IDIAP RESEARCH REPORT

IDIAP

Martigny - Valais - Suisse

Optimal Setting of Weights, Learning Rate, and Gain

G. Thimm and E. Fiesler
Email: Thimm@idiap.ch, EFiesler@idiap.ch

IDIAP–RR 97-04

April 1997

Dalle Molle Institute for Perceptive Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 − 27 − 721 77 11
fax +41 − 27 − 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

IDIAP Research Report 97-04

Optimal Setting of Weights, Learning Rate, and Gain

G. Thimm and E. Fiesler
Email: Thimm@idiap.ch, EFiesler@idiap.ch

April 1997

he optimal setting of the initial weights, learning rate, and gain of the activation function, which are key parameters of a neural network, influencing training time and generalization performance, are investigated by means of a large number of experiments using ten benchmarks using high order perceptrons.

The results are used to illustrate the influence of these key parameters on the training time and generalization performance and permit general conclusions to be drawn on the behavior of high order perceptrons, some of which can be extended to the behavior of multilayer perceptrons. Furthermore, optimal values for the learning rate and the gain of the activation function are found and compared to those recommended by existing heuristics.

Keywords: high order perceptron, learning rate, initial weights, gain, generalization, training time

1 Introduction

The time to train neural networks with the backpropagation learning rule depends much on the initial values of the weights and biases, the learning rate(s), the type of sigmoidal function(s), the network topology, and on learning rule parameters like the momentum term. The optimal values for these parameters are *a priori* unknown because they depend on the training data set used. In practice it is infeasible to perform a global search for obtaining the optimal values of these parameters in this multidimensional space.

However, there are many possible ways to optimize the training time, the generalization performance, or other properties of neural networks. Specialized techniques modify the topology of neural networks [5] or the learning rule [9]. Others try to find optimal values for the initial weights, learning rate, momentum term, and so on. The sophisticated methods, which for example modify the network topology, usually assume that the more basic parameters, like the learning rate, the initial weights, and so on, are (almost) optimal, or at least assume such values to be initially optimal. It is therefore the intension of this study to gain more insight on the influence of these parameters on the neural network behavior. Furthermore, it is attempted to find a good approximation of the optimal initial value of some basic parameters, where "good" means that the behavior of the network is close to that with optimal values.

2 Optimization of Initial Weights and Learning Rate, and Gain

The study of weight initialization methods, of which an overview is given in [15], shows that researchers mainly try to optimize learning speed and generalization performance of neural networks initialized with random weights in two ways. Firstly, by using different distributions for the weights. Secondly, by estimating a good initial weight variance[1] based for example on:

- the steepness of the sigmoidal function,
- the number of connections feeding into a neuron (= fan-in of a neuron),
- (analysis of) the data set on which the network will be trained,
- the number of connections in the network, and
- constants that emerged from experiments (for example: a constant multiplied with the maximum of the first derivative of the activation function determines the initial weight range).

[1] For a uniform distribution over the interval $[-u, u]$ the variance σ^2 equals $\frac{u^2}{3}$.

As a previous study showed that the shape of the weight distribution has almost no influence on the training time or the generalization performance of a trained neural network [15], it is chosen to be a uniform distribution in this publication.

Another parameter for the optimization of the training that can be neglected is either the learning rate or the gain of the activation functions as a theorem proven in [16] shows that two neural networks N and N' that differ only in their learning rates η and η', their weights $\mathbf{w}'$ and $\mathbf{w}$, and the gain of the activation function β and β' behave in the same way if

$$\frac{\beta}{\beta'} = \sqrt{\frac{\eta'}{\eta}} = \frac{\mathbf{w}'}{\mathbf{w}}. \tag{1}$$

This relation permits to neglect a variation of one of the three parameters during a search for an optimal combination. Hence, the gain will be chosen to be one in the following sections[2].

On the other hand, no rule for the optimal selection of the learning rate exist: the learning rate is usually chosen by some *rule of thumb* and changed until the network appears to converge. It is however not justifiable on the basis of equation 1 to hold two parameters fixed and to search only for an optimal value for the remaining one. The experiments will show that this is indeed insufficient.

In order to circumvent the problem of finding the optimal learning rate, an adaptive approach is used where the learning rate is automatically modified and optimized during the training process. It should be remarked that these methods are not applied here, as they aim at decreasing the training time independent of the initial learning rate [9, 13]. This means that they can always be used as "add-on". Furthermore, these methods often introduce new parameters, which reduces the user-friendliness. Another drawback is that most of these methods require off-line learning, which is commonly assumed to be slower than on-line training [7, page 119]. However, some adaptive learning rates methods can compensate to a large extent for a bad initial value [9].

3 The Choice of the Activation Function

The convergence of the training process, the generalization performance of the network, *etc.* depends, besides the learning parameters and the topology, also on the activation function. It is, for example, easily verifiable that for classification tasks (with Boolean target values), the linear activation function leads to bad results: the weighted sum of the connection outputs has to be almost exactly 0 and 1 (respectively -1 and 1), which is very restrictive: the training does not only aim at separating the classes by adjusting a hyper-surface[3] between them but maximizing the distances between this surface and the data. Better performance is therefore often obtained if a logistic activation function is used in the output neurons, which prevents the network to "overshoot" the correct output values. Furthermore, high order perceptrons appear to learn classification problems faster if the hyperbolic tangent is used as activation function. As compared to the case when the standard sigmoid is used, the weight changes for incorrectly classified patterns with a network output **false** are more important.

