Statistical methods for construction of neural networks.

Włodzisław Duch and Rafał Adamczak
Email:duch,raad@phys.uni.torun.pl

Department of Computer Methods, Nicholas Copernicus University, Grudziądzka 5, 87-100 Toruń, Poland.

ABSTRACT

Despite all the progress in neural networks the technology is still brittle and sometimes difficult to apply. Automatic construction of networks and proper initialization of their adaptive parameters are the key factors to create robust neural networks. Methods of initialization of MLPs are reviewed and new methods based on statistical discriminants and logical rules are suggested. These methods in many cases achieve higher accuracy before learning starts than random initialization achieves after the learning process finishes.
KEYWORDS: Neural networks, discriminant analysis, initialization, neural architectures

1. Introduction

Learning and generalization in neural networks strongly depends on the complexity of the network used, including the architecture of the network, the number and types of parameters used by the network, the procedure used for initialization of its parameters and the details of the learning procedure. Models that are too complex may learn the training data perfectly but will not generalize well. It is commonly believed that the simplest models have the best generalization capabilities, but proper regularization of the cost function may ensure good generalization even in overparametrized models [1]. Finding global minimum of a complex, nonlinear error function with many parameters is an NP-hard problem [2]. Construction of appropriate architecture and proper initialization of adaptive parameters should enable finding close to optimal solutions for real-world problems, significantly decreasing the learning time.

We have observed that simple statistical discriminant or clusterization methods in many cases give results that are comparable or better than those found by neural networks. A review of different approaches to classification and comparison of performance of 20 methods on 22 real world datasets has been done within the StatLog European Community project [3]. The algorithms that appeared most frequently as the top five were all of statistical nature, including four discriminant approaches: linear (LDA), logistic (LogDA), quadratic discriminant (QDA) and a more involved DIPOL92 method that uses hyperplanes to discriminate between clusters. The last of the top five method, ALLOC80, is based on clusterization/density estimation techniques. It is also worth noting that the simplest version of the nearest neighbor method, using a single neighbor, achieved best results in 4 cases, and close to the best in another 3 cases. Our own results on several medical datasets [4] showed that logical rules that discriminate using hyperplanes perpendicular to the axes of the input variables are sometimes more accurate than any other method of classification. Among the top 5 algorithms MLPs trained with the backpropagation algorithm appear only once at the third position and 3 times at the fifth position. This clearly shows that in most cases MLPs did not find solutions as good as those found by statistical discriminant function methods.

In this paper we propose to use the statistical discriminants and logical rules for construction of the architectures and initialization of parameters in multilayered perceptrons. We have already used clusterization techniques to initialize our density estimation Feature Space Mapping (FSM) architecture obtaining significant improvements in classification accuracy with reduced complexity of the network [5]. After a suitable transformation of inputs to $N+1$-dimensional space clusterization techniques may also be used to initialize the MLP network using prototypes [5]. Procedures for automatic MLP network construction should propose architecture (the number of nodes and connections), weights (biases are also counted as weights), and also the slopes of sigmoidal functions (many MLP programs use unipolar sigmoids only). The structure of MLP networks is frequently optimized during learning, either using genetic or other global minimization methods. Such methods are computationally quite costly and so far have not produced competitive results. Regularization methods [2] are a good way to simplify an overparametrized architecture, but still may miss an optimal solution. Using computationally inexpensive discriminant or clusterization methods for MLP construction provides an interesting alternative that should simplify the training process.

In the next section methods of MLP initialization are briefly reviewed and several statistical methods of classification suitable for construction of MLPs described. The third sections presents the use of logical rules for network construction. Short discussion closes this paper.

2. Statistical methods for network construction.

Long training times and the suboptimal results of MLPs seem to be due to the lack of a proper initialization. In a series of computer experiments Schmidhuber and Hochreiter [6] observed that repeating random initialization ("guessing" the weights) many times is the fastest way to convergence. In other words, even sophisticated learning procedures are not able to compensate for bad initial values of weights, while good initial guess leads to fast convergence even with simple gradient-based error minimization techniques. Therefore good strategy is to abandon training as soon as it slows down significantly and start again from random weights. Wrong initialization may create network of sigmoidal functions dividing the input space into areas where the network function gives constant inputs for all training data, making all gradient procedures useless. If some weights overlap too much (scalar product $\mathbf{W}\cdot\mathbf{W}'/|\mathbf{W}||\mathbf{W}'|$ is close to 1) the number of effective hyperplanes is reduced.

