Rules Acquisition from Classic, Deep and Neuro-Fuzzy Systems

Alexey Averkin[a,b], Sergey Yaryshev[c]

[a]*Federal Research Centre of Informatics and Computer Science of RAS, Moscow, Russia*
[b]*Educational and Scientific Laboratory of Artificial Intelligence, Neuro-technologies and Business Analytics, Plekhanov Russian University of Economics, Moscow, Russia*
[c]*Plekhanov Russian University of Economics, Moscow, Russia*

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 three 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 explanational artificial intelligence and machine learning.

1. Introduction

This article presents the basic methods of machine learning and explanational 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. This step also proposes algorithms that can efficiently extract rules from these more complex and neuromorphic neural networks.

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

Russian Advances in Artificial Intelligence: selected contributions to the Russian Conference on Artificial intelligence (RCAI 2020), October 10-16, 2020, Moscow, Russia
averkin2003@inbox.ru (A. Averkin); yarushev.sa@rea.ru (S. Yaryshev)

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 explanational 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 overcome these shortcomings of neural networks, data scientists have come up with 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. 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[Crav94].

Increasing the intelligibility of neural networks by extracting knowledge representation models has two important advantages. 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. In an attempt to make black boxes of neural networks more understandable, methods for extracting knowledge representation models reduce the discrepancy between accuracy and comprehensibility[Joh06].

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 [Andr95].

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[Crav96].

To distinguish different approaches to rule extraction from neural networks a multidimensional taxonomy is used [Andr95]. 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[Joh06]. 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[Andr95].

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[Thrun93]. In addition, we will investigate only algorithms capable of extracting rules from forward propagation neural networks. In accordance with[Crav99] 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[Fu94]. 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 [Tsuki00]. In addition, the algorithm uses several heuristics to stop the tree from growing in situations where further improvement is impossible.

The polynomial algorithm to extract rules from the neural network is very similar to the KT method. It also uses a multilevel decomposition algorithm to extract IF-THEN rules for each neuron and monitors the strategy for finding inputs that exceed the neuron's threshold. The main advantage of the Tsukimoto method is its computational complexity, which is polynomial, while the KT method is exponential[Fu94]. The algorithm achieves polynomial complexity by searching for corresponding terms using the space of multiline functions. In the second stage, these terms are used to create IF-THEN rules. At the last stage, the Tsukimoto algorithm tries to optimize comprehensibility by removing insignificant attributes from the rules.

Another method of rule extraction by induction of the solution tree was introduced in[Tsuki01]. 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"[Tick98]. The main idea is to extract rules by directly displaying inputs to outputs[Thrun95].

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[Crav96]. 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[Joh06] 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 [Taha99].It is very similar to the "divide and conquer" algorithm of C4.5[Quin93] 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.

Another of these very common pedagogical approaches that use sampling to extract rules from a neural network is presented in[Crav96]. An algorithm called binarized input-output rule extraction can only handle neural network with binary or binarized input attributes. Binarized input-output rule extraction creates all possible input combinations and requests 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[Taha99]. 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 [Taha99] and compares it with the original value. The STARE algorithm[Towell93]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 for learning. The advantage of STARE is its ability to work with continuous input data. To generate truth tables, the algorithm rearranges the input data, and for each continuous attribute, it is required to sample it at high frequency across all values. An example of pedagogical approach using educational data sampling is KDRuleEx [Sethi12]. Similar to Trepan, the KDRuleEx algorithm generates an additional teaching sample when the bases for the next branching points are insufficient. KDRuleEx uses evolutional 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 [Crav99]. In particular, the eclectic approach uses knowledge about the internal architecture and neural network weight vectors to complement the symbolic learning algorithm [Andr95]. It tries to identify corresponding latent neurons as well as corresponding inputs to the network. For this purpose, the solution tree is built using the well-known C4.5 algorithm. The rule extraction process leads to the generation of M-of-N and IF-THEN rules. With a set of correctly classified learning examples, FERNN analyses the activation values of each hidden unit. For each hidden vertex, the activation values are sorted in ascending order. The C4.5 algorithm is then used to find the best branching point to form a decision tree. 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 [Arr19] and [Arya19].