In this publication, Boolean values **true** and **false** are represented by 0.9 and -0.9, in order to prevent the absolute weight values from growing without bound. Alternatively, a modified logistic in the range $(0, 1.1)$, or hyperbolic tangent function in the range $(-1.1, 1.1)$ could be used.

For data sets with continuous valued targets, the choice of the sigmoidal function is more difficult, and arguments similar to those given for the Boolean case are not conclusive. The experiments for this type of data sets are therefore performed with both the linear and logistic activation function.

[2] Another application of equation 1 are optical hardware implementations of neural networks which impose a certain gain on the activation function, and therefore require an adaptation of the other two learning parameters [12].

[3] A n-dimensional surface(s) represented by input vectors for which the output of the network is zero, and on either side of the surface, the network output has a different sign.

3.1 Experiments with Weight Variance and Learning Rate

In order to relate the initial weight variance, learning rate, and training time to the generalization performance, as well as to evaluate the weight initialization techniques and to determine the best learning rate, the following scheme is used:

1. The optimal learning rate and weight initialization variance for fast convergence are globally searched for several data sets and three different activation functions with a fixed gain. These functions are the hyperbolic tangent, the logistic, and the (linear) identity function. A search for an optimal gain is not required, as any network can be "normalized" to have only activation functions with a gain equal to one (compare equation 1).

2. Similarly, the optimal learning rate and weight initialization variance for good generalization performance are searched for.

3. The outcome of these experiments is used to estimate the efficiency of some heuristics for the estimation of the optimal weight range (compare [15] and table 2).

The search for an optimal combination of learning rate and weight initialization variance can theoretically be done by a *line-search* algorithm, assuming that both the average training time and the generalization performance, form functions with a smooth surface: the gain is kept to a standard value of one, and either the initial weight range or the learning rate is varied until an optimum for both values is found. However, due to the randomness of the initial parameters, the results of a simulation are subject to statistical fluctuation. Observable effects of this fluctuation are, among others, variations of the convergence time and generalization performance of neural networks for different sets of initial weights, even if they are drawn from the same distribution. Consequently, first a line-search algorithm is used to get close to the optimal combination of the initial weight variance and learning rate. Then, the surrounding of this found combination is then searched for the optimal combination of learning rate and weight variance.

During the experiments, the networks are considered to have converged if the criteria of table 1 were met. The Digits data set is a subset of the NIST 3 data base [6], whereas the others are available from [10]. Their details are discussed in [15].

Data set	precision on training set
Solar	MSE smaller than 0.06
CES	MSE smaller than 0.1
Monk 1-3	100% correctly classified
Auto-mpg	MSE smaller than 0.06
Glass	MSE smaller than 0.03
Servo	MSE smaller than 0.07
Wine	100% correctly classified
Digits	99% correctly classified

Table 1: The convergence conditions for the experiments concerning the optimal choice of training parameters.

4 First results

To give a typical example of the behavior of the required training time as a function of initial weight variance and learning rate, a series of experiments using the Solar data set is discussed here in detail. The outcome of these experiments is shown in figure 1, where the training time in number of iterations is displayed as a function of the learning rate and the initial weight variance. The contour plot beneath the graph shows its channel-like shape with an outlet towards where the weight variance is zero. It can be seen that for a constant learning rate the convergence time remains almost constant for weight variances in the interval $[0.0, 0.1]$. If the learning rate is well-chosen, and the weight variance is optimal, then the high order perceptrons always converge in a near-optimal number of training cycles. The overall shape of the plot in figure 1 is common to all the experiments performed during this study, only the location and the width of the channel vary with the data set, the order of the network, as well as with other parameters.

Interestingly, the optimal learning rate for high order perceptrons is sometimes, as for this example, well above 1.0. In stark contrast to this observation is the recommendation of using a learning rate below 1.0 for a "standard" setting[4] of the other training parameters.

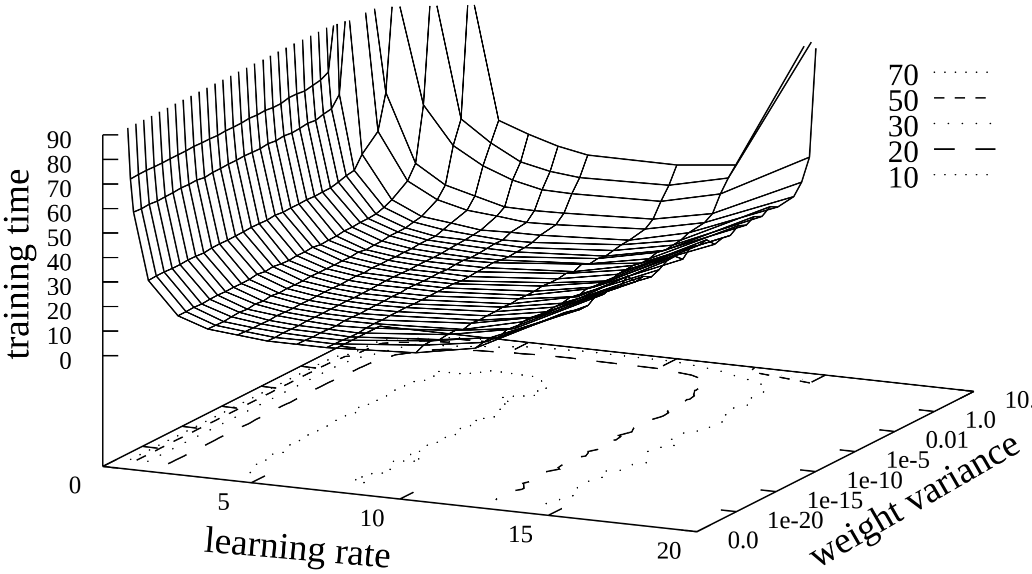

Figure 1: The training time of a high order perceptron as a function of weight variance and learning rate for the Solar data set.