Random weight initialization is still the most popular method. Wessel and Bernard [7] aim at a uniform covering of all data space with hyperplanes, setting the hidden-output layer weights to 1. Several random initialization schemes have recently been compared by Thimm and Fiesler [8] using a very large number of computer experiments. The best initial weight variance is determined by the dataset, but differences for small deviations are not significant and weights in the range ± 0.77 give the best mean performance. A few authors proposed initialization methods which are not based on random weights

but we have no space to review them here [5].

The results of the Statlog [3] project, as well as our own results, show that in many cases statistical methods are more accurate than neural or machine learning approaches. These methods are more than an order of magnitude faster than neural networks and use much simpler, linear description of decision borders. A more drastic example in which random initialization fails comes from classification of "interesting" events in High Energy Physics data [9]. The best results were obtained for a one-bit representation of weights (interpreted as +/- 5 weight values). The generalization levels reached 90% while in the standard MLP they were only 62%.

There are several inexpensive methods that should be suitable for initialization of neural classifiers, including statistical discriminant functions, decision trees such as C4.5 or CART, and clusterization methods. We will consider statistical discriminant and clusterization techniques here, although the use of decision trees is equally worth investigating. Classical statistical discriminant techniques include Fisher, linear, logistic, multiple discriminant analysis, and more modern methods like DIPOL92 [10]. The Fisher discriminant analysis [11] is based on a projection on a single line passing through the origin, trying to increase the separation of projected points between the classes and decrease the separation of projected points inside the same class. The error function becomes:

$$E(W) = \frac{W^T S_B W}{W^T S_W W} \quad (1)$$

$$S_W = \sum_i \sum_{X \in C_i} (X - \bar{X_i})(X - \bar{X_i})^T \quad (2)$$

$$S_B = \sum_{i>j} (\bar{X_i} - \bar{X_j})(\bar{X_i} - \bar{X_j})^T \quad (3)$$

where $\bar{X_i}$ is the sample mean for class i, the matrix S_B is the between-class scatter (covariance) matrix and S_W is the in-class scatter matrix. The solution to the maximization of $E(W)$ for the two-class case is:

$$W_1 = S_W^{-1}(\bar{X_1} - \bar{X_2}) \quad (4)$$

Thus for the k-class problem the simplest neural architecture based on the FDA is composed of the input layer and the hidden layer, with one hidden neuron per output class, serving also as the output units (in practice $k-1$ output units suffice, with the default class corresponding to zero output of all hidden units). The weights W_i separating class i from all other classes are used for the hidden neuron number i and the bias is determined from the probability distribution of the projections $x = W \cdot X$. The bias b is obtained from the solution of the $p_i(x) = p_r(x)$, where $p_r(x)$ is the probability distribution of projections from all other classes (both may be zero for well separated classes). This probability distributions are estimated from smoothed histograms for the one-dimensional projections. The hidden neuron performs $\sigma((W \cdot X + b)T)$ where the variable T determining the slope of sigmoid may additionally be optimized by fitting a line to the difference $p_i(x) - p_r(x)$ around x for which the two distributions cross.

The simplest network constructed from FDA solution gives classification error which is as good as the original FDA. For such datasets [12] as Wisconsin breast cancer, hepatitis, Cleveland heart disease or diabetes the network obtains better results already before the learning process starts, but for some datasets this is not the best approach since separation of a single class from all others may be difficult. Suppose that vectors from class C_1 are not separated well by FDA procedure from all other vectors. In such a case separation from individual classes may still work (since weights are computed from means of classes), or classes should be broken into several subclasses (clusters) before applying FDA. The weights depend directly on the selection of the vectors used to compute $\bar{X_2}$ mean, giving us a lot of flexibility in their selection. This leads to a more sophisticated construction of the network, with several hidden neurons per class and one output neuron per class connected to those hidden layer units that discriminate this class from all others. The hidden-output layer weights are all equal to 1.0 and the bias is determined by selecting the smallest activation of the output unit after presentation of all vectors from a given class.

