Hybrid intelligent system of rules extraction for decision making

A N Averkin¹³ and S A Yarushev²

¹ Scientific Laboratory of Artificial Intelligence, Neuro-technologies and Business Analytics, Plekhanov Russian University of Economics, Moscow, 117997, Russia
² Department of Informatics, Plekhanov Russian University of Economics, Moscow, 117997, Russia
³ Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences

E-mail: averkin2003@inbox.ru

Abstract. This article attempts to give an overview of several algorithms for extracting rules from an artificial neural network. The goal of this article is to find critical links between important parts of artificial intelligence – production models, fuzzy logic and deep learning. Such an approach will stimulate researchers in the field of soft computing to develop applied systems in the field of explanatory artificial intelligence and machine learning.

1. Introduction

This article presents the basic methods of machine learning and explanatory artificial intelligence that can help in the issue of extracting rules and other models of knowledge representation not only from data, but from the artificial neural networks themselves. The paper discusses classification methods for rule-based learning methods for neural networks and the current state of technologies for extracting rules from neural networks. Next, we formulate the main problems that arise when extracting rules from neural networks, as well as the main methods for solving them. A number of rule extraction algorithms are described in detail below. The last part discusses specific issues when working with deep neural networks and neuro-fuzzy systems.

Artificial neural networks are well-known parallel computing models that are highly effective in solving complex artificial intelligence problems such as pattern recognition and text analysis. However, many users are afraid to use them in critical situations due to the fact that they are a "black box". This means that explaining how a neural network makes a particular decision is a very difficult problem.

This is a serious problem because it is difficult to trust solutions to a neural network that solves real problems without the ability to explain the decisions made. This is particularly true for security-critical tasks in which hidden errors can lead to dangerous human consequences or large military, political or economic losses.

Moreover, understanding how neural networks extract, accumulate and modify formal knowledge is important and necessary for the evolution of machine learning methods and explanatory artificial intelligence. For example, increasing the transparency of neural networks reveals “hidden dependencies” that are not present in the input data but appear because of processing by the neural network. To solve these problems of neural networks, data scientists realized the idea of extracting rules
directly from neural networks, which is one of the methods of artificial intelligence. In this way, we establish an additional link between symbolic and connectional (sub-symbolic) models of knowledge representation in artificial intelligence.

Most authors are focused on extracting the most understandable rules, and at the same time they have failed to mimic neural network maintenance as accurately as possible. After the appearance in 1992 in Jang's doctoral dissertation of a method of isomorphic representation of fuzzy rules in the form of a neuro-fuzzy system, tremendous work was done in this area, which ended with the creation of the directions of soft computing and computational artificial intelligence. Since then, many methods for extracting rules from neural networks have been developed and critically investigated, and in most cases excellent results have been obtained.

But while there are currently quite a few effective algorithms for extracting rules directly from neural networks, none have ever been explicitly tested in deep neural networks. In addition, most authors focus on networks with very few hidden layers. In the past few years alone, several innovative analyses of specific methods for extracting rules from modified deep networks had emerged, and some approaches were presented that could accomplish that task.

2. Fuzzy models classification

Fuzzy models from the point of view of the degree of their specificity for performing various tasks of analyzing systems and processes can be conditionally divided into two large classes: universal and problem-oriented fuzzy models.

2.1 Universal fuzzy models

Universal fuzzy models are the main, basic class of fuzzy models. Their construction, use and analysis is based on the provisions of the theory of fuzzy sets, relations, statements, logic and calculations.

The universalism of these models is determined, first of all, by the fact that with their use it is possible to typify the setting and perform (with varying degrees of success) a wide range of individual (localized) analytical tasks, such as, for example, cluster analysis, classification, regression analysis, multivariate analysis, and predictive assessment, identifying trends and forecasting time series, identification, diagnostics, multi-criteria assessment and selection of alternatives, optimization, establishment and interpretation of dependencies, finding associations and other tasks.

