Searching for optimal MLP

Włodzisław Duch, Krzysztof Grąbczewski
Department of Computer Methods, Nicholas Copernicus University,
Grudziądzka 5, 87-100 Toruń, Poland.
E–mail: {duch, kgrabcze}@phys.uni.torun.pl

Abstract

Backpropagation based on minimization algorithms is replaced by heuristic search techniques for quantized weights. The resulting algorithm is fast, avoids local minima of the cost function, and may be used either as initialization method for standard backpropagation or as a logical rule extraction technique.

I. INTRODUCTION

BACKPROPAGATION (BP) methods use local, gradient-based minimization techniques for training the multilayer perceptrons (MLP). Gradient-based minimization of the cost function is relatively fast, but for complex problems, with a large number of vectors and input features, it has several drawbacks and may be very time consuming. Despite the use of momentum and multistart techniques, requiring extensive experimentation, BP training may find local minima of the error function that are far from the global minimum. An additional problem with MLPs using sigmoidal transfer functions is that minimization of the mean square error (MSE) is not the same as minimization of the classification error. Smooth sigmoidal functions contribute to the MSE even for vectors that are far from the decision borders. MLP decision borders tend to have wide margins (placed far from vectors of both classes that are discriminated), which is good for generalization, but for overlapping distributions decision borders are shifted away from the class which has higher density of vectors near the decision border. An optimal placement of decision borders requires an increase of the slopes of sigmoids, but gradient procedures are not stable during such transition since the volume of the feature space, in which gradients are non-zero, shrinks rapidly to zero when sigmoids are changed into step functions. In effect linear discrimination analysis may sometimes obtain better results than non-linear neural techniques [1].

To avoid such drawbacks one may use either global minimization techniques [2], which from computational point of view are rather expensive, use better initialization methods [3], [1] or try to average over an ensemble of neural models [4], [2]. In this paper a radically different approach is proposed. Minimization and search methods [5] share the same goal of optimizing some cost functions. Quantization of network parameters (weights and biases) allows to replace minimization by search. Increasing step by step the resolution of quantization from coarse to fine, allows to find the network parameters to arbitrary precision. Search-based optimization allows to use step-like functions or smooth sigmoids. At worst heuristic search procedures may be applied to initialize network parameters, bringing the network close to the global minimum of the cost function. Further training with standard backpropagation should then be short and give better solutions than random initialization allows. At best search-based MLP may replace the BP algorithm completely. An additional advantage of the search procedures concerns the logical rule extraction from data. Although we have obtained excellent results converting MLPs into logical-like networks [6] the process is not automatic and requires tuning of regularization parameters by the user.

II. SEARCH-BASED OPTIMIZATION OF MLP COST FUNCTION

There are several ways to introduce search techniques into neural networks. In this paper we will consider fully discrete situation, in which both the data and all network parameters are discrete, although the resolution of the network parameters may be different than the resolution of the data. Inputs may also be continuous and only the network parameters discretized. If the step functions are used instead of smooth sigmoids the MSE error function becomes the classification error function.

Replacing the gradient-based BP methods by global search algorithm to minimize the value of the error function is rather expensive, therefore some form of a heuristic search should be used, for example the best first search or the beam search [5]. Even if the best first search algorithm is used (corresponding to the steepest gradient descent) a good solution may be found decreasing the resolution of the discrete network parameters. In BP this would roughly correspond to a period of learning with large learning constants, with some annealing schedule for decreasing the learning constant. It is surprising that the correspondence between search and minimization techniques has not yet been analyzed in details since it seems to be quite fruitful.

The first step of the algorithm is to design a network. Although the method does not require special architectures our tests were made on networks with similar structure for all the datasets (up to three hidden neurons per output, with adaptive weights only between the input and the hidden neurons). We have also used a constructive method, adding more neurons as long as the results were improving. If logical rules are extracted from the network it is easy to stop this procedure before overfitting occurs – the number of logical rules grows rapidly when overfitting occurs. On the other hand if an ensemble of networks is used in some averaging procedure one should also include networks that overfit the data [7], [8]. The transfer functions used are of the standard sigmoidal type. For the purpose of logical rules extraction sigmoids with very large slopes or the threshold functions are used.

