INFORMATICA, 1994, Vol. 5, No. 1-2, 241-255

SEARCHING FOR MINIMUM IN NEURAL NETWORKS

Vytautas VYŠNIAUSKAS

Institute of Mathematics and Informatics
2600 Vilnius, Akademijos St.4, Lithuania

Abstract. Neural networks are often characterized as highly nonlinear systems of fairly large amount of parameters (in order of $10^3 - 10^4$). This fact makes the optimization of parameters to be a nontrivial problem. But the astonishing moment is that the local optimization technique is widely used and yields reliable convergence in many cases. Obviously, the optimization of neural networks is high-dimensional, multi-extremal problem, so, as usual, the global optimization methods would be applied in this case. On the basis of Perceptron-like unit (which is the building block for the most architectures of neural networks) we analyze why the local optimization technique is so successful in the field of neural networks. The result is that a linear approximation of the neural network can be sufficient to evaluate the start point for the local optimization procedure in the nonlinear regime. This result can help in developing faster and more robust algorithms for the optimization of neural network parameters.

Key words: neural networks, optimization theory, pattern recognition.

1. Introduction. The main emphasis of neural networks is a massively connected and highly parallel system of simple processing units. This idea steams from the real biological systems. In fact, these units are huge simplification of the real biological neurons, so the term "artificial neural networks" (ANN) is widely used. The information in ANN is stored in connections (weights) in between the units. In principle, ANN can be easily adopted from one problem to another simply by

changing these weights properly.

With the invention of a new training technique (so called *backward error propagation* or *backpropagation* method) (Rumelhart, 1986) ANNs have been successfully applied for many diverse real–world problems such as mapping text to phonemes (Sejnowski and Rosenberg, 1987), determining the secondary structure of proteins (Qian and Sejnowski, 1988), playing backgammon (Tesauro and Sejnowski, 1988), identification of sonar signals (Gorman and Sejnowski, 1988). In principle, this new training method is a version of stochastic gradient descent, known in the literature as a stochastic approximation, which was solved conceptually by Robbins and Munroe (1951). However, ANN training has given a reputation for being very slow. Numerical optimization technique offers a rich and robust set of techniques which can be applied in an attempt to improve learning rates (Battiti, 1992). In particular, the conjugate gradient method is easily adopted to ANN (Johansson and *et.al.*, 1992).

Obviously, optimization of ANN is highly dimensional and multi-extremal problem, so, as usual, the global optimization methods would be preferable in this case. But the most astonishing moment is that the local optimization technique yields reliable convergence in many cases. This contradiction indicates that the process of searching for minimum in ANN is poorly understood. On the basis of Perceptron–like unit (Rosenblatt, 1958) we analyze why the local optimization technique is so successful in the field of neural networks.

2. Perceptron and discriminant function. A primary building block for the most architectures of ANN is Perceptron–like unit, which output is described as follows

$$y = \phi(\sum_{i=0}^{K} w_i x_i), \tag{1}$$

where $\{w_0, \cdots, w_K\}$ are adjustable weights (parameters) of the unit, $\{x_0, \cdots, x_K\}$ is $K+1$ dimensional input with assumption that $x_0 = 1$. The weight w_0 on this extra input is called the bias and is equivalent to a threshold of the opposite sign. It can be treated just like the other weights. $\phi(\cdot)$ is nonlinear, squashing function, which is essential to have a stable, nonlinear system. A large variety of ANN architectures can be composed by connecting the outputs of one group of the units with the inputs of another group of units. As an example, the feedforward ANN with one hidden layer can be easily constructed as shown in Fig. 1.

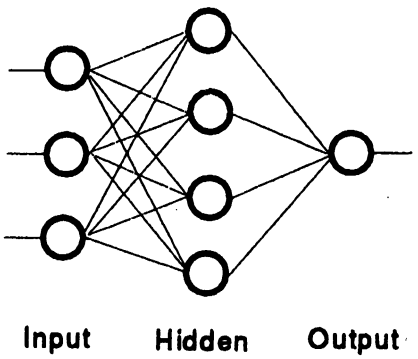

Fig. 1. Architecture of the feedforward ANN with one hidden layer. Solid circles denote the processing units, straight lines between them are connections.

