ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 109-114

Extraction of crisp logical rules using constrained backpropagation networks

Włodzisław Duch[a,b], Rafał Adamczak[b] and Krzysztof Grąbczewski[b], Masumi Ishikawa[a] and Hiroki Ueda[a]

[a]Department of Control Engineering and Science, Kyushu Institute of Technology, 680-4 Kawazu, Iizuka, Fukuoka 820, Japan

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

Abstract. Two recently developed methods for extraction of crisp logical rules from neural networks trained with backpropagation algorithm are compared. Both methods impose constraints on the structure of the network by adding regularization terms to the error function. Networks with minimal number of connections are created, leading to a small number of crisp logical rules. The two methods are compared on the Iris and mushroom classification problems, generating the simplest logical description of this data published so far.

1. Introduction

Knowledge acquisition by extraction of logical rules from the sample data is an important and difficult problem in computational intelligence. Neural networks, in particular multi-layered perceptrons (MLPs), are useful classifiers that can learn arbitrary vector mappings from the input to the output space and successfully use this mapping in novel situations. The knowledge acquired by neural systems is represented in a set of numerical parameters and architectures of these networks in a way that is incomprehensible for humans. Some classification problems have an inherent logical structure and even in other cases it may be preferable to use logical rules instead of adaptive classifiers.

Many methods to analyze trained neural networks, extract logical rules and select classification features have been devised in the past (for a recent review see [1]). These methods focus on analysis of parameters (weights) of trained networks, trying to achieve high fidelity of performance, i.e. obtaining identical classification results by extracted logical rules in comparison to the original networks. Analysis of complex networks is quite difficult and may lead

W.D. is grateful for support by the Polish Committee for Scientific Research, grant 8T11F 00308, and Heiwa Nakajima Foundation, Japan. M.I. acknowledges the support by Grant-in-aid for Scientific Research (c) 07680404 from the Ministry of Education, Japan. E-mails: ishikawa@kyutech.ac.jp, duch,raad,kgrabcze@phys.uni.torun.pl

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 109-114

to a large number of rules, too large to be useful in practice. Non-standard form of rules, such as $MofN$ (M out of N antecedents should be true) or decision trees [2], are sometimes useful.

In this paper we take a different approach, simplifying the network structure to the point when the logical functions performed by the network are quite clear. In the next section two new methods of rule extraction are presented. In the following sections they are compared on the iris and mushroom classification problems.

2. Presentation of algorithms

Logical rules require symbolic inputs (linguistic variables), therefore the continuos input data has to be quantized first, i.e. the features defining the problem should be identified and labeled. The problem of optimal selection of input features is very important. So far we have found two solutions to this problem. First, neural networks that use factorizable transfer functions, such as the Feature Space Mapping (FSM) networks [3], may be treated as neurofuzzy systems and their output interpreted in a fuzzy logic sense using membership functions. In such cases rules are of the type:

$$\text{IF } \left(x_1 \in X_1 \wedge ... x_N \in X_N\right) \text{THEN } \left(y_1 \in Y_1 \wedge ... y_M \in Y_M\right) \qquad (1)$$

Crisp decision regions may be obtained in an adaptive way by using as the neuron processing function a pure product form of sigmoidal functions $\prod_i \sigma(x_i - b_i)(1 - \sigma(x_i + b_i'))$, product of differences $\prod_i(\sigma(x_i - b_i) - \sigma(x_i + b_i'))$ or a filtered combination of differences $\sigma\left(\sum_i(\sigma(x_i - b_i) - \sigma(x_i + b_i')) - B\right)$ [4] and slowly increasing the gain of the sigmoidal functions $\sigma(x)$ during learning. In this process fuzzy rules are transformed into crisp logic rules, i.e. complex decision regions are transformed into simpler, hypercuboidal decision regions.

We will not pursue this further here, concentrating instead on the extraction of crisp logical rules using standard multilayered perceptrons. Interpretation of the activation of the MLP network nodes trained on the classification problem is not easy since the resulting networks are rather complex. In the structural learning with forgetting (SLF) approach Ishikawa [5] has used the Laplace-type regularizing term:

$$E(W) = E_0(W) + \lambda \sum_{i,j} |W_{ij}| \qquad (2)$$

where $E_0(W)$ is the mean square error (MSE) criterion function and W_{ij} is the connection weight between units i and j. The additional change in the weights due to the regularizing term during backpropagation learning is simply equal to $-\epsilon\ \mathrm{sgn}(W_{ij})$, where a constant ϵ determines the amount of forgetting. In selective forgetting procedure only the weights smaller than some threshold are included in the regularizing term. This term leads to a constant decay of smaller weights. Small weights that do not contribute to the MSE are automatically pruned and a skeletal network emerges.

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 109-114

To facilitate logical interpretation of the function performed by the network "clarification" of hidden unit performance is done by forcing them to be fully active or fully inactive. This is achieved by adding an additional penalty term $c\sum_i \min(1-h_i, h_i)$, where h_i is the output of the hidden unit i. The SLF procedure can be applied to rule extraction in a series of successive steps, starting from rather large regularization parameter λ to acquire dominant rules first. While these rules, or corresponding part of the network, are kept fixed, regularization parameter is decreased. Connections once deleted revive and new rules are derived. Since the network has only skeletal structure and the hidden units have outputs close to 0 or 1 each node is represented as a logical function of nodes in the adjacent lower layer. These logical functions are combined into a final logical output expression.