Empirical comparison of FDA with neural networks on some datasets taken from UCI repository [12] show that despite the extreme simplicity of the method it is sometimes highly accurate. For the Wisconsin breast cancer data Fisher discriminant obtains in the 10-fold crossvalidation tests 96.8% accuracy, while randomly initialized MLPs give slightly worse results with much larger effort. For the Hepatitis dataset FDA achieves 84.5% while MLP result is 82.1%, for Cleveland Heart disease (original version) 84.2% comparing to 81.3% for MLP, and for Indian Pima diabetes almost the same accuracy is obtained. In all these cases there were only two classes, therefore only one projection line has been used (one neuron in MLP). We are performing experiments now to determine what is the optimal number of projections, although it is quite possible that for these datasets best results are obtained with the simplest architecture.

Linear discriminant analysis (LDA) is an alternative statistical procedure that may be used for construction of neural networks. LDA fits a separating plane between vectors of a given class and all other vectors. If the weight vector W includes the bias $W_0 = -b$ and the input data vectors X include $X_0 = 1$ the discrimination of class C_1 vectors from all others requires to find a solution to:

$$X_i^T \cdot W = \begin{cases} > 0 & \text{if } X_i \in C_1 \\ < 0 & \text{otherwise} \end{cases} \quad (5)$$

This equation is further simplified if all vectors from classes $C_i, i = 2..k$ are replaced by their negative. Collecting all input vectors in a rectangular matrix A the discriminating plane is found by solving $A \cdot W > 0$. Introducing a vector B with small positive components the inequality is replaced by a linear equation $A \cdot W - B = 0$ which may be solved by a pseudoinverse method [11] or several other methods. LDA is sometimes justified also by the maximum likelihood method using multivariate normal distribution, but there is no need to make such assumptions. If one hyperplane is not sufficient for discrimination between a given class and all others a clustering procedure to decompose this class into several subclasses is applied.

Each hyperplane requires a hidden neuron. If only one neuron per class is sufficient there is no need for additional output neurons, otherwise an output layer combines results from hidden neurons discriminating a single class. The LDA procedure determines all weights and biases for the input-hidden layer, while the output-hidden layer weights and biases are set in the same way as in the FDA case. One problem is that LDA procedure, minimizing the classification error, does not determine an optimal separating plane, i.e. the plane may be close to the vectors of one class and far from the vectors of other classes. MLP minimizing the error function for soft sigmoids places the separating hyperplanes at better positions. Solving $A \cdot W - B = 0$ by Ho-Kashyap method [11] with additional requirement that the norm $||B||$ should be maximum increases the overall separation of the discriminating hyperplane from the vectors of different classes. A better condition for optimal separation should include only the maximization of the squared sum of distances $X_i \cdot W/|W|$ between the hyperplane and the vectors X_i that are close to the hyperplane, i.e. many rows from the A matrix corre-

sponding to vectors far from the hyperplane may be deleted.

Thus many variants of the LDA procedure are suitable for network construction. Empirical comparison of the simplest version based on pseudoinverse solution for the Hepatitis dataset [12] gives 86.4% accuracy (10-fold crossvalidation tests) versus 82.1% for randomly initialized MLP, for the Cleveland Heart disease 84.5% versus 81.3% and for the Pima Indian Diabetes 77.5% versus 76.4%. For the Wisconsin breast cancer data LDA accuracy is 96.0%, while MLP after training achieves slightly better result of 96.7%. In all cases networks with a single neuron are constructed since these are two-class problems. We are experimenting now with adding more than one hyperplane to increase the complexity (and hopefully the accuracy) of MLP networks.

Logistic Discriminant Analysis (LogDA) maximizes conditional likelihood, modeling ratio of the probability density functions between two classes. This method frequently gives very similar results to the LDA (for normal distributions with equal covariances results are identical) but since it requires much more computational and programming effort we will not consider it here. Quadratic Discriminant Analysis (QDA) provides decision borders that are quadratic surfaces. Although combination of slowly varying (small slopes) sigmoidal functions may provide similar decision borders we have not tried to use it for initialization yet. More sophisticated statistical methods, such as DIPOL92 [10] may also be used for network construction but we have not tried them so far. Results from the StatLog project [3] show that it would certainly be worthwhile to try.

3. Logical rules for network construction.

Table 1: Classification results for a number of optimized MLP training algorithms applied to the thyroid dataset – only the best results are shown. BP = Backpropagation.