A concept central to the practice of pattern recognition is that of discriminants. The idea is that a pattern recognition system learns adaptively from experience and distills various discriminants $D(x)$. In the case of a two classes A and B the task is to learn the set of weight values so that all patterns can be classified correctly using the same set of weights

$$\begin{aligned} D(x) &> 0, \qquad x \in A \\ D(x) &< 0, \qquad x \in B \end{aligned} \tag{2}$$

The simplest discriminant is a linear function

$$D(x) = \sum_{i=0}^{K} w_i x_i. \tag{3}$$

Note, that in essence, the Perceptron belongs to the class of linear discriminant functions, because y in (1) can be rewritten as follows

$$y(x) = \phi(D(x)) \tag{4}$$

Function ϕ is a subject to how the error of the classifier is measured. If we have a linear function ($\phi(x) \equiv x$) the error is computed as a some distance measure between the actual output and the target value. If we have "hard–limiting" step function, then we simply count the number of misclassifications. The linear output makes the classifier to be very sensitive to the outliers, e.g., only one specific data point drawn from the tail of the distribution can shift significantly the resulting linear discriminant. On the contrary, the error counting criterion makes no sense whenever the misclassification is tolerable or significantly large. But the positive feature in this case is outliers–insensitive classifier. A reasonable choice for ϕ is smooth, monotonic squashing function which gives outliers–insensitive classifier, capable to weight the classification error depending on the distance between the output and the target values.

3. Test on Highleyman's classes problem. We will analyze the performance of Perceptron–like unit (1) on the classical example of Highleyman's classes (see Highleyman, 1962). The set of data consists of two overlapping Gaussian distributions with the mean μ and dispersion σ parameters shown in Fig. 2. The overlap of these two classes is 6% but the optimal result for the linear discriminant function is 10% of errors. For this problem the linear discriminant function is

as follows

$$D(x) = w_o + w_1 x_1 + w_2 x_2. \tag{5}$$

The task is from a given set of N points ($N/2$ points belong to the class A, $N/2$ - to the class B) to find the optimal discriminant function by minimizing the error function

$$E = \frac{1}{2}\sum_{p=1}^{N}(y(x^{(p)}) - t^{(p)})^2, \tag{6}$$

where $t^{(p)}$ is the target (desired output value) associated to the point $x^{(p)}$. Our choice is $t = -0.5$ for the class A and $t = 0.5$ for the class B. $y(x^{(p)})$ is the output of the classifier for the point $x^{(p)}$.

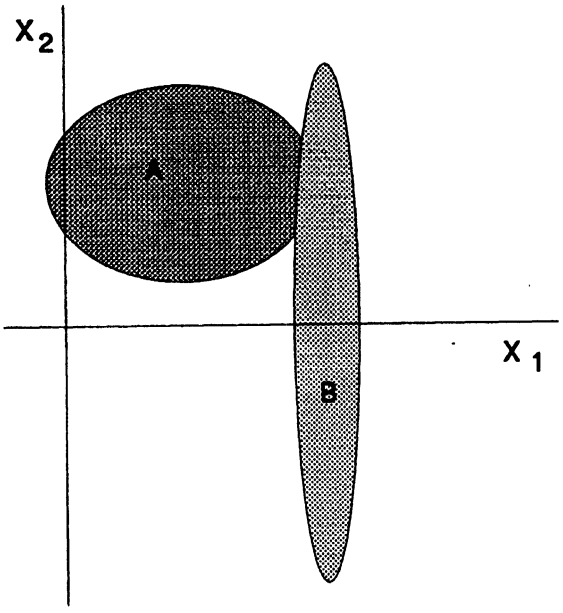

Fig. 2. Two overlapping Gaussian distributions of Highleyman's classes. Class A: $\mu = (1,1)$, $\sigma = (1,0.5)$, class B: $\mu = (2,0)$, $\sigma = (0.1,2)$.

