Rule extraction from the Artificial Neural Network

D V Marshakov

Department of Computing systems and information security, Don State Technical University, Rostov-on-Don, Russia

Abstract. Applying machine learning techniques to various critical decision-making domains implies the need to explain why a machine learning algorithm makes certain conclusions. Artificial neural networks (ANNs) have achieved high classification accuracy for many classification problems, but their results can’t be interpreted, which is why ANNs are often considered as «black box». To interpret the results of neural network classification, the paper presents a method for extracting rules from ANN. The proposed method is based on the structuring of information flows processed in the ANN information field through transforming complex multidimensional data into a simpler structure of a lower dimension, followed by a trivial transformation of the results into a set of fuzzy rules of a certain type. The implementation of the proposed method provides for the construction of the hybrid ANN configuration based on self-organizing and multilayer ANNs. The results of an experimental research of the proposed method are presented on the example of solving the well-known problem of multiparameter classification. The results obtained confirm the adequacy of the proposed method, which can be used both in independent neural network pattern recognition systems and within decision support systems.

1. Introduction

At the present stage, machine learning technology uses a wide variety of algorithms to transform datasets into predictive models, ranging in complexity from linear regression and logistic regression to deep neural networks and ensembles.

Among the known types of machine learning, a special place is occupied by artificial neural networks (ANNs), which modeling with the artificial neurons the work of the human brain mind, solving a specific problem, with the possibility of learning from experience. The computational capabilities of ANNs are best suited for pattern recognition tasks, such as image classification and speech recognition that are not available for simpler algorithms.

In the process of training the ANN, the generalized features of the input patterns are distributed in the inter-neural connections of the network with the produce of an implicit algorithm for solving the problem that provides high accuracy for a variety of classification problems with huge datasets. In this case, the neural network approach is identified with the «black box» method, which implies the creation of a simulation model without formulating the rules for neural network decision-making. The process of forming the final result remains hidden and «opaque» for the user due to the impossibility of substantiating the generated responses to the values of the input signals by the neural network. The problem of interpretability of the ANN output results remains relevant in many critical decision-making domains [1,2].

For an ANN two types of interpretability are possible: global and local [3]. With global interpretability users can understand how a model works globally by inspecting the structures and parameters of a complex model, which is achieved by understanding the representations captured by
neurons at an intermediate level. Local interpretability examines each conclusion of the ANN, trying to identify the reason for its adoption by identifying the contribution of each feature of the input data, corresponding to it. Global interpretability explains to some extent the internal working mechanisms of machine learning models, providing them transparent. Local interpretability helps to uncover causal relationships between a particular input signal and the corresponding model prediction. Together, these two approaches increase user confidence in the model and the final forecast result.

The solution of the problem of interpretability of the ANN results can be carried out by extracting the rules formed in the learning process, which implies the process of developing a syntax similar to natural language that describes the behavior of a neural network. In this case, rule extraction is usually based on sample learning of ANN using classification methods to obtain classification rules [4]. The rule extraction algorithms are basically based on analyzing the ANN structure and/or training data content and reduction of neural network for each input in search of the conditions that make up the rules.

Of the existing methods of rule extraction (pedagogical, decompositional, eclectic), the article focuses mainly on the decompositional method, which is based on the analyzing of activation and weights of hidden layers of a neural network to extract the symbolic rules of the internal algorithm of the ANN.

One of the stage to extract an explicit algorithm for neural network problem solving is to use the procedure for contrasting (reduction) of ANN to «logical transparency» and then verbalize it (verbalization) in the form of a symptom-syndromic structure and explicit formulas for the formation of output values of neurons (syndromes) from input ANN values (symptoms) [5,6]. However, the contrast of ANN is reasonably only for particular problems and is difficult if used regardless of the problem being solved. In addition, the pruning of the structure of the neural network leads to the loss of its structural redundancy, which can lead to a significant decrease in the degree of fault tolerance of the ANN in the case of its hardware implementation.