The universal fuzzy models include fuzzy production, relational and functional models.

Fuzzy production models are a set of agreed fuzzy production rules of the form "If A, Then B", which defines fuzzy causal relationships between premises A and conclusions B, and is designed to determine the degree of truth of conclusions based on the degree of truth of premises. The functioning of fuzzy production models is based on the corresponding fuzzy inference algorithms.

In fuzzy relational models, input fuzzy variables are mapped to output fuzzy variables based on fuzzy relationships. These models allow pre-fixing the linguistic terms of input and output fuzzy variables and then adjusting fuzzy mappings by defining and changing elements of fuzzy relations. These models are in many ways similar to production models, differing, in fact, in the ways of forming fuzzy relations and setting up fuzzy mappings.

Fuzzy functional models are characterized by different ways of defining fuzzy functional dependencies. Historically, the following types of fuzzy functional models are distinguished: with fuzzy constraints, with the spread of fuzzy from independent to dependent variables, with clear variables, with fuzzy variables, combined types of functional models.

And a quite obvious feature of the considered universal fuzzy models is their lack of focus on the specifics of various aspects of the analyzed systems and processes, for example, on modeling on structures, states, system dynamics, interaction of elements, etc.
2.2. Problem-oriented fuzzy models

Another large class of fuzzy models includes the so-called problem-oriented fuzzy models designed to analyze individual properties and features of systems or processes. These models are usually more complex than the generic ones.

The problem-oriented fuzzy models include:
- fuzzy functional and relational scoring models;
- fuzzy event models;
- fuzzy state and control models.

Fuzzy functional and relational evaluation models are focused on solving problems of evaluating single objects, comparing and choosing alternatives. The basis of such models are relations of strict or non-strict fuzzy order on a set of alternatives or functional dependencies that compare the evaluated object with its value and generalize the value of the object by particular indicators.

Fuzzy event models are focused, first of all, at solving problems of setting connections (deterministic, random, non-stochastic) between events and determining their reachability.

The following main fuzzy event models can be distinguished:
- linguistic lotteries;
- fuzzy event trees;
- fuzzy fault trees;
- fuzzy Bayesian networks;
- fuzzy game models.

A linguistic lottery is defined as a linguistic random variable with a known linguistic probability distribution. In other words, such a lottery sets a set of outcome-probability pairs for a random variable whose values are described linguistically.

Fuzzy event trees and fuzzy fault trees are based on fuzzy graphs and are used to determine the reachability of some events when others occur. The vertices of these models correspond to events or fuzzy operations.

Fuzzy Bayesian networks make it possible to visualize the causal relationships between fuzzy events, expressed through fuzzy conditional probabilities, as well as to form direct and inverse fuzzy-probabilistic Bayesian inference in conditions of statistical, stochastic and non-stochastic uncertainty.

Fuzzy game models are aimed at determining optimal (rational) behavior strategies in confrontation problems with fuzzy described conditions. In this case, the initial data are the winnings / losses of the players when choosing one or another action.

Fuzzy state and control models are focused on solving problems of describing and assessing situations, as well as determining control methods to achieve target situations in dynamics.

This group of models includes:
- fuzzy “situation – action” models;
- fuzzy situational networks;
- fuzzy automata;
- fuzzy classification trees;
- fuzzy cognitive maps;
- fuzzy Markov and semi-Markov networks;
- fuzzy Petri nets.

Fuzzy “situation – action” models implement the idea of situational control - the correspondence of the control decision to a fuzzy current situation described in the form of premises of fuzzy rules or fuzzy sets of the 2nd level. In this case, it is not the control object that is modeled, but the way of controlling it to achieve a fuzzy target situation.

Fuzzy situational networks are represented in the form of graphs, the nodes of which correspond to fuzzy situations (described by fuzzy sets of the 1st or 2nd level), and the arcs correspond to possible transitions, which are "weighted" by fuzzy control decisions to transfer from one situation to another.
They are designed to determine the result of applying a sequence of control decisions (control trajectories) during the transition from the current to the target fuzzy situation.