In our analysis we will use the hyperbolic tangent function ($\phi(\cdot) = tanh(\cdot)$) which fulfills the necessary properties

$$\begin{aligned} &\lim_{x \to -\infty} tanh(x) = -1, \\ &\lim_{x \to \infty} tanh(x) = 1, \\ &tanh(0) = 0. \end{aligned} \tag{7}$$

An important issue of our analysis is to investigate the influence of the "steepness" factor of the output function ϕ. First of all, we will parameterize the linear discriminant function (5) (see also Wolff, 1966) in terms of spherical angles α_1, α_2 and radius distance R

$$\begin{aligned} w_0 &= R \cos \alpha_1 \\ w_1 &= R \sin \alpha_1 \cos \alpha_2 \\ w_2 &= R \sin \alpha_1 \sin \alpha_2 \end{aligned} \tag{8}$$

It is easy to see that these spherical angles α_1, α_2 completely define the location of the discriminant line with the slope a and intersection b

$$D(x) = 0: \qquad a = 1/\tan \alpha_2, b = 1/(\tan \alpha_1 \sin \alpha_2), \tag{9}$$

the radius R defines the error weighing strategy (Fig. 3) as follows

$R \ll 1$	linear output, squared distance criterion
$R \sim 1$	nonlinear smooth output function, weighed errors
$R \gg 1$	"hard-limiting" function, error counting

4. Results. At the first glance, minimization of (6) is not a trivial problem, because (6) is composed as a sum of non convex functions, and the final result can be extremely complicated, multi-modal function. The main goal of our investigation is to analyze how the error function (6) is effected

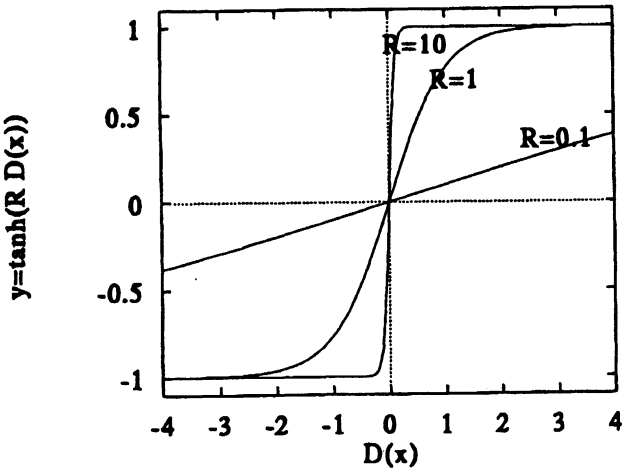

Fig. 3. Output function dependence on the radius parameter R.

by the radius parameter R and the sample size N. The influence of the "steepness" factor R can be investigated by plotting 3D projection of (6) versus α_1 and α_2 when R is fixed. Then a comparison for a different sample size can be made. These results are presented in Fig. 4–7.

In the linear case ($R = 0.1$) the error function is very smooth, unimodal surface (Fig. 4) and the variation of sample size from 6 to 200 makes a very little change of the surface shape. With the increasing of the radius parameter R new local minima appear, see for example the top picture in Fig. 5 when $R = 1$, $N = 6$. In this picture a new local minimum is approximately at $\alpha_1 = 0.4, \alpha_2 = 3.2$. For the larger radius parameter ($R = 10$) the error function becomes very complicated surface with multiple local minima. (see Fig. 5 at the bottom). A similar behavior is observed for the sample size $N = 20$ in Fig. 6. Note, that the increasing of the sample size N acts as a "low frequency" filter, which smoothes the error surface (compare the bottom pictures in Fig. 5–7). The most interesting result is that the location of the global minimum

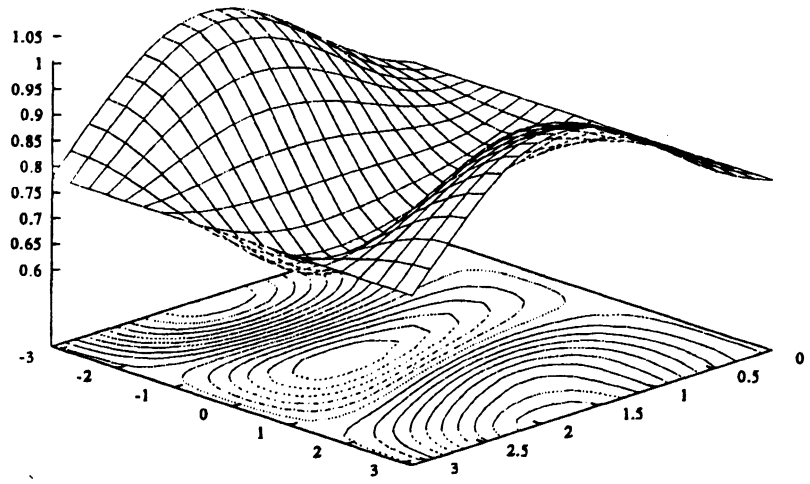

Fig. 4. The error function at the radius $R = 0.1$ when the sample size $N = 6$. Similar picture holds for 20 and 200 samples also.