Methods of converting neural network algorithms to the form of production rules of the form IF-THEN [7,8] are widely used. They are used for constructing neuro-fuzzy models, as well as hybrid models using ANN and expert system technologies (neuro-expert systems). In the case of neuro-fuzzy models, the system becomes more «transparent» for the analysis of the structure of connections, but this also reduces the redundancy of the information field of the neural network system. For neuro-expert systems, activation of a neural network knowledge base is similar to extracting rules from the information field of a neural network, which usually requires high computational complexity [9].

In practice, there are also known cases of using a neural network knowledge base without adapting it to the system of rules, when the output values of the neural network are accepted as the initial premises for the inference mechanism and the ready base of rule the expert system. However, in this case, additional mechanisms for evaluating the output states of the ANN and their interpretation are required.

2. Problem statement

The purpose of this work is to develop a method for extracting rules from the ANN at the stage of its operation for interpreting classification results obtained using neural networks under specified input influences.

The solution to this problem is to establish a cause-and-effect relationship between a specific set of input features and the generated output solution of the ANN based on the analysis of the internal algorithm of the ANN functioning. The set of input features, which in some cases is multidimensional data, undergoes multiple changes in the process of neural network parallel processing, being distributed within the adaptive information field of weight coefficients of neurons in the ANN. At the same time, according to the author, the problem of interpretation may be partly related to the lack of information markup inside the inter-neuron connections. This markup is possible by combining intermediate data into groups, i.e. their cluster analysis. Clusterization of information implies that if there is one or more representatives within a group that have certain a priori information (markup), it can be extended to all elements of this group.
When recognizing patterns among sets of similar features, simulation of such processes seems appropriate with the use of intelligent methods for processing subjective information that use the characteristic features of recognized objects. One of the approaches to constructing an algorithm for solving such a problem is to generalize various types of information with the application of empirical experience to it, to identify a number of characteristic features of an object, on the basis of which this object is assigned to one of the possible classes. This approach may be closest to the processes of modeling human reasoning [10], in which, through observation for the surrounding world, certain entities (concepts) are identified and certain relationships are established between them, usually based on similarities and differences of the observed situations.

The implementation of this approach is possible through the combined use of self-organizing and multilayer ANNs with fuzzy modeling methods at the stage of interpreting intermediate and output results.

At the same time, cluster analysis provides the division of a set of signals within layers into similar groups (subsets) of individual parameters. An ANN trained to activity with these subsets assigns the specified set of input parameters to one of the output classes defined at the training stage. Fuzzy modeling allows you to describe the selected subsets as fuzzy rules for formalizing the decision formation process and further interpretation.

3. Methodology
A hybrid ANN configuration is proposed. It consists of adding clustering layers between the layers of a perceptron-type ANN (perceptron neural network) that are pre-trained to divide the input set of signals describing the selected features of the object into similar groups (subsets) of individual input parameters. The proposed ANN configuration includes a cascading connection of layers: the input layer, the S layers of neural network information processing, and the output layer. The input layer retransmits input signals to the $l = 1$ ($l = 1, S$) ANN layer, the output layer is used for interpreting the solutions formed at the ANN output. Each internal ANN layer consists of a combination of the clustering layer and the perceptron layer of the neural network.

The clustering layer can be a set of self-organizing Kohonen networks trained by the winner-takes-all algorithm (WTA). The number of self-organizing networks in the layer of the proposed ANN configuration corresponds to the number of input signals of the considered layer.

This cascading connection of self-organizing layers of Kohonen and ANN of the perceptron type allows combining the ability of the former to localize and the approximation capabilities inherent in multilayer ANN.

Let us consider the mathematical model of the proposed configuration on the example of single layer ($l = 1$), the input value of which is the input vector $X = [x_0, x_1, ..., x_N]^T$, where $x_0 = 1$ defines a signal of bias. The clustering layer consists of $N$ self-organizing networks that perform segmentation of input data features. The perceptron layer consists of $H$ sigmoid neurons. The symbol is entered for each self-organizing layer network:

$$\ldots, CBA = \Omega \ldots$$

etc. From the preliminary analysis of the training sample set the number of $m$ possible clusters for data received at each of the ANN inputs is determined. The outputs of the corresponding self-organizing networks corresponding to the selected clusters are indicated $\sigma_k \in \Omega$, where $k = 1, M$ is the cluster number for each $i$-th input, and $M(i)$ is the number of clusters depending on the input number, set at the stage of forming the layer configuration.