Method	Training %	Test
k-NN (Manhattan)	–	93.8
Bayes rule [14]	97.0	96.1
BP+conjugate gradient	94.6	93.8
Best BP	99.1	97.6
RPROP	99.6	98.0
Quickprop	99.6	98.3
BP + genetic optimization	99.4	98.4
Local adaptation rates	99.6	98.5
Cascade correlation	100.0	98.5
PVM [14]	99.8	99.33
CART [14]	99.8	99.36
C-MLP2LN (our logical rules)	99.9	99.36

Logical rules proved to be quite accurate, especially for medical diagnosis, in a number of our previous studies [4]. Although our method is a combination of neural/search based optimization logical rules may be induced in a great number of ways. Rules provide feature selection, a very simple description of decision borders and MLPs may refine this symbolic knowledge achieving higher classification accuracy. In the hypothyroid dataset case [12] two types of the disease, primary hypothyroid and compensated hypothyroid, are diagnosed and differentiated from normal (no hypothyroid) cases using the results of 22 medical tests. Thus the problem has 3 classes and 22 attributes, 3772 cases for training and 3428 cases for testing. This data was used by Schiffman *et al.* [13] in optimization of several MLPs and related models. About 15 MLPs trained with different variants of backpropagation and cascade correlation algorithms were used. In addition tedious genetic optimization has been performed [13] on many network architectures. The best results of this study are reported in the Table 1. On the other hand we have derived crisp logical rules [4] using our C-MLP2LN method. These rules divide the input space with discriminating hyperplanes perpendicular to the axes. The best set of rules for the three classes is:

C_1: $\mathsf{TSH} \geq 30.5 \wedge \mathsf{FTI} < 64.2$
C_1: $\mathsf{TSH} \in [6.2, 29.5] \wedge \mathsf{FTI} < 64.1 \wedge \mathsf{T3} < 23.2$
C_2: $\mathsf{TSH} \geq 6.0 \wedge \mathsf{FTI} \in [64.3, 186.7] \wedge \mathsf{TT4} < 148.5 \wedge$ on thyroxine=no $\wedge$ surgery=no
C_3: ELSE

These rules make only 4 errors on the training set (99.89%) and 22 errors on the test set (99.36%). They are similar to those found using heuristic version of PVM method by Weiss and Kapouleas [4]. The differences among PVM, CART and C-MLP2LN are for this dataset negligible (Table 1), but all other methods give results that are significantly worse. Poor results of k-NN are especially worth noting: the default rate corresponding to the normal instances of the test set is 92.7% or 250 errors (standardized data with k=1 were used, as in the StatLog study, results with other distance function are even worse).

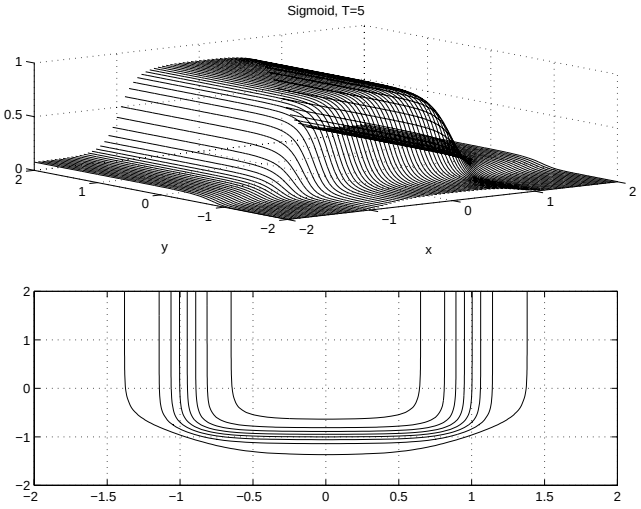

Figure 1: Decision borders corresponding to $x \in [-1, +1]$ and $y \geq -1$ rules implemented by neurons with $T = 5$.

We have tried to construct an MLP starting from logical rules, using 3 neurons for two classes (the third class is treated as a default). One may either remove completely all connections corresponding to inputs that are not used by the rules or leave them with zero weights. If the network is initialized with very steep slopes the same results as from the logical rules are obtained on the test and training set. Very steep slopes create a problem for gradient-based learning, since steep slopes give non-zero gradient only for a few vectors falling in the range of high network function variability. Even with $T = 5$ the region of high sigmoid variability may span the whole range of the standardized data. The MLP decision borders for $T \approx 1$ have hyperellipsoidal or hyperbolical shapes instead of approximating the hyperrectangular shapes that logical rules provide. Gradually decreasing the slope may lead to rapid changes in the number of errors since several sigmoidal functions added together create irregular decision border shapes. We have used batch training to avoid the deterioration of the initial high accuracy but softening of the sigmoids always leads to inferior results. Thus in this case there is no easy

way to obtain better results with gradient based techniques. However, the problem may be specific to medical data: if the degree to which thyroid problem is present was given, instead of the yes-no decisions based on values of medical tests, the decision borders could be softer and training would improve the accuracy.