These results are in contrast with the behavior of multilayer perceptrons. In figure 2 it can be seen that the training time of the multilayer perceptron as a function of the weight variance and learning rate has a bowl-like shape if trained on the Solar data set. The overall shape of this graph is probably representative for most multilayer perceptrons and data sets as it was observed during all experiments performed for this study (only the location of the minimum and the width of the bowl changed). Also, multilayer perceptrons usually fail to converge for weight variances equal to zero, and their training becomes slow when the initial weights become very small, as already stated by S. E. Fahlman [4]. In figure 2 it is shown that, similar to high order perceptrons, the optimal learning rate for multilayer perceptrons can also be bigger than one.

The average generalization performance of high order perceptrons, that have the same topology and are trained on the same data set, as a function of the learning rate and initial weight variance is displayed in figure 3. It can be seen that, similar to the training time, the generalization error is almost constant if the initial weight variance is below a certain value and the learning rate is unchanged. Furthermore, the generalization error increases for values above this limit; only a few exceptions to this behavior were encountered among 27 series of experiments. For a constant weight variance below this limit, the generalization performance improves (the error decreases) with a decreasing learning rate - just up to the point where the high order perceptrons cease to learn, that is, they do not converge in a certain number of iterations. This point is symbolized by the gray bar in figure 3.

[4] A "standard" setting is based on a gain of 1.0. Any other learning rate could be used if the gain is not defined [16].

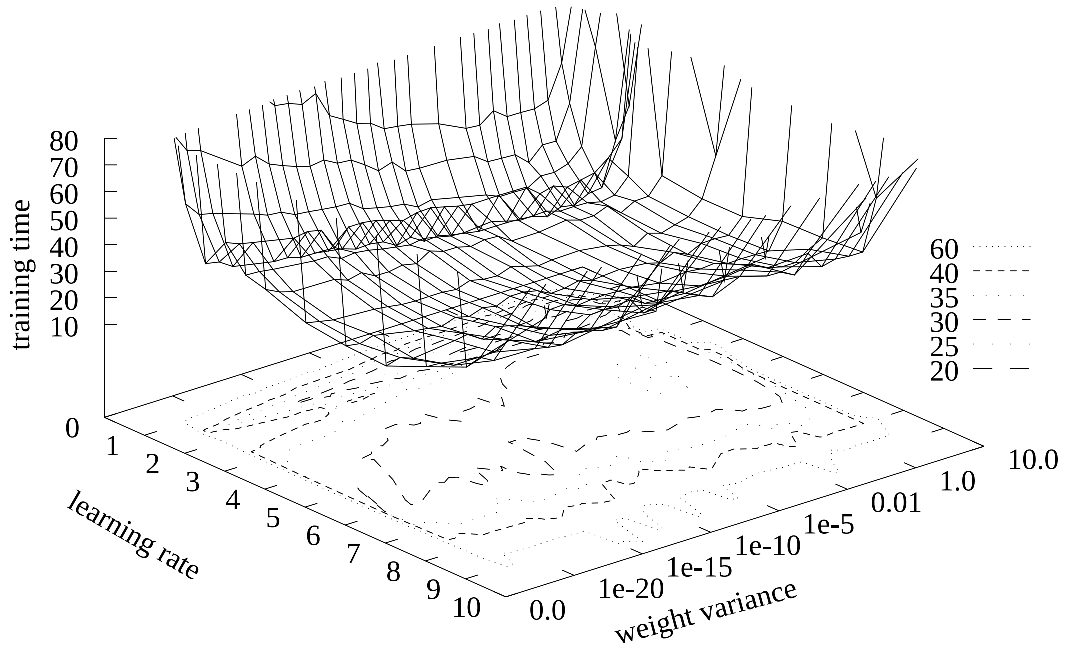

Figure 2: The training time of a multilayer perceptron as a function of weight variance and learning rate for the Solar data set.

Multilayer perceptrons behave similarly, as shown in figure 4, as confirmed by experiments performed with the Solar, Wine, Glass and Servo data sets. The most important difference with high order perceptrons is that the networks do not or only very slowly converge for weight variances close to zero. Such variances should therefore not be used for multilayer perceptrons.

However, the average over many simulations, which figure 3 displays, is somewhat misleading: the minimal error observed for all pairs of learning rate and initial weight variance for which the high order perceptrons converge is almost constant. Only the upper limit of the interval, in which errors can be observed, depends on the learning rate and weight variance. In other words, the minimal error is constant, whereas its maximum varies, as shown in figure 5, where the lower and the upper graph are the minimal, respectively the maximal, error as a function of learning rate and weight variance.

In contrast to the behavior of the learning time as described above, the behavior of generalization performance is less uniform and decreases for some data sets before the network ceases to learn due to too small a learning rate (giving a similar graph as for the learning rate in figure 1). A variation of this behavior is shown by a network with a logistic activation function trained on the CES data set: the generalization performance decreases together with a decreasing learning rate. More precisely, the distance between the minimal and maximal generalization performance, as displayed in figure 6, is almost constant over the whole range of learning rates and weight variances. Other variations of this behavior can be expected.