The second approach considered here [7] is aimed directly at making a smooth transition from MLP to a logical network, therefore we will called it here MLP2LN. This is achieved by: a) increasing the slope of sigmoid functions to obtain crisp decision regions; b) simplifying the network structure by inducing the weight decay through a gaussian penalty term; c) enforcing the integer weight values 0 and ± 1, interpreted as 0 = irrelevant input, $+1$ = positive and -1 = negative evidence. The error function has two extra terms:

$$E(W) = E_0(W) + \frac{\lambda_1}{2}\sum_{i,j} W_{ij}^2 + \frac{\lambda_2}{2}\sum_{i,j} W_{ij}^2 (W_{ij}-1)^2 (W_{ij}+1)^2 \qquad (3)$$

The first term, scaled by λ_1 hyperparameter, encourages weight decay, leading to skeletonization of the network and elimination of irrelevant features. The second term, scaled by λ_2, forces the remaining weights to approach ± 1, facilitating logical interpretation of the network function. Additional change of weights in the backpropagation training algorithm due to these terms is $\lambda_1 W_{ij} + \lambda_2 W_{ij}(W_{ij}^2-1)(3W_{ij}^2-1)$.

This approach may be justified from the Bayesian point of view [8]. The cost function specifies our prior knowledge about the probability distribution $P(W|M)$ of the weights in our model M. Since we model a network for classification tasks and expect crisp logical decision weights of connections for positive evidence should be $+1$, for negative -1 and for irrelevant inputs 0, therefore $P(W|M) \propto \exp(-\alpha E(W|M)) = \exp(-\alpha_1 W_{ij}^2)\exp(-\lambda_2 W_{ij}^2 (W_{ij}-1)^2 (W_{ij}+1)^2)$.

Although MLP2LN method may be applied to any MLP network a simplified constructive procedure is recommended. The training proceeds separately for each output class. One hidden neuron is created and is trained on the input data by backpropagation procedure until convergence is achieved. The weights and the threshold obtained are then analyzed and the first group of logical rules is found, covering the most common input-output relations. The input data that is correctly handled by the first neuron will not contribute to the error function. Therefore the weights of this neuron are kept frozen during further training and a second neuron is trained on the remaining data. After convergence the second weight vector is analyzed and corresponding rules found.

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 109-114

This procedure is repeated until all the data are correctly classified. The same procedure is repeated for the remaining classes. Each time only one neuron is trained, therefore the training is very fast.

The two approaches to logical rule extraction share some similarities. In both cases the skeletonization of the network is stressed by adding the Laplacian or Gaussian regularization term. Clarification of the hidden units in SLF plays similar role to increasing the slopes in MLP2LN, providing only 0 or 1 outputs from the hidden units. In SLF with scuccessive regularization and MLP2LN after the most common rules, obtained for large regularization parameters, are found, a part of network is kept frozen and new hidden neurons are trained to obtain more rules. The rules obtained by these algorithms are ordered, starting with rules that are used most often and ending with rules that handle only a few cases. The final solution may be presented as a set of rules or as a network of nodes performing logical functions. Direct logical interpretation of MLP2LN networks is somewhat easier because the weights are constrained to ± 1. In the following two sections empirical comparison of the results of these two approaches applied to two classification problems is presented.

3. Classification of iris flowers

In the first example the classical iris dataset was used. The data has 150 vectors evenly distributed in three classes, called iris-setosa, iris-versicolor and iris-virginica. Each vector has four features: sepal length x_1 and width x_2, and petal length x_3 and width x_4 (all in cm). This data was used to train SLF network while for the MLP2LN linguistic variables obtained from analysis of histograms of the individual features for each class were used. For example, Iris-virginica class is more frequent for the value of x_3 above 4.93 and Iris-versicolor are more frequent below this value. Discretization based on histograms leads to the linguistic variables presented in Table 1. With this discretization of the input features two vectors of the iris-versicolor class (coded as (m, m, l, l) and (m, l, m, l)) become identical with a number of iris-virginica vectors and cannot be classified correctly, i.e. the best classification accuracy is 98.7%.