Fuzzy automata are fuzzy directed graphs, the vertices of which are the states of the modeled system or process, and the arcs characterize possible transitions from state to state. Transitions from state to state are implemented as a result of control actions, and output signals are associated with each state.

Fuzzy cognitive maps are designed to formalize a problem in the form of a set of concepts that reflect systemic factors, and to identify fuzzy relationships of influence between them, taking into account the impact on these factors or changes in the nature of relationships.

Fuzzy classification trees are focused on recognition and sequential refinement of the state depending on the fuzzy values of the features.

Fuzzy Markov and semi-Markov networks are designed to model the transitions of a control object from state to state.

Fuzzy Petri nets are intended for adequate representation and analysis of the structure of dynamic discrete models of complex systems and fuzzy logical-temporal features of the processes of their functioning.

3. Methods for extracting rules from the neural network

In artificial intelligence problems, neural networks and machine learning methods based on knowledge representation represent two different approaches to solving classification problems. Both methods are the methods of designing models that create the classes for the experimental data. For most of the tasks, neural network training methods are very accurate.

However, neural networks have one major weakness: for a neural network, the ability to understand what event it models is weaker than for approaches based on knowledge representation models. The data used by neural networks for training is more difficult to understand, because it is represented in a neural network using a huge number of parameters [1].

Extracting knowledge representation models has two important features. First, it gives the user a clear understanding of how the neural network uses input data to make decisions. Second, it can reveal hidden functions in the neural network to explain the work of individual neurons. Identifying important attributes or identifying the causes of neural network errors is also part of the understanding process. To make black boxes of neural networks more understandable, methods for extracting knowledge representation models reduce the discrepancy between accuracy and comprehensibility [2].

More understandable presentation of the results of the solution is required if, for example, the neural network is to be used in security critical applications such as military operations or nuclear power plants. In such situations it is important for the system user to be able to implement the scenario of verification of the artificial neural network output with all possible input data [3].

To formalize the task of drawing rules from a neural network, a description of the network hypothesis is usually created, which is understandable, but its behavior approaches the network prediction [4].

To distinguish different approaches to rule extraction from neural networks a multidimensional taxonomy is used [3]. The first parameter that it describes is the expressive force of extracted rules (e.g., IF-THEN rules or fuzzy productive rules).

The second parameter is called transparency and describes the strategy that follows the results of the rule extraction algorithm. If the method uses a neural network only in black box quality, we call it a pedagogical approach. If the algorithm takes into account the neural network topology, we call this approach decompositional. If the algorithm uses elements of both pedagogical and decompositional methods, this approach is called the eclectic one. The third parameter is the quality of the extracted rules. When quality is a rather general term, it is divided into several criteria, namely: accuracy, fidelity, consistency and comprehensibility. While accuracy measures the ability to correctly classify previously unknown examples, fidelity measures the extent to which rules can mimic the narrative of a neural network [2].

Consistency can only be measured when an algorithm for attracting rules involves learning about the learning process of a neural network other than learning about an already learned neural network. The
resulting set of rules is considered consistent, when it correctly classifies the test data for different training samples. Clarity here is regarded as a measure of the rule size. Short and few rules are considered more comprehensible [3].

In this review we will focus only on three described parameters. We will focus on methods that do not impose special requirements on how the neural network was trained before the rules were extracted [5]. In addition, we will investigate only algorithms capable of extracting rules from forward propagation neural networks. In accordance with [6] we believe that the algorithm implies a high level of generalization.

Let us consider some methods of rule extraction, which correspond to the above description, starting with the decomposition approach. As mentioned above, decomposition approaches to rule extraction from neural networks work at the neuronal level. Usually decomposition method analyzes each neuron and forms rules imitating the behavior of this neuron. Among the possible decomposition approaches we consider the KT algorithm, the Tsukimoto polynomial algorithm and the rule extractor through the induction of the decision tree.