5 Optimizing Learning Speed

Table 3 shows an overview of the approximately 800,000 simulations performed with high order perceptrons. It lists the combinations of initial weight variances and learning rates for which the convergence time is (near) optimal for the different activation functions and network orders. The notation $a = 1.04 \pm 8$ means that the 95% confidence interval is (smaller than) the interval given by the limits obtained by adding 8 to, respectively subtracting 8 from, the last digit of 1.04. In other words, a lies in the interval $[0.96, 1.12]$.

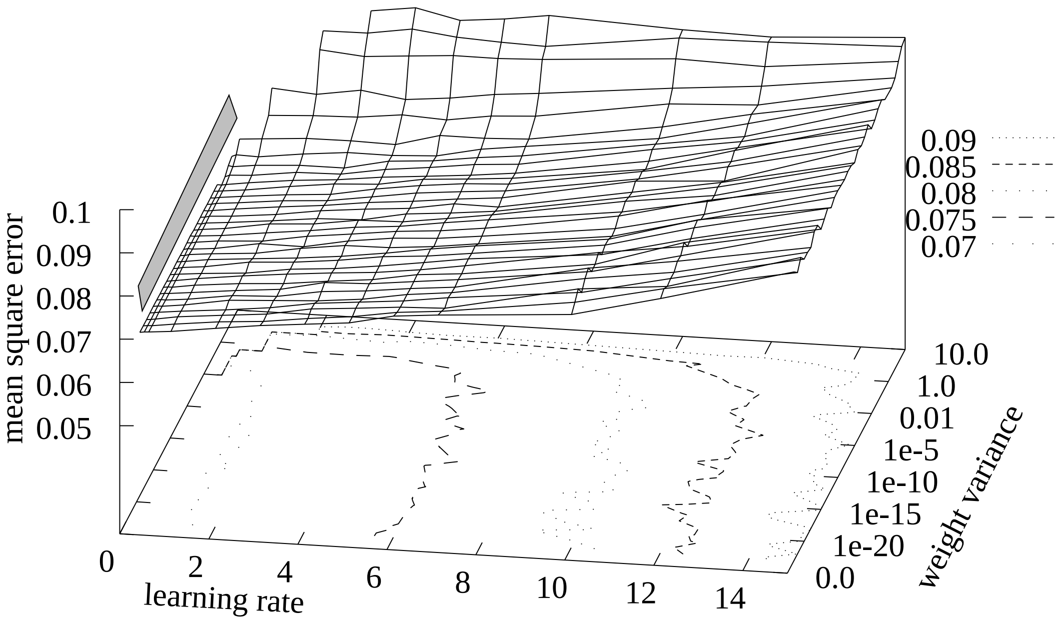

Figure 3: Generalization performance of a high order perceptron as a function of weight variance and learning rate for the Solar data set.

Globally, for fixed learning rates not far from the optimum, a weight variance exists below which the network converges in almost the same number of iterations. This includes zero weights. Above a certain weight variance, the convergence time increases very fast.

When comparing the results of other methods for determining the ‘optimal’ weight variances from table 2 with the results in table 3 or table 4, one observes that these methods rarely give a good estimation for the upper limit of the weight variance. However, as high order perceptrons converge in an almost optimal time if all weights are zero or very close to, there is no reason to use a higher value.

It can be easily seen that the activation function has an important influence on the optimal learning rate: the latter is for high order perceptrons with a logistic activaton function on average 25 times higher than for a linear activation function. This factor varies between 5 and 85 for the different data sets. This behavior is at least partly related to the lower first derivative of the logistic as compared to the linear activation function ($\frac{1}{4}$ at zero and even smaller for other values). If the logistic activation function is scaled to have a first derivative of one at zero, then the learning rate has to be divided by 16 in order to obtain the same network behavior. This number compares well with the difference between the optimal learning rate for the linear and logistic activation function (compare equation 1).

For the linear activation function, optimal learning rates between 0.02 and 0.6 have been observed. Surprisingly, for the logistic activation function this range is $[0.5, 7.0]$, where most rates are above 1.0, even though a learning rate smaller than 1.0 is usually recommended. The range $[0.005, 2.5]$, in which optimal learning rates have been found for the networks with the hyperbolic tangent activation function and trained on the classification problems, is very big as compared to the approximation problems. However, it is more likely that the data sets and target patterns being Boolean causes this behavior rather than the use of a different activation function (compare section 2).

The choice of the activation function changes also the convergence time: networks with a logistic activation function converge on average faster than those with a linear one. Although for the Solar data set, the network using logistic functions is not able to attain the same precision as a network with a linear activation function. *Vice versa*, first order perceptrons with a logistic activation function are able to learn the Servo data set up to a higher precision than the ones with a linear activation function.