Each $k$-th neuron of the active layer of the $i$-th self-organizing network has its own weight $w_{ik}^{(\Omega)}$, which is compared with the input value of $x_i$ by calculating the distance between them according to the equation (1).

$$k = \arg\min_{\tilde{k}} \|x_i - w_{\tilde{k}}^{(\Omega)}\|,$$  \hspace{1cm} (1)
where: \( x_i \in X \) is component of the input vector \( X \); \( k \) is the number of the winning neuron whose weight has the smallest distance to the input signal value; \( w^{(ii)}_{ik} \) is the weight coefficient of the self-organizing network that connects the \( i \)-th input with the \( k \)-th output.

The Euclidean distance [11] can be used as a metric:

\[
d(x, w^{(ii)}_{ik}) = \sqrt{\sum_{i=1}^{N} (x_i - w^{(ii)}_{ik})^2}.
\]

For input data features localized using the \( l \)-th clustering layer, the corresponding weight coefficients of the \( l \)-th perceptron layer are formed. The result of calculations of this layer is a vector of output signals \( Z = [z_1, z_2, \ldots, z_H]^{T} \). Each component of this vector represents the output signal of the \( j \)-th neuron \( j = 1, H \), described by the following function

\[
z_j = f\left(\sum_{i=1}^{N} w^{(ii)}_{ij} |\sigma_k| x_i\right),
\]

where \( w^{(ii)}_{ij} |\sigma_k| \) are the weight coefficients corresponding to the output of the \( \sigma \)-th self-organizing network.

Here is the learning process of the ANN of the proposed configuration for \( S = 2 \). Learning is carried out in layers, sequentially for each alternating algorithms for unsupervised learning WTA for clustering layers and supervised learning with the backpropagation method for perceptron layers of ANN.

At the beginning, \( l = 1 \) clustering layer is trained on the set of components of vectors input data, namely, is trained each self-organizing network connected to the corresponding input. The number of clusters for each input is determined experimentally, based on the degeneration of additional clusters. As a result, the neurons of this layer are organized in such a way that their weights are best used by the distribution of the training vector data. For the winner neuron of each network \( \Omega = A, B, C, \ldots \) set of weight coefficients is formed that connect its output with the input of the \( j \)-th neuron \( l = 1 \) of the perceptron layer of the ANN.

Then, based on the markup of the input data, the \( l = 1 \) and \( l = 2 \) perceptron layers of the ANN are trained with the backpropagation.

Next, each self-organizing network \( l = 2 \) clustering layers is trained on the set of components of the output data vectors \( l = 1 \) of the perceptron layer. Similarly to the first layer, the set of weight coefficients is formed for the winning neurons of each self-organizing network, connecting their output with the input of the \( j \)-th neuron \( l = 2 \) of the perceptron layer of the ANN.

At the final stage, a separate training of the \( l = 2 \) perceptron layer of the ANN is performed, the output values of which are received in the output layer. They are a generated neural network solution.

The procedure for extracting rules from the ANN consists of a layer-by-layer analysis of the processed input data with registration of cluster numbers indicating the properties of signals generated by the neural network.

4. Illustrative example
The practical applicability of the proposed method is illustrated on the example of the classic classification problem «Fisher’s Iris Database», the purpose of which is to classify iris sorts based on measurement data. The data set contains 150 samples belonging to three different classes of irises (setosa, virginica, versicolor), 50 samples of each class. For each instance, there are four characteristics that describe the geometric features of recognized objects: sepal length, sepal width, petal length and petal width.

The set of initial data was divided into two components: 80% was included in the training sample, and 20% in the test sample.

The object of research is a two-layer feedforward neural network with two neurons in the first and three neurons in the second layer (Figure 1). The number of inputs in the input layer corresponds to the number of input features and is 4, the number of outputs in the output layer is 3. Sigmoidal activation
functions are used as the activation function of each of first-layer neurons. For neurons in the second layer are used competing the activation function softmax, that allows to interpret the outputs as the probability of assigning an object to each of the classes.