is weakly effected by varying R and N in a wide range. It means that a solution obtained for a "small" R can be used as the start point of the local optimization procedure to find the minimum for the larger R.

Summarizing one can say that the error function surface mainly consists of elongated ravines, and one of them (where the global minimum is located) starts from the origin.

5. Discussion and conclusions. These results suggest some interesting conclusions. In practice, the optimization of ANN parameters is performed simultaneously (in our case α_1, α_2, R or w_0, w_1, w_2 in Cartesian domain). First of all, it is potentially dangerous to use a very small sample size, because of a large possibility to be trapped in to a local minimum. This conclusion completely corresponds the practice of pattern recognition.

Secondary, now we can explain why it is a good practice

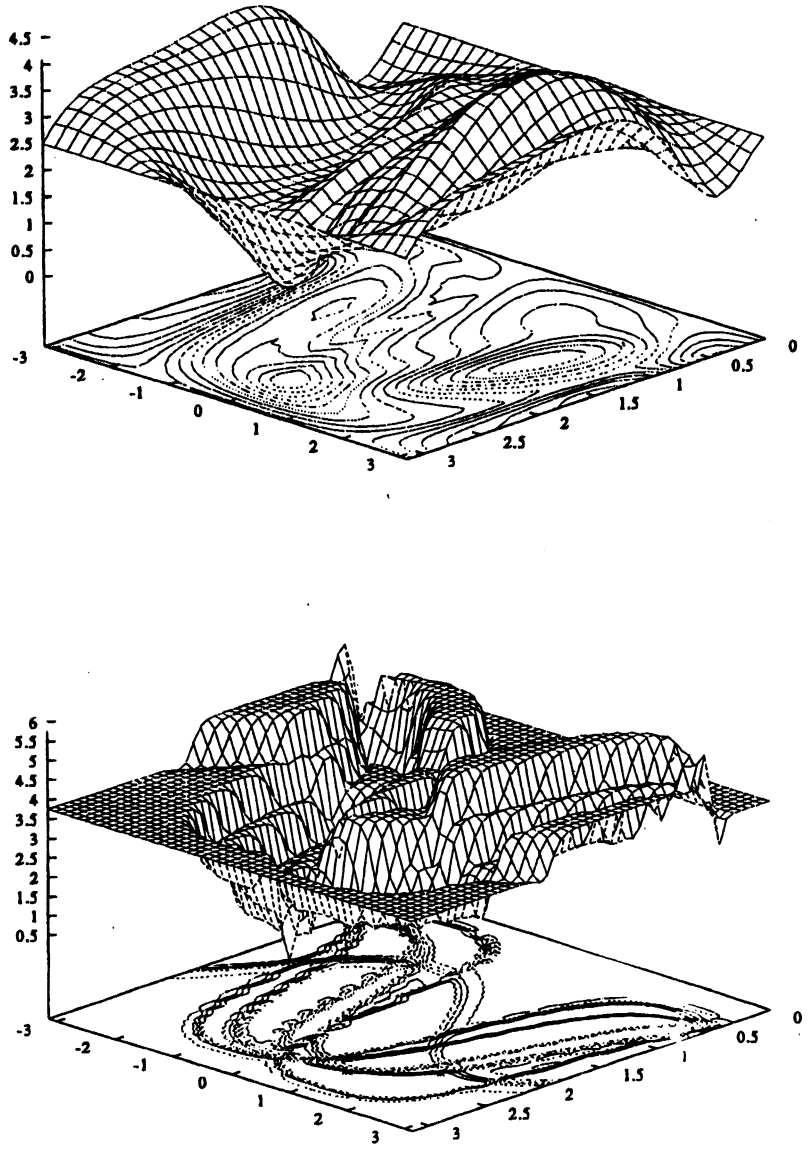

Fig. 5. The error function for $N = 6$ at the radius $R = 1$ (top) and $R = 10$ (bottom).