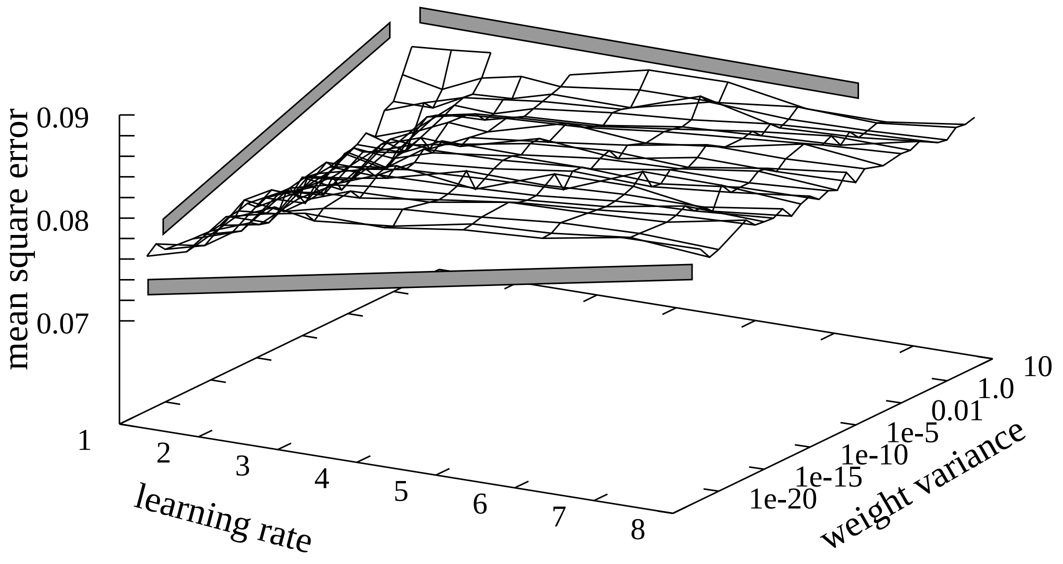

Figure 4: Generalization performance of a multilayer perceptron as a function of weight variance and learning rate for the Solar data set.

The estimation of the optimal learning rate according to P. Haffner *et al.* does not depend on the number of inputs or connections. This coincides with table 3 which reveals no correlation of these parameters and the optimal learning rate. On the contrary, the experiments performed with the Solar and Servo data sets show that two networks can have the same topology but different optimal settings for the learning rate and weight variance. These special settings for these parameters can therefore be regarded as a property depending on the information contents of the data set. However, those values are only of little help: the discrepancies among optimal learning rates for different data sets are big, and a learning rate can cause non-convergence for a certain data set, although it is optimal for another. No value for the constant c in his formula exists which is can be used for any data set.

The method of Y. K. Kim *et al.* does not match with the outcome of the experiments performed with the high order perceptrons, as he states that very small weight variances are not good.

The optimal settings seem also independent of the complexity of the problem: the Servo data set, which is supposed to be difficult to learn (which is confirmed by the high order perceptrons needing a larger number of training cycles to learn this data set as compared to others), has an optimal learning rate comparable to "simpler" data sets, as for example the Solar data set. Similar hypotheses based on a relation between the number of inputs, outputs, or patterns and an optimal setting of a training parameter can not be confirmed from the experiments.

6 Optimizing Generalization Performance

Table 4 shows the ranges of initial weight variances and learning rates for which the high order perceptrons performed best on the test data in terms of generalization performance.

In all but one experiment, high order perceptrons initialized with zero weights, or random values of a variance close to zero, performed optimally. The exception is represented by a first order perceptron with a linear activation function trained on the Servo data, which has a better generalization performance for initial weight variances above 3.0 (experiments were performed for variances up to 10^5, for which the training time was about 11 times as high as for a zero weights and 5 times as compared to a weight variance of 3.0).

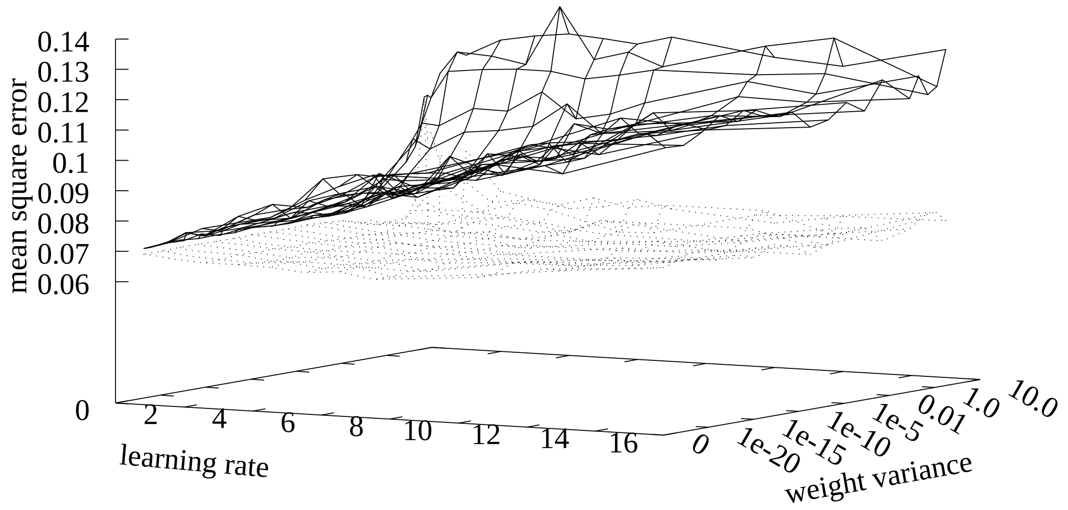

Figure 5: Minimal and maximal error as a function of weight variance and learning rate for the Solar data set.

No activation function is overall preferred since for three experiments the networks with the logistic function yield a better performance and during three experiments those where linear activation functions are used performed better (in the other cases the differences are within statistical error margins, or the results are not comparable due to different convergence criteria).

