MC | 16 | 60 | 6 | 20 epoch |
| spiral | random | 194 | - | 42 | ~ 0 epoch |
| spiral | MC | 194 | 420 | 42 | 180 epoch |

## 4 Conclusions

The initialization of the multilayer perceptron network with maximum covariance method was studied. The experimental results showed that with this initialization the training convergence can be improved significantly. On the other hand, the maximum covariance method was found to have acceptably low computational cost. These findings indicate that the proposed method can be useful for initializing multilayer perceptron network, and inspire further studies.

## References

[1] M. Pfister and R. Rojas, "Speeding-up backpropagation - a comparison of orthogonal techniques," Proceedings of International Joint Conference on Neural Networks, IJCNN'93, Nagoya, Japan, vol. 1, pp. 517-523, 1993.

[2] G. Drago and S. Ridella, "Statistically controlled activation weight initialization (SCAWI)," IEEE Transactions on Neural Networks, vol. 3, no. 4, pp. 627-631, 1992.

[3] L. Wessels and E. Barnard, "Avoiding false local minima by proper initialization of connections," IEEE Transactions on Neural Networks, vol. 3, no. 6, pp. 899-905, 1992.

[4] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the RPROP algorithm," Proceedings of IEEE International Conference on Neural Networks, San Francisco, CA, 1993.

[5] M. Riedmiller, "Advanced supervised learning in multilayer perceptrons - from backpropagation to adaptive learning algorithms," International Journal of Computer Standards and Interfaces, Special Issue on Neural Networks, vol. 5, 1994.

[6] M.Lehtokangas, "Modeling with Layered Feedforward Neural Networks," Doctoral Thesis, Tampere University of Technology, Electronics Laboratory, Finland, September 1995.

[7] S. Fahlman and C. Lebiere, "The Cascade-Correlation learning architecture," Research report CMU-CS-90-100, Carnegie Mellon University, 1991.

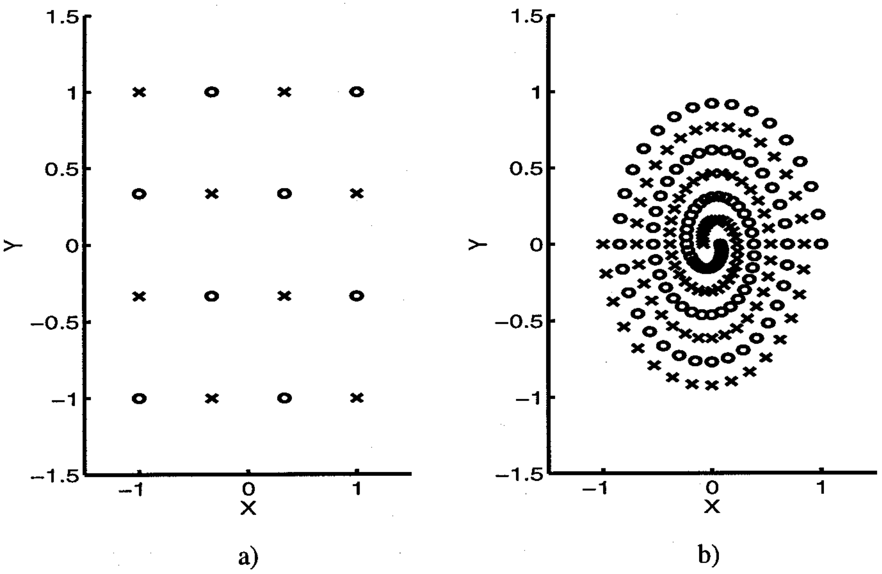

**Figure 2.** Circles represent the 'off' and crosses the 'on' values. a) 4x4 sized chessboard problem. b) Two-spirals problem.

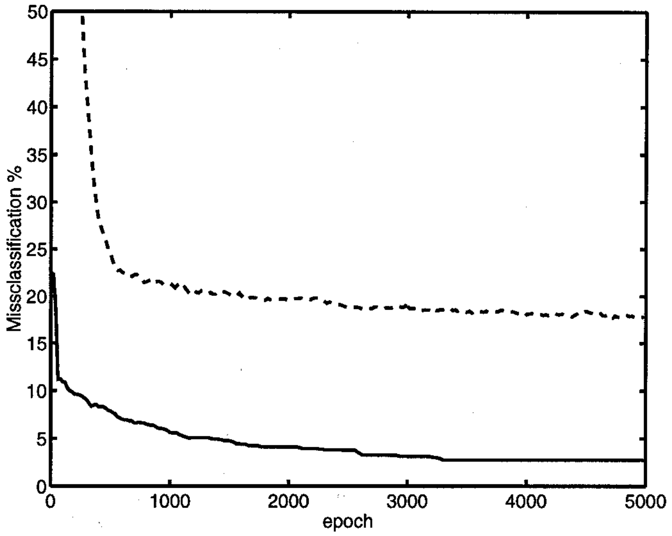

**Figure 3.** Average learning curves for the chess problem. Dashed line was obtained with random initialization and solid line with maximum covariance initialization.

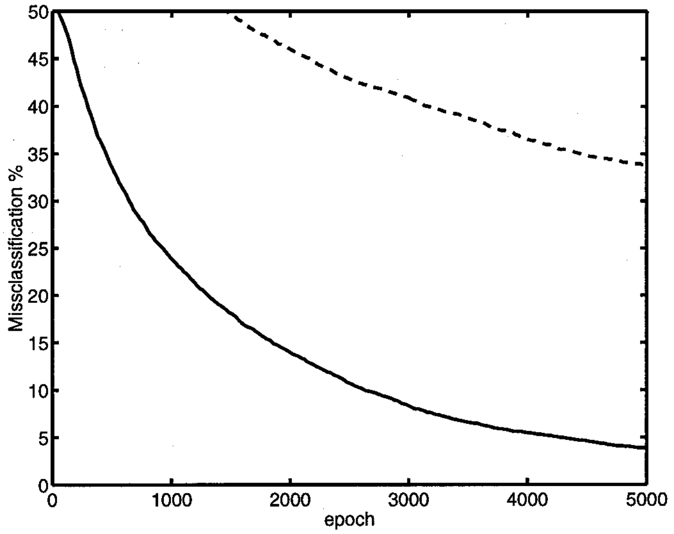

**Figure 4.** Average learning curves for the spiral problem. Dashed line was obtained with random initialization and solid line with maximum covariance initialization.