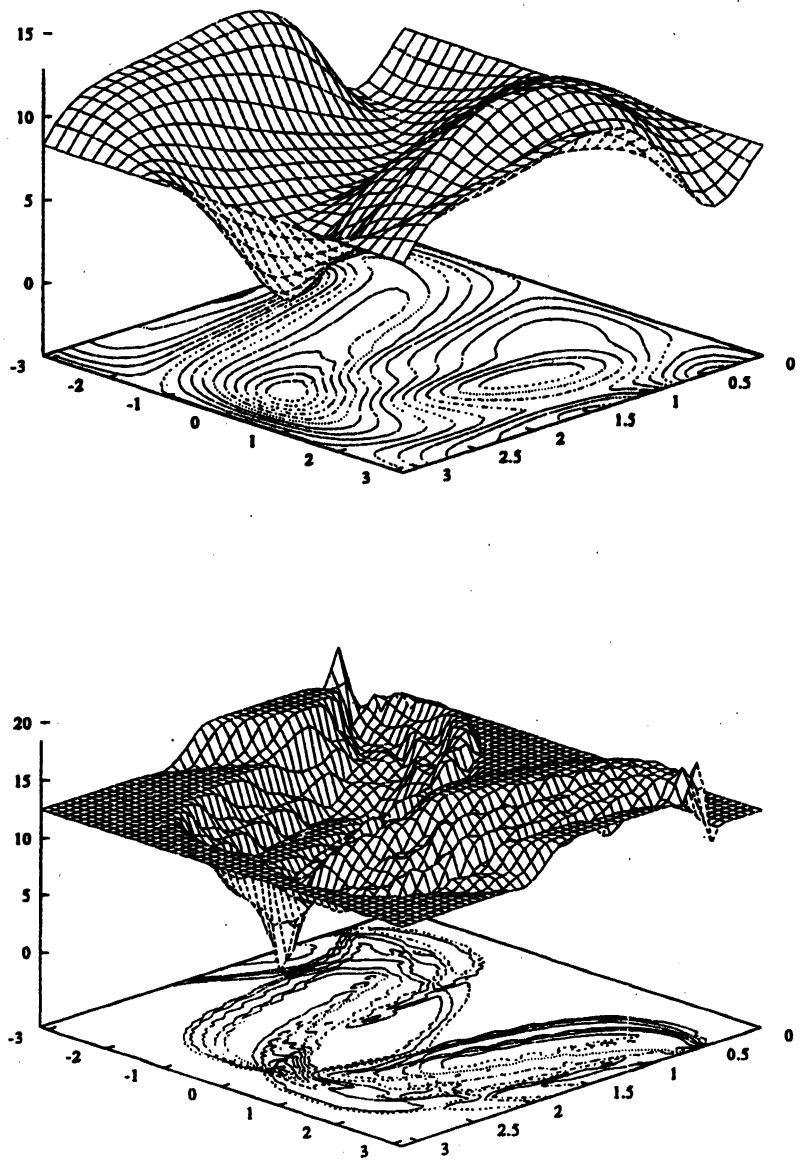

Fig. 6. The error function for $N = 20$ at the radius $R = 1$ (top) and $R = 10$ (bottom).