Note that the networks with all initial weights equal to zero do not produce the same solution for each simulation. The presentation of the patterns in a random sequence is sufficient to diversify them. Furthermore, the variance of the generalization performance on trained networks initialized with zero weights is usually similar to those for initial weight variances in the range of optimal values. This leads to the conclusion that the variety of the solutions for zero and small random weights is equal. However, small random weights may perform better for data sets for which a random presentation of the elements is insufficient to prevent weights from assuming similar values [11], although this behavior was not observed during all the experiments with high order perceptrons.

Comparing the optimal learning rates for fast convergence with those for a good generalization performance, the following can be observed: for the linear and hyperbolic tangent activation functions, the values are equal or similar. This behavior differs for the logistic activation function: the learning rate for fastest convergence is almost always higher as compared to those for best generalization performance. Similar to the results for fast convergence, a correlation between the number of connections and the optimal learning rate is not observed, even if only networks of the same order are considered.

7 Conclusion

For high order perceptrons an upper limit for the initial weight variance exists, below which both the network convergence and generalization performance are near-optimal (only one exception was observed for 27 series of experiments). In contrast to the multilayer perceptrons, even an initialization with zero weights gives near-optimal results if the learning rate is well-chosen. Consequently, a near-optimal generalization performance can be achieved with an initialization of high order perceptrons using zero or very small random weights. The latter choice should be preferred in order to prevent

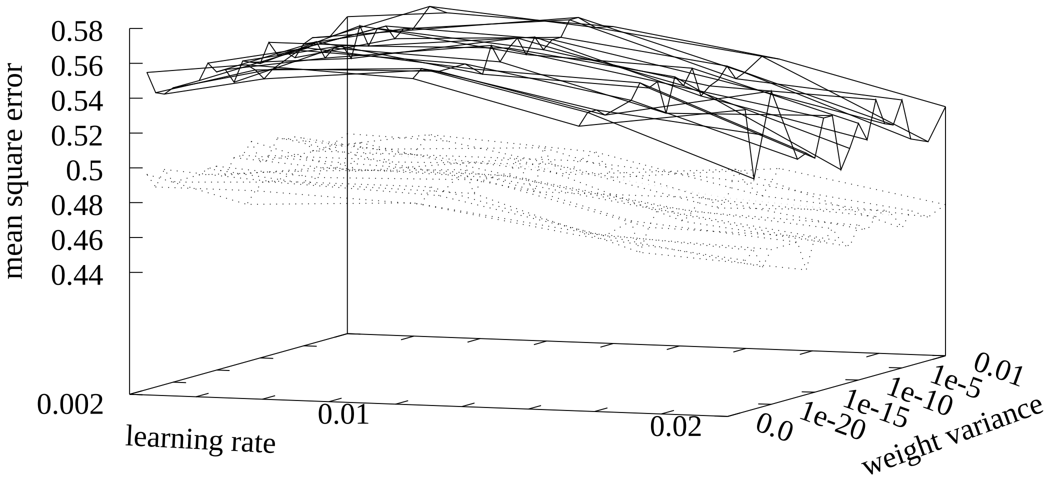

Figure 6: Minimal and maximal error as a function of weight variance and learning rate for the CES data.

trouble with exceptional data sets. However, the use of all initial weights equal to zero does not prevent the networks to assume different solutions: the presentation of the patterns in a random order (which is in any case advantageous to ensure convergence) is sufficient to 'break the symmetry'.

The optimal initial weight variance depends on the data set for both an optimal training time and generalization performance. These values do not depend in an observable way on the number of connections or the order of the network.

A data set independent method for the determination of an optimal learning rate could not be found. Moreover, the experiments show that the methods using only parameters concerning the network topology, such as the number of connections or the order of the network, as well as the type and steepness of the activation function, are most likely to fail. The optimal learning rate probably depends mainly on the clustering of the data and is therefore impossible to estimate in a simple way. However, the shape of the activation function changes the range of optimal learning rates (see the tables 3 and 4 for these ranges) which further depends on whether one optimizes training time or generalization performance. The best generalization performance can even be observed for learning rates which sometimes cause slow or non-convergence.

	Ω	$\frac{W}{N_2}$	[2]	[1]	[17]	[3]	[8]	[14]
Linear activation function								
CES	2	6	-	0.5	0.4	0.2	0.08η	-
	3	10	-	0.3		0.1	0.03η	-
Auto-	1	8	-	0.4	0.5	0.2	0.05η	-
mpg	2	36	-	0.8	0.03	0.05	0.002η	-
Solar	1	13	-	0.2	0.3	0.1	0.02η	-
	2	79	-	0.04	0.007	0.02	0.0005η	-
Servo	1	13	-	0.2	0.3	0.1	0.02η	-
	2	79	-	0.04	0.007	0.02	0.0005η	-
Glass	1	16	-	0.2	0.3	0.1	0.01η	-
	2	121	-	0.03	0.003	0.02	0.0002η	-
Logistic activation function								
CES	2	6	0.3	0.5	1.7	0.2	0.08η	0.1
	3	10	0.2	0.3		0.1	0.03η	0.04
Auto-	1	8	0.2	0.4	2.0	0.2	0.05η	0.06
mpg	2	36	0.05	0.8	0.1	0.05	0.002η	0.003
Solar	1	13	0.2	0.2	1.2	0.1	0.02η	0.02
	2	79	0.02	0.04	0.03	0.02	0.0005η	0.0006
Servo	1	13	0.2	0.2	1.2	0.1	0.02η	0.02
	2	79	0.02	0.04	0.03	0.02	0.0005η	0.0006
Glass	1	16	0.1	0.2	1.0	0.1	0.01η	0.02
	2	121	0.02	0.03	0.01	0.02	0.0002η	0.0003
Hyperbolic tangent activation function								
Br. vowels	2	66	0.002	0.05	0.009	0.03	0.0002η	0.001
Wine	2	92	0.002	0.03	0.005	0.02	0.0001η	0.005
Monk 1-3	2	154	0.0009	0.02	0.002	0.01	$4*10^{-5}\eta$	0.0002
Fi. vowels	2	231	0.0006	0.01	0.0009	0.008	$2*10^{-5}\eta$	$8*10^{-5}$
Digits	2	2081	0.0009	0.001	$1*10^{-5}$	0.0009	$7*10^{-7}\eta$	$9*10^{-7}$