3. 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 [Aver18]. 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[Pilato18],[Aver18].

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)[Jang]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[Bon17]. 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 [Fan17], where the author established a fundamental connection between two important areas of artificial intelligence, i.e., deep study and fuzzy logic. He shows the benefit that deep study can bring to comparative research by rethinking many of the heuristics of traces and errors in the lens of fuzzy logic, and thus distilling essential ingredients with a strict foundation. The author proposed deep generalized hamming network (GHN), which not only can be thoroughly analyzed and interpreted within the framework of fuzzy logic theory, but also demonstrates fast learning speed, well-controlled behavior, and state-of-the-art understanding of various learning objectives. The [Zilke16] presents another approach for including such rule-based methodology in neural networks by embedding systems of fuzzy conclusions in networks of deep learning.

Thanks to the theory of fuzzy sets, using fuzzy relationships and rules, you can create an effective model for predicting time series with a large number of 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 as a whole 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 as a whole. 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 fore-cast accuracy of the entire system is achieved. The system has three main modules responsible for the forecasting task. The ANFIS neuro-fuzzy network performs a time series forecast based on numerical indicators and gives us the so-called quantitative fore-cast, the results of which pass through a verification system (assessment of the adequacy of the forecast), if the fore-cast corresponds to the necessary accuracy, then it is transmitted to the next module. In parallel with the neuro-fuzzy network, a module with a fuzzy cognitive map is working, which receives data on the event effect on the time series as an input, a cognitive map is constructed in which all factors of influence on a specific predicted indicator are taken into account. At the output, the cognitive map gives us a forecast with the probability of its fulfillment, that is, with the consonance of a factor that tells us whether the forecast will be fulfilled or not. Further, all the data received from these modules is sent to the third module, which operates on the basis of the ANFIS network, which aggregates the information received from the previous modules and gives the final consensus forecast. Figure 1 presents a model of a forecasting system.

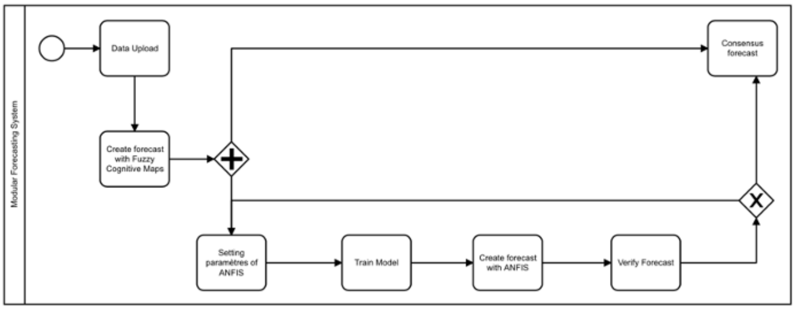

Figure 1: Modular forecasting system

4. Conclusion

This paper attempts to provide a review of several rule extraction algorithms from an artificial neural network. Some of the state-of-the-art algorithms are discussed from each category named as decompositional, pedagogical, and eclectics. Currently, deep learning provides an acceptable solution for lots of problems. It is a new machine learning area which is believed to move machine learning a step ahead. But it is still a black box system.

Quite a number of authors are trying to establish a link between two important areas of artificial intelligence - deep learning and fuzzy logic. Until recently, fuzzy logic has been poorly used in machine learning. Extracting fuzzy rules is one way to help semantically interpret neural networks. This research will allow researchers of fuzzy logic to develop artificial intelligence applications and solve problems that are of interest to machine learning.

4.0.1. Acknowledgements

The paper is partially support by grants of RFBR 20-07-00770 A and 20-010-00828 A.

References

[Crav94] M. Craven. Using sampling and queries to extract rules from trained neural networks. *ICML*, 37–45, 1994.