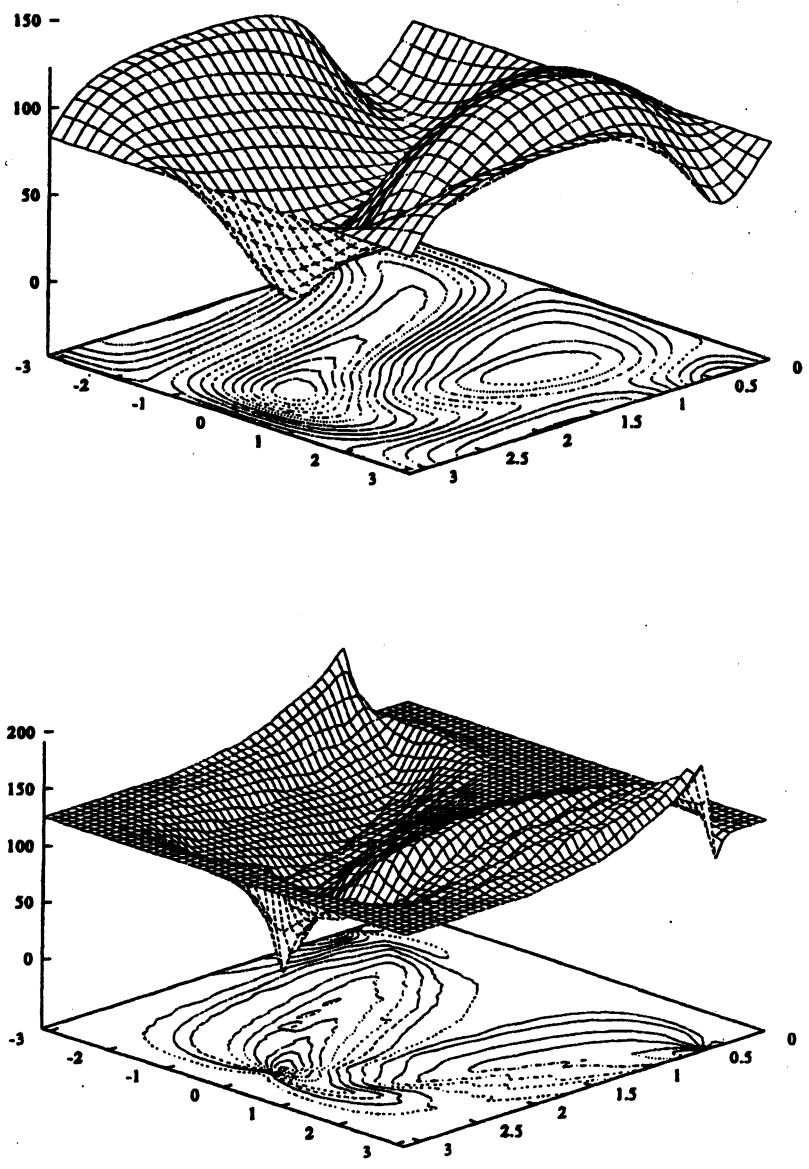

Fig. 7. The error function for $N = 200$ at the radius $R = 1$ (top) and $R = 10$ (bottom).

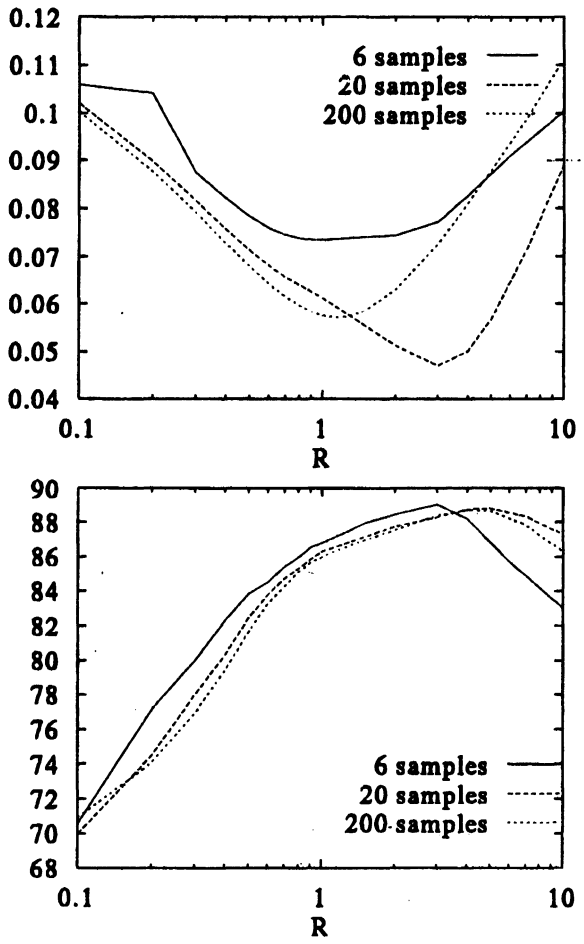

Fig. 8. Normalized learning error (top) and percentage of correct classification (bottom) dependence on the radius parameter R.