An entry '-' means that this method could not be applied.

Table 2: Initial weight variances as calculated by different authors.

References

[1] Egbert J. W. Boers and Herman Kuiper. Biological metaphors and the design of modular artificial neural networks. Master's thesis, Leiden University, Leiden, The Netherlands, August 1992.

[2] Léon-Yves Bottou. Reconnaissance de la parole par réseaux multi-couches. In *Neuro-Nîmes'88; Proceedings of the International Workshop on Neural Networks and Their Applications*, pages 197–217, 269–287, rue de la Garenne, 92000 Nanterre, France, 1988. EC2 and Chambre de Commerce et d'Industrie de Nîmes. ISBN 2-906899-14-3.

[3] Gian Paolo Drago and Sandro Ridella. Statistically controlled activation weight initialization (scawi). *IEEE Transactions on Neural Networks*, 3(4):627–631, July 1992.

[4] Scott E. Fahlman. An empirical study of learning speed in back-propagation networks. Technical Report CMU-CS-88-162, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, September 1988.

[5] E. Fiesler and R. Beale (editors). *Handbook of Neural Computation*. Oxford University Press and IOP Publishing, 198 Madison Avenue, New York, NY 10016, 1997. ISBN 0-7503-0312-3 and 0-7503-0413-8.

[6] M. D. Garris and R. A. Wilkinson. *NIST Special Database 3.* National Institute of Standarts and Technology, Advanced System Division, Image Recognition Group, February 1992.

[7] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation*, volume I of *Computation and Neural Systems Series; Santa Fe Institute Studies in the Sciences of Complexity; Lecture notes.* Addison-Wesley Publishing Company, The Advanced Book Program, Redwood City, California, 1991. ISBN 0-201-51560-1.

[8] Y. K. Kim and J. B. Ra. Weight value initialization for improving training speed in the back propagation network. In *International Joint Conference on Neural Networks*, volume 3, pages 2396–2401. IEEE, 1991.

[9] M. Moreira and E. Fiesler. Neural networks with adaptive learning rate and momentum terms. IDIAP-RR 4, IDIAP, C.P. 192, 1920 Martigny, Switzerland, October 1995.

[10] P. M. Murphy and D. W. Aha (librarians). *UCI Repository of Machine Learning Databases.* UCI Repository of machine learning databases, ftp access ftp.ics.uci.edu: pub/machine-learning-databases, Irvine, CA: University of California, Department of Information and Computer Science,, 1996.

[11] David E. Rumelhart, James L. McClelland, and the PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1: Foundations. The MIT Press, Cambridge, Massachusetts, 1986. ISBN 0-262-18120-7.

[12] I. Saxena and E. Fiesler. Adaptive multilayer optical neural network with optical thresholding. *Optical Engineering*, 34(8):2435–2440, August 1995. Invited paper.

[13] W. H. Schiffmann, M. Joost, and R. Werner. Optimization of the backpropagation algorithm for training multilayer perceptrons. Technical report, Institute für Physics, University of Koblenz, Koblenz, Germany, 1992.

[14] Frank J. Śmieja. Hyperplane "spin" dynamics, network plasticity and back- propagation learning. GMD report, GMD, St. Augustin, Germany, November 28, 1991.

[15] Georg Thimm and Emile Fiesler. High order and multilayer perceptron initialization. *IEEE Transactions on Neural Networks*, 8(2), 1997.

[16] G. Thimm, P. Moerland, and E. Fiesler. The interchangeability of learning rate and gain in backpropagation neural networks. *Neural Computation*, 8(2):451–460, February 15, 1996.

[17] Lodewyk F.A. Wessels and Etienne Barnard. Avoiding false local minima by proper initialization of connections. *IEEE Transactions on Neural Networks*, 3(6):899–905, November 1992.