Table 1: Linguistic variables obtained by analysis of histograms.

	s	m	l
x_1	[4.3,5.5]	(5.5,6.1]	(6.1,7.9]
x_2	[2.0,2,75]	(2.75,3.2]	(3.2,4.4]
x_3	[1.0,2.0]	(2.0,4.93]	(4.93,6.9]
x_4	[0.1,0.6]	(0.6,1.7]	(1.7,2.5]

Thus initial MLP2LN network had 12 discrete input nodes, while SLF network 4 real-valued input nodes. The MLP2LN network needed about 1000 epochs on average and the final weights were within 0.05 from the desired ± 1

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 109-114

or 0 values. Only two features, x_3 and x_4 were found to be relevant and a single rule per class was found:

$$\begin{array}{ll} \text{IF} & (x_3 = s \wedge x_4 = s) \text{ THEN iris-setosa} \\ \text{IF} & (x_3 = m \wedge x_4 = m) \text{ THEN iris-versicolor} \\ \text{IF} & (x_3 = l) \vee (x_4 = l) \text{ THEN iris-virginica} \end{array} \quad (4)$$

These rules allow for correct classification of the 147 vectors, achieving 98% of accuracy. Replacing one of the rules with the condition ELSE, and noting that for the iris-setosa one may remove one antecedent without changing classification results one gets two rules: iris setosa if $x_3 = s$, iris virginica if $x_3 = l \vee x_4 = l$, else iris versicolor. Decreasing regularization parameters allows to replace one rule by four rules necessary to classify correctly just one additional vector, a clear indication that overfitting occurs. Increasing regularization parameters selects only one attribute, petal length x_3, and two rules giving 95.3% accuracy: iris setosa if $x_3 < 2.5$, iris virginica if $x_3 > 4.9$, else iris versicolor. This is the simplest description of the Iris dataset that we know of.

4. Classification of mushrooms

The mushroom dataset contains 8124 cases, each with 22 discrete attributes, with about half of the cases (51.8%) representing edible and the rest nonedible (mostly poisonous) mushrooms. A single neuron is capable of learning all the training samples, but the resulting network has many nonzero weights and is difficult to analyze from the logical point of view. The following single rule has been obtained with MLP2LN as well as SLF method, giving 48 errors, or 99.41% accuracy on the whole dataset:

edible if odor=(almond∨anise∨none)∧ spore-print-color=¬green

This rule uses only two features and four antecedents. Using weaker regularization parameters for edible mushrooms SLF has discovered (after some simplifications of resulting logical expressions) a conjunctive rule with 6 attributes, achieving perfect accuracy. MLP2LN has discovered systematically equivalent disjunctive rules for poisonous mushrooms:

R_1) odor=¬(almond∨anise∨none), 120 poisonous cases missed, 98.52%
R_2) spore-print-color=green, 48 cases missed, 99.41% correct
R_3) odor=none∧stalk-surface-below-ring=scaly∧ (stalk-color-above-ring= ¬brown), 8 cases missed, 99.90%
R_4) habitat=leaves∧cap-color=white, all poisonous cases correctly classified.

We have tried training on randomly sampled 10% of the database as well as on the whole data, achieving identical results. This is the simplest systematic logical description of this dataset that we know of, although some of these rules have probably also been found by RULEX and TREX algorithms [1].

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 109-114

5. Discussion and summary

The problem of extracting rules from neural networks has a natural geometrical interpretation. Crisp logical rules correspond to a division of the input space with perpendicular hyperplanes into areas with symbolic names. This may be achieved in MLP networks by using neurons with high gain and ± 1 or zero weights. If the classes in the input space are correctly separated with such hyperplanes logical description of the data is possible. Logical approximation may become arbitrarily accurate by increasing the number of linguistic variables, but the number of rules may become unacceptably large.

We have presented here two methods of rule extraction based on the standard backpropagation technique with modified error function. Crisp logical rules are found automatically by analyzing nodes of trained networks. These methods seem to outperform in many ways previous methods of rule extraction [1]. It is too early to tell which of these two methods will be more useful for larger datasets, but both seem to be capable of finding the simplest logical structure in some of the benchmark datasets used for neural network testing.

References

[1] R. Andrews, J. Diederich, A.B. Tickle, A Survey and Critique of Techniques for Extracting Rules from Trained Artificial Neural Networks, *Knowledge-Based Systems* 8 (1995) 373–389

[2] M.W. Craven, J. W. Shavlik, Extracting Tree-Structured Representations of Trained Networks, *Adv. in Neural Info. Processing* 8 (1996) 24–30

[3] W. Duch, G.H.F. Diercksen, Feature Space Mapping as a universal adaptive system, *Computer Physics Communic.* 87 (1995) 341–371

[4] W. Duch, N. Jankowski, Bi-radial transfer functions, *Proc. second conference on neural networks and their applications*, Orle Gniazdo, Poland, pp. 131–137, 1996; W. Duch, R. Adamczak, K. Grąbczewski, Constrained backpropagation for feature selection and extraction of logical rules, *Proc. of "Colloquiua in AI"*, Łódź, Poland 1996, p. xxx

[5] M. Ishikawa, Structural learning with forgetting, *Neural Networks* 9 (1996) 509–521

[6] M. Ishikawa, Rule extraction by successive regularization, in: *Proc. of the 1996 IEEE ICNN*, Washington, June 1996, pp. 1139–1143.

[7] W. Duch, R. Adamczak and K. Grąbczewski, Extraction of logical rules from training data using backpropagation networks, *The First Online Workshop on Soft Computing*, 19-30.Aug.1996; http://www.bioele.nuee.nagoya-u.ac.jp/wsc1/

[8] D.J. MacKay, A practical Bayesian framework for backpropagation networks, *Neural Computations* 4 (1992) 448–472