The KT algorithm was one of the first decomposition approaches for extracting rules from neural networks [7]. The KT algorithm describes each neuron with IF-THEN rules by heuristic search of combinations of input attributes exceeding the neuron threshold. To find suitable combinations the KT method applies a tree search, i.e. a rule (represented as a node in a tree) at this level generates its child nodes by adding an additional available attribute [8]. In addition, the algorithm uses several heuristics to stop the tree from growing in situations where further improvement is impossible.

Another method of rule extraction by induction of the solution tree was introduced in [9]. Their CRED algorithm converts each output vertex of a neural network into a solution where tree nodes are tested using hidden layer nodes and leaves are a class. The intermediate rules are then extracted. Then another solution tree is created for each branching point used in these rules, using branching points on the input layer of the neural network. Extracting rules from the second solution tree leads us to the description of the state of hidden neurons that depend on input variables. As a last step, intermediate rules describing the output layer through the hidden layer and rules describing the hidden layer based on neural network input data are replaced. Then they are combined into constructive rules describing the output of the neural network based on its input data.

The main class of pedagogical approaches of rule extraction from the neural network include validity interval analysis, approaches for rule using sampling and rule by reverse engineering.

Pedagogical approaches do not consider the internal structure of the neural network. The basis of pedagogical approaches is the attitude towards the trained neural network as a single object or a "black box" [10]. The main idea is to extract rules by directly displaying inputs to outputs [11].

Pedagogical approaches usually have access only to neural network function. This function makes the output-exit of the neural network dependent on the input but does not give an understanding of the inner structure of the neural network or any weights. This class of algorithms tries to find a relationship between possible input and output variations created by the neural network, some of them using given learning data, and some do not.

Rule extraction based on interval analysis uses validity interval analysis to extract rules that simulate neural network behavior [12]. The main idea of this method is to find input intervals at which the neural network output signal is stable, i.e., the predicted class matches for small changes in inputs. Thus, interval analysis provides the basis for precise, reliable rules.

Obtaining rules by sampling is a series of methods that follow the same strategy for extracting rules from a neural network with the help of sampling, i.e., they create an extensive set of data as a basis for extracting rules. After that, the selected set of data is passed to the standard learning algorithm for generating rules that simulate the behavior of the neural network. In [2] it is proved that the use of sample data exceeds the use of conventional tutorial data in rules extraction tasks.

One of the first methods, which followed this strategy, was Trepan's algorithm [13]. It is very similar to the "divide and conquer" algorithm of C4.5 by searching for points of division into teaching data for separate instances of different classes. The main differences from the "divide and conquer" method are
the best expansion strategy of the tree structure, additional branching points and the possibility to select teaching examples in deeper tree nodes. As a result, the algorithm also creates a decision tree, which can be transformed into a set of rules.

A binarized input-output rule extraction algorithm is presented in [4]. The algorithm generates all possible binarized input combinations and collects the results from the network. Using the neural network output, a truth table is created for each example. From the truth table, if necessary, it is just as easy to go to the rules.

The ANN-DT method is another decision-based sampling method to describe neural network behavior [13]. The general algorithm is based on CART method with some changes in the initial implementation. ANN-DT uses a sampling method to extend the learning set so that most of the learning set remains representative. This is achieved by using the nearest-neighbor method, which calculates the distance from the sampling point to the nearest point of the learning set [13] and compares it with the original value. The STARE algorithm [14] implements the principle of creating a large set of examples at the first stage. By analogy with BIO-RE, the STARE method also builds large truth tables but it works with continuous input data also. An example of pedagogical approach using educational data sampling is KDRuleEx [15]. Similar to Trepan, the KDRuleEx algorithm generates an additional teaching sample when the bases for the next branching points are insufficient. KDRuleEx uses evolutionary methods to create new learning examples. The technology leads to a solution table, which can be easily converted into IF-THEN rules.