For sets values of input features in this example uses the classic min-max normalization for the entire data sample.

Figure 1. Configuration of the ANN

In clustering layers, classic A-F Kohonen networks are used for each of the outputs of the previous layer, which are trained using the WTA algorithm.

Based on the preliminary analysis of the initial data, the Kohonen networks (A, B, C, D) in the first clustering layer are trained to cluster the input sets of each input into 3 groups of features, which conditionally corresponds to the geometric dimensions of the sepals and petals: «small», «medium» and «large». The clusterization results for each of the input features are shown in Figure 2.

The Kohonen networks (E, F) in the second clustering layer are trained to cluster data from the outputs of neurons of the first neural network layer into 2 groups. The number of groups in this case is determined experimentally, based on the degeneration of additional groups in a number of numerous examples of the problem being solved. The symbols of the clusters can be set arbitrarily. In this case, the notation based on the numerical value of the outputs of neurons: «low value» and «high value». The clustering results for the specific example below are shown in Figure 3.
Figure 2. Clusterization results for the input set of characteristics:
   a) $x_1$ – the length of the sepal; b) $x_2$ – the width of the sepal;
   c) $x_3$ – the petal length; d) $x_4$ – the petal width

Figure 3. Results of clusterization of the data set of the first neural network layer:
   a) $z_1$ – neuron $1^{(1)}$; b) $z_2$ – neuron $2^{(1)}$
ANN training is performed using the above algorithm. One of the variants of the matrix of weights of neural network layers is shown in Table 1 and Table 2.

**Table 1. Matrix of weights of the first neural network layer – \( w^{(1)}_i \)**

| \( i \) = 0 | \( i \) = 1 | \( i \) = 2 | \( i \) = 3 | \( i \) = 4 |
|---|---|---|---|---|
| \( j = 1 \) | -0.45 | -11.7 | -11.6 | 2.22 | -18.3 | -20.6 | 24.0 | 0.42 | 6.74 | 5.61 | 0.31 | 1.94 |
| \( j = 2 \) | -1.31 | -1.97 | 2.56 | -0.64 | -5.30 | 3.92 | -4.68 | 9.13 | -10.9 | 6.22 | -7.27 | -1.66 | 12.9 |

**Table 2. Matrix of weights of the second neural network layer – \( w^{(2)}_i \)**

| \( i \) = 0 | \( i \) = 1 | \( i \) = 2 |
|---|---|---|
| \( j = 1 \) | 6.02 | -15.20 | 0.01 | -13.16 | -15.47 |
| \( j = 2 \) | -5.38 | -38.62 | 0.36 | 24.98 | 13.59 |
| \( j = 3 \) | -9.90 | 36.41 | 0.04 | -8.20 | 3.50 |

Consider the principle of extracting rules from ANN, on the example certain input feature vector from the test set: length of sepal – 6.2 cm, width of sepal – 2.9 cm, length of petal – 4.3 cm, width of petal – 1.3 cm, which correspond to the second class iris – *virginica*. After normalization the input feature vector takes the form: \( X = {0.7820, 0.3589, 0.5384, 0.1538} \).

The first element of the input vector corresponds to the cluster \( a_1 \in A \) («large»), the second element to the cluster \( a_2 \in A \) («medium»), the third element to the cluster («medium»), and the fourth element to the cluster \( d_1 \in D \) («medium»).

The output value of the first layer is the vector \( Z = {8.34 \cdot 10^{-6}, 0.9799} \), the first element of which corresponds to the cluster \( e_1 \in E \) («low value»), according to the results of the second clustering layer, and the second element to the cluster \( f_1 \in F \) («high value»).

The result of calculations of the second layer is the vector \( Y = {0, 1, 0} \), which corresponds to the maximum probability of assigning the input vector to the second class.

Based on the calculations performed from the ANN, its output results are interpreted by constructing a chain of interrelated rules:

**RULE 1:** IF \( x_1 \) is \( a_1 \) AND \( x_2 \) is \( b_2 \) AND \( x_3 \) is \( c_2 \) AND \( x_4 \) is \( d_2 \) THEN \( z_1 \) is \( e_2 \) AND \( z_2 \) is \( f_1 \)

**RULE 2:** IF \( z_1 \) is \( e_2 \) AND \( z_2 \) is \( f_1 \) THEN \( y_2 \)

It should be noted that in this case the implementation of RULE 2 can have different combinations of preconditions, why when constructing a chain of retrospective reasoning to interpret the output result ANN it can use only subject to RULE 1: RULE 2 ← 1 RULE.