Linear activation function				
	Learing rate	Weight variance	Ω	Iterations
CES	0.5 - 0.6	0.0 - 0.1	2	14.9$\pm$4
CES	0.4	0.0 - 0.1	3	11.7$\pm$5
Auto-mpg[A]	0.1 - 0.15	0.0 - 0.01	1	10.4$\pm$6
Auto-mpg	0.1	0.0 - 0.01	2	34.3$\pm$33
Solar	0.2 - 0.3	0.0 - 0.001	1	5.3$\pm$4
Solar	0.2	0.0 - 0.0005	2	4.1$\pm$2
Servo[B]	0.07	0.0 - 0.001	1	3.0$\pm$3
Servo	0.12	0.0 - 0.1	2	166$\pm$4
Glass	0.1	0.0 - 0.01	1	8.7$\pm$5
Glass	0.02	0.0 - 0.0001	2	6.0$\pm$2
range	[0.02, 0.6]	max: [0.0001, 0.1]		
Logistic activation function				
CES	3.0	0.0 - 0.5	2	15.4$\pm$6
CES	2.0	0.0 - 0.01	3	10.1$\pm$4
Auto-mpg[A]	2.0 - 4.0	0.0 - 0.01	1	1.6$\pm$1
Auto-mpg	1.5 - 2.0	0.0 - 0.2	2	34.0$\pm$35
Solar[A]	5.0 -7.0	0.0 - 0.01	1	1.7$\pm$1
Solar	5.0 - 7.0	0.0 - 0.1	2	8.7$\pm$4
Servo	6.0 - 7.0	0.0 - 1.0	1	31.8$\pm$12
Servo	4.0 - 5.0	0.0 - 0.2	2	15.8$\pm$3
Glass	2.0	0.0 - 0.2	1	8.1$\pm$4
Glass	0.5	0.0 - 0.01	2	5.3$\pm$4
range	[0.5, 7.0]	max: [0.01, 0.5]		
Hyperbolic tangent activation function				
British vowels	0.005	0.0 - 0.0001	2	55.0$\pm$10
Wine	2.5	0.0 - 0.2	2	36.4$\pm$36
Monk 1	0.05 - 0.07	0.0 - 0.01	2	3.4$\pm$1
Monk 2	0.05	0.0 - 0.01	2	50.4$\pm$11
Monk 3	0.05	0.0 - 0.005	2	14.8$\pm$9
Finish vowels	1.5 - 2.0	0.0 - 0.2	2	54.3$\pm$14
Digits	0.1	0.0 - 0.01	2	13.3$\pm$7
range	[0.005, 2.5]	max: [0.0001, 0.2]		

[A] The error of 0.06 could not be reached, 0.075 is used instead

[B] The error of 0.07 could not be reached, 0.13 is used instead

Table 3: Best settings for learing rate and weight variance for high order perceptrons with a gain of 1 if fast learing is important.

Linear activation function					
	Learing rate	Weight variance	Ω	W	error
CES	0.5 - 0.7	0.0 - 0.1	2	2	$0.073_{\pm 1}$
CES	0.6	0.0 - 0.5	3	6	$0.070_{\pm 2}$
Auto-mpg[A]	0.1[C]	0.0 - 0.1	1	8	$0.066_{\pm 1}$
Auto-mpg	0.15	0.0 - 0.005	2	36	$0.057_{\pm 1}$
Solar	0.1	0.0 - 0.1	1	13	$0.073_{\pm 1}$
Solar	0.004 - 0.005	0.0 - 0.0001	2	79	$0.072_{\pm 0}$
Servo[B]	0.2	>3.0	1	13	$0.126_{\pm 1}$
Servo	0.12 - 0.15	0.0 - 0.1	2	79	$0.112_{\pm 1}$
Glass	0.15	0.0 - 1.0	1	16	$0.027_{\pm 1}$
Glass	0.04	0.0 - 0.001	2	121	$0.035_{\pm 1}$
range	[0.004, 0.7]	max: [0.0001, 1.0]			
Logistic activation function					
CES	1.0 - 2.0	0.0 - 0.5	2	2	$0.082_{\pm 1}$
CES	1.0	0.0 - 0.1	3	6	$0.086_{\pm 1}$
Auto-mpg[A]	0.2 - 2.0	0.0 - 0.01	1	8	$0.063_{\pm 1}$
Auto-mpg	0.3 - 0.4	0.0 - 0.2	2	36	$0.057_{\pm 1}$
Solar[A]	3.0 - 20.0[C]	0.0 - 5.0	1	13	$0.084_{\pm 1}$
Solar	0.3 - 1.0[C]	0.0 - 0.1	2	79	$0.069_{\pm 1}$
Servo	1.0[C]	0.0 - 2.0	1	13	$0.080_{\pm 0}$
Servo	0.1 - 7.0	0.0 -0.1	2	79	$0.0118_{\pm 1}$
Glass	3.5[C]	0.0 - 2.0	1	16	$0.026_{\pm 1}$
Glass	0.9	0.0 - 0.01	2	121	$0.034_{\pm 1}$
range	[0.1, 20.0]	max: [0.01, 5.0]			
Hyperbolic tangent activation function					
Br. vowels	0.005 - 0.01	0.0 - 0.01	2	726	$47.8_{\pm 2}$%
Wine	2.5[C]	0.0 - 0.7	2	276	$12.1_{\pm 5}$%
Monk 1	0.1[C]	0.0 - 0.05	2	154	$10.9_{\pm 5}$%
Monk 2	0.05[C]	0.0 - 0.05	2	154	$17.2_{\pm 5}$%
Monk 3	0.07[C]	0.0 - 0.01	2	154	$16.2_{\pm 3}$%
Fi. vowels	2.0[C]	0.0 - 0.2	2	1,155	$20.9_{\pm 7}$%
Digits	0.01 - 0.02[C]	0.0 - 0.01	2	20,810	$4.0_{\pm 1}$%
range	[0.005, 2.5]	max: [0.01, 0.7]			

[A] A max. error of 0.06 could not be reached, 0.075 is used instead

[B] A max. error of 0.07 could not be reached, 0.13 is used instead

[C] Better gener. is observable for learn. rates causing sometimes non-convergence.

The error is given as mean square diff., resp. as percent misclassification (%).

Table 4: Best settings for learning rate and weight variance for high order perceptrons with a gain of 1 if good generalization is important.