The eclectic approach to rule extraction includes elements of both pedagogical and decomposition approaches. In particular, the eclectic approach uses information about the internal architecture and neural network weight vectors to improve the symbolic learning algorithm [3]. The problems of rule extraction from artificial neural networks are only a small part of the problem of explainability based on subversive models (e.g. deep neural networks), which was not present in classic AI (namely, rule-based expert systems and models). These problems are included in the field of explainable AI (XAI), which is admittedly a crucial part of practical deployment of AI models.

4. Extracting Rules from Deep Neural Networks and Neuro-Fuzzy Networks
At present, the direction of rule extraction using neural fuzzy models is actively developing. Systems based on fuzzy rules (FRBS), developed using fuzzy logic, have become a rapidly growing field over the past few years. These algorithms have proven their strengths in such tasks as controlling complex systems, creating fuzzy controls. The relationship between the two approaches (ANN and FRBS) has been carefully studied and results have been obtained on their mutual correspondence 16. This leads to two extremely important conclusions. First, we can apply the methods used in one model, to the other model. Secondly, we can present the knowledge embedded in the neural network in a more understandable algebraic language of fuzzy production rules. In other words, we can get algebraic interpretation of neural networks [17].

Since 2012, we have started a stormy neural network of deep learning. One of the first deep revolutionary neural networks is Alexnet, which won the annual Imagenet competition and was trained on the Imagenet data set containing 15 million images. One of the last winners in 2016 is the Chinese University of Hong Kong neural network containing 269 layers.

In order to obtain a clear semantic interpretation for in-depth networks, it is possible to use fuzzy neural networks instead of the last full-connected neural network on the last layer.

For example, ANFIS (Adaptive Neural Fuzzy Inference System) is a multilayer direct distribution network. This architecture has five layers, such as a fuzzy layer, a product layer, a normalized layer, a defuzzification layer, and a common output. ANFIS has the property of a neural network and a fuzzy logic system. The goal of fusion of fuzzy logic architectures and neural networks is to design an architecture that uses fuzzy logic to demonstrate knowledge in a clear way while the neural network maximizes its parameters in training. ANFIS is used in many applications such as function approximation, intelligent management and time series forecasting. Deep neural networks and fuzzy neural networks can be combined in different ways. A hypothetical system can be created using two
components. The first is the generation of a deep learning function, which can be used to create representative features directly from text. Initially, the deep learning system will be trained to work with undetected data. Once these elements are extracted from the in depth-learning system, they will be integrated into systems with fuzzy findings. These systems may include both elements found in the in depth-learning process and subjective information from the analyst. These two parts together can be used for classification purposes. In this way, the final system will be able to report both on the classification results and on the specific features and rules that have been activated for the system to be completed. In addition, the resulting system can be further used by the analyst as a feedback form.

A very interesting approach is proposed in [18], where the author established an important link between two big parts of artificial intelligence, i.e., deep learning and fuzzy logic. He shows the benefit that deep learning can bring to comparative research by rethinking many of the heuristics of traces and errors in the field of fuzzy logic, and discover parts with a strict foundation. The author proposed deep generalized hamming network (GHN), which not only can be thoroughly analysed and interpreted within the framework of fuzzy logic theory, but also demonstrates fast learning speed, well-controlled behaviour, and state-of-the-art understanding of various learning objectives.