in ANN training to initialize the weights near the origin (e.g., to set small random values). In this case ANN is a linear system and a preliminary search for the global minimum can be easily performed by the local optimization technique. At the next phase the outputs of processing units begin to saturate, and a precise location of the global minimum is found by searching along the ravine, see Fig. 8 at the top, where the global minimum of (6) is shown versus the radius parameter R. When the optimization is performed over all 3 parameters simultaneously, obtained solution corresponds to the optimal R value. Alternatively, one can split explicitly linear and nonlinear stage of the optimization solving directly the linear approximation at first, and then using this result as the start point for the optimization in the nonlinear stage. Note, that the minimal learning error does not necessary correspond the optimal performance on the test set (we used 2000 independent test samples), as shown in Fig. 8 at the bottom. The advantage of the nonlinear output function with adaptive steepness factor is that a better performance on the test set can be achieved (70% in the linear case, 88% with the nonlinear output function).

Third, a possibility of the local optimization procedure to be trapped at the spurious minimum (where R is large) can be prevented by introducing an additional "regularization" term which "lifts up" the error surface at a large distance R. One of the possible constrains is the squared sum of all weights

$$r(w) = \sum_{i=0}^{K} w_i^2. \tag{10}$$

Then the minimization of (6) is replaced by

$$E' = E + \lambda r(w). \tag{11}$$

This additional constrain adds a weight "decay" term to the gradient of (6) and prevents the weights to get large values.

Indeed, this modification is also widely used in neural network optimization.

Finally, we do not have a complete confidence that the same picture can be always expected in ANN. A straightforward extension of these results to ANNs, which are very complex systems composed from a large amount of Perceptron–like units, is not so obvious. However, we hope that these results can help to develop faster and more robust algorithms for ANN training.

6. Acknowledgments. The author is indebted to Prof. Jonas Mockus for fruitful discussions. His criticism and expertise in the optimization theory highly stimulated this work.

REFERENCES

Battiti, R. (1992). First– and second–order methods for learning: between steepest discent and Newton's method. *Neural Computation,* **4**, 141–166.

Gorman, R.P., and T.J. Sejnowski (1988). Analysis of hidden units in a layered network trained to classify sonar signals. *Neural Networks,* **1**(1), 75–90.

Highleyman, W.H. (1962). Linear decision functions with applications to pattern recognition. *Proc. IRE–50,* 1501pp.

Johansson, E.M., F.U. Dowla and D.M. Goodman (1992). Backpropagation learning for multilayer feed–forward neural networks using the conjugate gradient method. *International Journal of Neural Systems,* **2**(4), 291–301.

Qian, N., and T.J. Sejnowski (1988). Predicting the secondary structure of globular protein using neural networks. *J. Molec. Biol,* **202**, 865–884.

Rumelhart, D.E., G.E. Hinton and R.J. Williams (1986). Learning representation by back–propagating errors. *Nature,* **323**, 533–536.

Robbins, H., and S. Munroe (1951). A stochastic approximation method. *Ann. Math. Statist.,* **22**(1), 400–407.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review,* **65**(6), 386–408.

Sejnowski, T.J., and C.R. Rosenberg (1987). NET Talk: a parallel network that learns to read aloud. *Complex Systems,* **1**, 145–168.

Tesauro, G., and T.J. Sejnowski (1988). A 'Neural' network that learns to play backgammon. In Anderson D.Z. (Ed.), *Neural Information Processing Systems,* AIP, NY. pp. 794–803.

Wolff, A.C. (1966). The estimation of the optimum linear decision function with a sequential random method. *IEEE Trans. Inform. Theory,* **IT–12**(3), 312–315.

Received April 1994

V. Vyšniauskas graduated from the Vilnius University, Department of Theoretical Physics in 1987. Since 1991 he has been with the Department of NeuroInformatics at the Institute of Mathematics and Informatics, where he is currently a Ph.D. student. The topic of the research is artificial neural networks with focus on function approximation, learning and generalization capabilities of artificial neural networks.