4. Summary and discussion

Various methods of initialization of adaptive parameters in MLP networks have been briefly discussed. Initialization of MLPs is still done more often by randomizing weights [8], although initialization by prototypes based on initial clusterization presented in [5] and network construction based on discriminant techniques should give much better results enabling solutions to complex, real life problems. Introduction of this methods of network construction should allow for creation of robust neural systems requiring little optimization in further training stages. However, for problems requiring sharp decision borders MLPs trained with the gradient-based techniques may not be the best models.

The main point of this paper is: there is structure in the data that is easily recognized by discriminant and cluster-based methods and it is foolish not to use this structure, relaying on random initializationa and forcing the networks to laboriously discover it. Since some statistical methods are computationally much less expensive than neural approaches in all cases where they are applicable they should be used as a reference and as a starting point for neural methods. Nonlinear methods are usually difficult to converge and a good start from linear approximation makes the learning process easier, especially in complex cases, when neural solutions are hard to find (good benchmark problems are the two spiral or the parity problem here). Software for statistical methods is much harder to find than the software for neural simulations. For example, we were able to find a program for FDA only in the IMSL Fortran library. This software is simple to write and we are convinced that statistical and neural methods should be integrated in a single software package.

Acknowledgments: Support by the Polish Committee for Scientific Research is gratefully acknowledged.

References

[1] Neal R, *Bayesian Learning in Neural Networks*, Lecture Notes in Statistics vol. 118, Springer, 1996

[2] Bishop C, *Neural networks for pattern recognition*. Clarendon Press, Oxford, 1995

[3] Michie D, Spiegelhalter D.J, Taylor C.C, *Machine learning, neural and statistical classification.* Elis Horwood, London, 1994

[4] Duch W, Adamczak R, Grąbczewski K, "Extraction of crisp logical rules using constrained backpropagation networks." *Int. Conf. Neural Networks (ICNN'97)*, Houston, 9-12.6.1997, pp. 2384-2389; Duch W, Adamczak R, Grąbczewski K, Żal G, "Hybrid neural-global minimization method of logical rule extraction", Journal of Advanced Computational Intelligence (submitted)

[5] Duch W, Adamczak R, Jankowski N, "Initialization and optimization of multilayered perceptrons.", 3rd Conf. Neural Networks and Their Applications, Kule, Poland, October 1997, pp. 105-110; "Initialization of adaptive parameters in density networks", ibid, pp. 99-104

[6] Schmidhuber J, Hochreiter S, Guessing can outperform many long time lag algorithms. Technical Note IDSIA-19-96, 1996

[7] Wessel F.L, Barnard E, "Avoiding false local minima by proper initialization of Connections", IEEE Trans. Neural Networks 3 (1992) 899-905

[8] Thimm G, Fiesler E, "Higher order and multilayer perceptron initialization", IEEE Trans. Neural Net. 8 (1997) 349–359

[9] Battiti R, Tecchiolli G, "Training neural nets with the reactive tabu search." IEEE Trans. Neural Net. 6 (1995) 1185-1200

[10] Schulmeister B, Wysotzki F, "The Piecewise Linear Classifier DIPOL92". Proc. European Conf. Machine Learning, Catania, Italy, 1994, pp. 411-414

[11] Schalkoff R, *Pattern Recognition. Statistical, Structural and Neural Approaches.* J. Wiley, 1992

[12] C.J. Mertz, P.M. Murphy, UCI repository of machine learning databases, http://www.ics.uci.edu/pub/ machine-learning-databases

[13] Schiffmann W, Joost M, Werner R, "Comparison of optimized backpropagation algorithms", Proc. European Symp. Artificial Neural Networks, Brussels 1993, M. Verleysen (Ed.), de Facto Press, pp. 97-104

[14] S.M. Weiss, I. Kapouleas. "An empirical comparison of pattern recognition, neural nets and machine learning classification methods". In: J.W. Shavlik and T.G. Dietterich, *Readings in Machine Learning*, Morgan Kauffman Publ, CA 1990