Thanks to the theory of fuzzy sets, using fuzzy relationships and rules, you can create an effective model for predicting time series with many inputs and one output (forecast). Such an approach allows us to make a kind of justification for the operation of an artificial neural network using neural-fuzzy models on the one hand and fuzzy cognitive maps on the other. We have developed a hybrid modular forecasting model that combines the theory of fuzzy logic, cognitive maps, and artificial neural networks. The modular system consists of several specialized modules. In general, these modules have the following characteristics: 1. System modules are specific and have specialized computing architectures to recognize and respond to specific subtasks of a large common task. 2. Each module, as a rule, is independent of other modules in its functioning and does not affect other operation of other modules. 3. Modules have a simpler architecture compared to the system. Thus, the module is faster than a complex monolithic system. 4. The results of each module individually are combined using a special integration module (in our case, the forecast consensus module), due to which the highest forecast accuracy of the entire system is achieved. The forecasting task is made by the three main modules of the system. The adaptive neuro-fuzzy network inference system (ANFIS) gives us the quantitative forecast, the results of which are send to a verification system. If the result gets in the given interval, then it is send to the next module. Simultaneously a fuzzy cognitive map (FCM) module is working, which receives the time series as an input. At the output, FCM gets a forecast with the fuzzy membership of its fulfilment. All the data from these modules is sent to the third module, which operates on the basis of the ANFIS network, which combines the information received from the previous modules and gives the final result. Figure 1 presents a model of a forecasting system.

![Figure 1. Developed forecasting system.](image-url)
5. Acknowledgments
This research was performed in the framework of the state task in the field of scientific activity of the Ministry of Science and Higher Education of the Russian Federation, project "Development of the methodology and a software platform for the construction of digital twins, intellectual analysis and forecast of complex economic systems", grant no. FSSW-2020-0008.

References
[1] Craven M 1994 Using sampling and queries to extract rules from trained neural networks (ICML) 37-45
[2] Johansson U 2006 Rule ex-traction from opaque models—a slightly different perspective (Machine Learning and Applications ICMLA’06 5th International Conference) 11(2) 22-27
[3] Andrews R 1995 Survey and critique of techniques for extracting rules from trained artificial neural networks (Knowledge-based systems) 8(6) 373-389
[4] Craven M 1996 Extracting comprehensible models from trained neural networks (University of Wisconsin-Madison)
[5] Thrun S 1993 Extracting provably correct rules from artificial neural networks (Technical report, University of Bonn, Institut für Informatik)
[6] Comer D 1999 Rule extraction: Where do we go from here (University of Wisconsin Machine Learning Research Group Working Paper) 99-108
[7] Fu L 1994 The ubiquitous b-tree (Systems, Man and Cybernetics IEEE Transactions) 24(8) 1114-1124
[8] Tsukimoto H 2000 Extracting rules from trained neural networks (Neural Networks, IEEE Transactions) 11(2) 377-389
[9] Tsukimoto H 2001 Rule extraction from neural networks via decision tree induction (Neural Networks. International Joint Conference) 3 1870-1875
[10] Tickle A B 1998 The truth will come to light: directions and challenges in extracting the knowledge embed- ded within trained artificial neural networks (IEEE Transactions on Neural Networks) 9(6) 1057-1068
[11] Thrun S 1995 Extracting rules from artificial neural networks with distributed representations (Advances in neural information processing systems) 505-512
[12] Craven M W 1996 Extracting tree-structured representations of trained networks (Advances in neural information processing systems) 24-30
[13] Taha I A 1999 Symbolic interpretation of artificial neural networks (Knowledge and Data Engineering, IEEE Transactions) 11(3) 448-463
[14] Towell G G 1993 Extracting refined rules from knowledge-based neural networks (Machine learning) 13(1) 71-101
[15] Sethi K 2012 KDRuleEx: A novel approach for enhancing user comprehensibility using rule extraction (Intelligent Systems Modelling and Simulation) (ISMS) 55-60
[16] Averkin A N 2018 Hybrid Neural Networks and Time Series Forecasting (Springer) 934(1) 230-239 [Pilato18]
[17] G. Pilato G 2018 Prediction and Detection of User Emotions Based on Neuro-Fuzzy Neural Networks in Social Networks (Springer) 875(2) pp 118-126
[18] Setiono D 2000 FERNN: An algorithm for fast extraction of rules from neural networks (Applied Intelligence) 12(1-2) 15-25