To assess the adequacy of the proposed ANN model a comparative analysis of the results of various ANN variants synthesized by the proposed algorithm and the classical multilayer feedforward ANN was performed.

The classical multilayer feedforward ANN also had 4 inputs, 2 neurons in the hidden layer and 3 neurons in the output layer, sigmoidal activation functions were used. The training method was backpropagation with the gradient descent with momentum and adaptive learning rate. The training sample was also 80% of the initial data set, and the test sample was 20%.
Figure 4 shows the receiver performance curves (ROC) for a combination of training and test data sets for the above ANN example and for the classical multilayer feedforward ANN described configuration.

![Figure 4. ROC curves for a combination of training and test samples of ANN: a) the proposed configuration; b) the classic configuration](image)

As can be seen from the results, the proposed ANN model doesn’t make contradictions in the process of functioning and evaluating the quality of the classification, and therefore it is adequate. In a number of cases, including the one presented in this paper, it is noted that the proposed model shows more accurate classification results. This is due to an increase in the computing capabilities of the ANN due to the use of additional weight coefficients in the composition of artificial neurons, which change depending on the value of the input signal.

5. Conclusions and further work
The proposed method for interpreting the functioning of ANN allows structuring information flows processed in the ANN information field through converting complex multidimensional data into a simpler structure of a smaller dimension, followed by trivial conversion of the results into a set of fuzzy rules of a certain type.

In this research was based on the well-known model of a multilayer perceptron. The proposed modification of its structure with the addition of clustering layers doesn’t lead to contradictory assessments at the stage of functioning and evaluating the quality of the classification. In this case, self-organizing Kohonen networks were used in the clustering method, which have some disadvantages, in particular, the lack of a pre-known number of clusters, as well as possible losses during data compression. It should also be noted that the cluster numbers of this network should be ordered.

Nevertheless, this approach can be applied in the synthesis of ANN for a critical decision-making domains that requires interpretation of output results. This is confirmed by a number of experiments, including the illustrative example of multiparameter classification considered in the paper.

For further research, it seems promising to develop models for analyzing the generated rules, assessing their quality and the possibility of using them in configurations of deep neural networks.

References
[1] Yan A, Chen Z, Wang L, Peng L, Hassan M U and Zhao C 2018 Proc. Int. Conf. on Security, Pattern Analysis, and Cybernetics (New York: IEEE) pp 146–151
[2] Elsarrar O, Darrah M and Devine R 2019 Proc. Int. Conf. on Machine Learning And Applications (New York: IEEE) pp 1917-1923

[3] Du M, Liu N and Hu X 2020 J. Comm. of the ACM 63 68–77

[4] Augasta M G and Kathirvalavakumar T 2012 Proc. Int. Conf. on Pattern Recognition, Informatics and Medical Engineering (New York: IEEE) pp 404–408

[5] Kamruzzaman S M and Hasan A R 2005 Proc. Int. Conf. on Information and Communication Technology in Management (Melaka: Multimedia University) pp 1–14

[6] Zhukov L A and Reshetnikova N V 2004 J. Rad. Elect., Comp. Sci., Cont. 12 112–117 (In Russ.)

[7] Kotenko I V, Shorov A V and Nesteruk P G 2011 J. SPIIRAS Proc. 18 19–73 (In Russ.)

[8] Benitez J M, Castro J L and Requena I 1997 J. IEEE Tran. on Neur. Net. 8 1156–1164

[9] Chorowski J and Zurada J M 2011 J. IEEE Tran. on Neur. Net. 22 2435-2446

[10] Marshakov D V 2014 Proc. Int. Symp. on System analysis, control and information processing (Rostov-on-Don: Don State Technical University) pp 296–299 (In Russ.)

[11] Kohonen T 2001 Self-Organizing Maps (New York: Springer)