In the fully discrete situation (if the data are not discrete we discretize them) we may replace all input data by binary variables. In such case the number of input units is equal to the sum of the numbers of discrete values for all the inputs. Each of the input data is then converted into the form of tuples consisting of ± 1 (as in the case of MLP2LN method [6]). The number of outputs is usually equal to the number of output classes. Choosing one of the classes as the default reduces the number of outputs by one. A data vector is then classified to the default class if none of the outputs gets sufficiently activated.

Given a network the algorithm starts with all weights $W_{ij} = 0$ and biases $\theta_i = -0.5$, so that all data is assigned to the default class. It is also possible to start with random values of the parameters, but so far we have investigated only the fully deterministic approach. At the beginning of the search procedure the step value Δ for weights (and biases) is set. This value is added or subtracted from weights and biases, $W_{ij} \pm \Delta, \theta_i \pm \Delta$. This significantly reduces the space searched through. The best first and the beam search strategies were used to modify one parameter at a time. Since computer experiments showed that it may not be sufficient variants of the search method modifying two weights at a time were used, though more time consuming. To improve the search methods they are performed in two stages. First, all the single changes of parameters are tested and a number of the most promising (i.e. decreasing the value of the cost function) changes is selected (the beam width). Second, all pairs of parameter changes from the chosen set, or even all the subsets of this set, are tested, and

the best combination of changes applied to the network. Since the first stage reduces the number of weights and biases that are good candidates for updating the whole procedure is computationally efficient.

One could train all the network parameters simultaneously, but more effective variant is the constructive approach, which determines parameters for each newly added hidden neuron separately. After convergence these parameters are kept frozen and new neurons are added for further training if necessary. Depending on the method of searching and the value of Δ results can be used for initialization of BP training on the neural network found by search procedure or for the crisp or fuzzy logical rule extraction.

A. Logical rule extraction

Adding some constraints to the cost function optimized can produce networks easily convertible to crisp logical rules or fuzzy logical rules with soft trapezoidal membership functions obtained by subtracting two sigmoids. If all the weights are integers – which can be obtained with setting the $\Delta = 1$) and the hidden neuron transfer function is sufficiently steep, then the resulting network can easily be converted to a number of M-of-N rules (if M conditions out of N are satisfied then the condition is true). The rules are generated by simple analysis of network parameters. All the input combinations are checked, and if their sum exceeds appropriate bias a logical rule is generated. Although sometimes the logical description of data is much shorter if M-of-N style rules are used classical logic form of rules is preferred, because they are more comprehensible.

To obtain small number of classical logic rules the space of weights values is searched assuming that biases are always equal to the sum of the incoming weights minus 0.5, i.e. $\theta_i = \sum_j W_{ij} - 0.5$. In such cases single neuron is equivalent to just one logical rule, since one combination of inputs gives a sum greater than the bias.

B. MLP initialization

Weights and biases found during the search represent a proper neural network which can be further trained with any variant of the backpropagation algorithm. The network parameters should be very close to a minimum of the error function, so usually a short training process is sufficient. Quite often backpropagation training has not been able to improve the accuracy of the initial network, which shows that search can yield more accurate results than gradient descent methods. Since the search method may use sigmoids with high slopes it works also in cases when the standard MLPs fall behind linear discrimination. Our experience with soft sigmoids also shows that backpropagation is frequently not able to improve classification results of the networks found using search procedures.

III. ILLUSTRATIVE RESULTS.

The search based (S-MLP) approach has been applied to several benchmark datasets giving in most cases very promising results. S-MLP results for classification on 3 datasets are given below and results of the rule extraction are given in the next subsection.