[Joh06] U. Johansson. Rule ex-traction from opaque models–a slightly different perspective. *Machine Learning and Applications. ICMLA'06. 5th International Conference*, 11(2):22–27, 2006.

[Crav99] D. Comer. Rule extraction: Where do we go from here. *University of Wisconsin Machine Learning Research Group Working Paper*, 99–108, June 1999.

[Sethi12] K. Sethi. KDRuleEx: A novel approach for enhancing user comprehensibility using rule extraction. *Intelligent Systems, Modelling and Simulation (ISMS)*, 55–60, 2012.

[Andr95] R. Andrews. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-based systems*, 8(6):373—389, 1995.

[Crav96] M.W. Craven. Extracting comprehensible models from trained neural networks. *PhD thesis, University of Wisconsin-Madison*, June 1996.

[Thrun93] S. Thrun. Extracting provably correct rules from artificial neural networks. *Technical report, University of Bonn, Institut für Informatik III*, 1993.

[Fu94] L. Fu. The ubiquitous b-tree. *Systems, Man and Cybernetics, IEEE Transactions*, 24(8):1114—1124, 1994.

[Tsuki00] H. Tsukimoto. Extracting rules from trained neural networks. *Neural Networks, IEEE Transactions*, 11(2):377—389, 2000.

[Tsuki01] H. Tsukimoto. *Rule extraction from neural networks via decision tree induction – Volume 3, pages 1870–1875 / Neural Networks.* Iternational Joint Conference, 2001.

[Tick98] A.B. Tickle. The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural networks. *IEEE Transactions on Neural Networks*, 9(6):1057—1068, June 1998.

[Thrun95] S. Thrun. Extracting rules from artificial neural networks with distributed representations. *Advances in neural information processing systems*, 505–512, 1995.

[Crav96] M.W. Craven. Extracting tree-structured representations of trained networks. *Advances in neural information processing systems*, 24–30, 1996.

[Taha99] I.A. Taha. Symbolic interpretation of artificial neural networks. *Knowledge and Data Engineering, IEEE Transactions*, 11(3):448—463, June 1999.

[Towell93] G.G. Towell. Extracting refined rules from knowledge-based neural networks. *Machine learning*, 13(1):71—101, 1993.

[Seti00] D. Setiono. FERNN: An algorithm for fast extraction of rules from neural networks. *Applied Intelligence*, 12(1-2):15–25, 2000.

[Aver18] A.N. Averkin. Hybrid Neural Networks and Time Series Forecasting. *Springer*, 934(1):230–239, 2018.

[Pilato18] G. Pilato. Prediction and Detection of User Emotions Based on Neuro-Fuzzy Neural Networks in Social Networks. *Springer*, 875(2):118–126, June 1979.

[Aver18] A.N. Averkin. The ubiquitous b-tree. *Computing Surveys*, 11(2):121–137, June 1979.

[Zilke16] J.R. Zilke. DeepRED - Rule Extraction from Deep Neural Networks. *Springer*, 457–473, 2016.

[Fan17] L. Fan. Revisit Fuzzy Neural Network: Demystifying Batch Normalization and ReLU with Generalized Hamming Network. *NIPS*, 1923–1932, 2017.

[Quin93] J. R. Quinlan. The ubiquitous b-tree. *Morgan Kaufmann*, volume 1:4–5, 1993.

[Bon17] D. Bonanno. An approach to explainable deep learning using fuzzy inference. *Next-Generation Analyst V*, 102070D, 2017.

[Jang] S. R. Jang. ANFIS: Adaptive-network-based fuzzy inference systems. *IEEE Trans. On Systems, Man, and Cybernetics*, 23:665-685, 1992.

[Arr19] A.B. Arrieta. Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. *Information Fusion*, arXiv–1910, 2019.

[Arya19] V. Arya. One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques. *arXiv preprint*, arXiv:1909.03012, 2019.