A. Black box classification

10-fold cross validation tests on the Iris flowers problem [9] gave 96% of accuracy. It is hard to obtain better results with cross validation since there are several vectors from the

TABLE I

Results for the Ljubljana cancer dataset.

Method	Accuracy %	Reference
S-MLP, 2 weights per step	83.6	This paper
C-MLP2LN, 2 rules	78.0	Ref. [10]
Assistant-86	78.0	Ref. [12]
CART	77.3	Ref. [13]
CART, PVM, C-MLP2LN single rule	76.2	Ref. [13]
Naive Bayes rule	75.9	Ref. [13]
MLP with backpropagation	71.5	Ref. [13]
AQ15	66-72	Ref. [14]
Weighted network	68-73.5	Ref. [15]
Default class	70.3	
LERS (rough sets)	67.1	Ref. [11]
k-NN, k=1	65.3	Ref. [13]

TABLE II

10-fold cross validation results for the appendicitis data.

Dataset and method	Accuracy
S-MLP	89.7
PVM, C-MLP2LN (logical rules)	89.6
RIAC (prob. inductive)	86.9
MLP+backpropagation	85.8
CART, C4.5 (dec. trees)	84.9
Bayes rule (statistical)	83.0

Iris-versicolor class that are very similar to the Iris-virginica samples.

The Ljubljana cancer data [9] contains 286 cases, of which 201 are no-recurrence-events (70.3%) and 85 are recurrence-events (29.7%). The data is already discretized, there are 9 attributes with 2-13 different values each. This is a difficult and noisy data. S-MLP found a solution with 47 errors, or 83.6% accuracy, which is the best result we know of. Results are summarized in Table I.

The appendicitis dataset contains only 106 cases, with 8 attributes (results of medical tests), and 2 classes: 88 cases with acute appendicitis and 18 cases with other problems. S-MLP gives similar results as the best logical rules and significantly better than MLP results obtained by Weiss [13]. Results are summarized in Table II.

B. Rule extraction

For the Iris three simple rules giving 4 errors, or 97.3% overall accuracy, were found:

R_1: If F4 $\leq$ 0.9 THEN Iris-setosa.
R_2: If F3 > 4.9 THEN Iris-virginica.
R_3: If F4 > 1.7 THEN Iris-virginica.
R_4: Else Iris-versicolor.

These rules are optimal from the generalization point of view and have been found by us using MLP2LN method [6], [16]. For the Ljubljana breast cancer the two-stage search procedure has been used, with selection of the eight best weights changes followed by evaluation of all possible combination of changes for these weights. Two logical rules for the 'recurrence-event' class were found:

R_1: Involved nodes$\neq$0-2 $\wedge$ Degree malignant = 3 $\wedge$ Tumor-size $\neq$45-49
R_2: involved nodes=[9,11] $\wedge$ tumor size = [35-39]

with ELSE condition for the no-recurrence, giving 64 errors or overall 77.6% of accuracy. A simplification of these rules:

Involved nodes$\neq$0-2 $\wedge$ Degree malignant = 3

with ELSE condition for the second class gives over 77% accuracy in cross validation tests. This rule is easy to interpret: the number of involved nodes is bigger than 2 and the cells are highly malignant. Slightly more accurate rules are obtained with better search strategy (two weights changed at a time):

R_1: Involved nodes$\neq$0-2 $\wedge$ Degree malignant = 3 $\wedge$ Tumor-size $\neq$45-49 $\wedge$ age $\neq$ 10-19
R_2: Involved nodes=[9,11] $\wedge$ age $\neq$ 40-49
R_3: Tumor size = [35-39] $\wedge$ age=30-39

with ELSE condition for the second class. These rules make 62 errors (78.3% accuracy). Rule accuracy and complexity depend on which class is the default one. For 'no-recurrence-events' class S-MLP finds more accurate rules (59 errors, 79.4% accuracy), however they are even more complex. Three rules given above are already too specialized, since the dataset is so small and rules cover only a few cases, capturing accidental correlations. It would be hard to improve these results – most methods give significantly lower accuracies. For example LERS, a machine learning technique based on rough sets, gives after optimization almost 100 rules achieving only 69.4% of accuracy (below the default rate, Table I).

For appendicitis a single rule gives 92.5% accuracy:

R_1: NOT(MBAP$\in$(12.6,18.9] $\vee$ MBAP > 25.2) $\wedge$ MNEA$\leq$6650 $\wedge$ NOT WBC1$\in$ (16775,17400]
Additional rule covers two more cases

R_2: NOT MNEP$\in$(71.6,82.4] $\wedge$ MBAP$\in$(1,3]$\wedge$ WBC1> 9525
giving together with the first rule 94.3% accuracy, but for such small dataset it may be just an accidental correlation.

IV. SUMMARY

Heuristic search procedures may effectively replace backpropagation algorithm with gradient-based minimization of the cost functions. Our preliminary results seem to confirm that the method is fast, accurate, useful for classification and it is a big step towards fully automatic logical rule extraction from raw data.

Acknowledgement: We would like to thank the Polish Committee for Scientific Research, grant no. 8T11F 014 14 for partial support.

References

[1] W. Duch and R. Adamczak, Statistical methods for construction of neural networks. *International Congress on Neural Information Processing*, Kitakyushu, Japan, Oct. 1998, pp. 629-642.

[2] W. Duch, J. Korczak, Optimization and global minimization methods suitable for neural networks. *Neural Computing Surveys* (submitted).

[3] W. Duch, R. Adamczak and N. Jankowski, Initialization and optimization of multilayered perceptrons. In *Third Conference on Neural Networks and Their Applications*, pp. 99-104, Kule, Poland, Oct. 1997; Initialization of adaptive parameters in density networks, ibid, pp. 105-110.

[4] W. Duch, Alternatives to gradient-based neural training. In *Fourth Conf. on Neural Networks and Their Applications*, this volume, Zakopane, Poland, May 1999.

[5] L. Kanal, V. Kumar (Eds), Search in Artificial Intelligence (Springer Verlag 1988)

[6] W. Duch, R. Adamczak and K. Grąbczewski, Extraction of logical rules from backpropagation networks. *Neural Processing Letters* 7: 1-9, 1998.

[7] L. Breiman, Bagging predictors. *Machine Learning* 26 (1996) 123-140

[8] T. Dietterich, An experimental comparison of three methods for constructing ensembles of decission trees: bagging, boosting and randomization. *Machine Learning* 28 (1998) 1-22

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

[10] W. Duch, R. Adamczak, K. Grąbczewski, G. Żal, Methodology of extraction, optimization and application of logical rules. Intelligent Information Systems VIII, Ustroń, Poland, June 1998 (in print)

[11] J.W. Grzymała-Busse, T. Soe, Inducing simpler rules from reduced data. Intelligent Information Systems VII, Malbork, Poland, 15-19.06.1998, pp. 371-378

[12] G. Cestnik, I. Konenenko, I. Bratko, Assistant-86: A Knowledge-Elicitation Tool for Sophisticated Users. In: I. Bratko and N. Lavrac (Eds.) *Progress in Machine Learning*, pp. 31-45, Sigma Press 1987

[13] 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

[14] R.S. Michalski, I. Mozetic, J. Hong, N. Lavrac, The Multi-Purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains. In: Proc. 5th National Conf. on AI, pp. 1041-1045, Philadelphia, PA: Morgan Kaufmann 1986

[15] M. Tan, L. Eshelman, Using weighted networks to represent classification knowledge in noisy domains. Proc. 5th Intern. Conf. on Machine Learning, pp. 121-134, Ann Arbor, MI 1988

[16] W. Duch, R. Adamczak, K. Grąbczewski, Extraction of crisp logical rules using constrained backpropagation networks. *ICANN'97*, Houston, 9-12.6.1997, pp. 2384-2389, Logical rules for classification of medical data using ontogenic neural algorithm, *EANN'97*, Stockholm, 16-18.06.1997, pp. 199-202