Taylor & Francis
Taylor & Francis Group

Applied Artificial Intelligence
An International Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

Uncertainty of Rules Extracted from Artificial Neural Networks

Hurnjoo Lee & Hyeoncheol Kim

To cite this article: Hurnjoo Lee & Hyeoncheol Kim (2021) Uncertainty of Rules Extracted from Artificial Neural Networks, Applied Artificial Intelligence, 35:8, 589-604, DOI: 10.1080/08839514.2021.1922845

To link to this article: https://doi.org/10.1080/08839514.2021.1922845

Published online: 12 May 2021.

Submit your article to this journal

Article views: 480

View related articles

View Crossmark data

Citing articles: 1 View citing articles

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=uaai20

APPLIED ARTIFICIAL INTELLIGENCE
2021, VOL. 35, NO. 8, 589–604
https://doi.org/10.1080/08839514.2021.1922845

Taylor & Francis
Taylor & Francis Group

Check for updates

Uncertainty of Rules Extracted from Artificial Neural Networks

Hurnjoo Lee and Hyeoncheol Kim

Department of Computer Science and Engineering, Korea University, Seoul, Korea

ABSTRACT

Artificial neural networks evolve into deep learning recently and perform well in various fields, such as image and speech recognition and translation. However, there is a problem that it is difficult for a person to understand what exactly the trained knowledge of an artificial neural network. As one of the methods for solving the problem of the artificial neural network, rule extraction methods had been devised. In this study, rules are extracted from artificial neural networks using ordered-attribute search (OAS) algorithm, which is one of the methods of extracting rules from trained neural networks, and the rules are analyzed to improve the extracted rules. As a result, we found that when the output value of the hidden layer has an intermediate value that is not close to 0 or 1 after passing through the sigmoid function, the problem of rule uncertainty occurs and affects the accuracy of the rules. In order to solve the uncertainty problem of the rules, we applied the hidden unit clarification method and suggested that it is possible to extract the efficient rule by binarizing the hidden layer output value. In addition, we extracted CDRPs (critical data routing paths) from the trained neural networks and used CDRPs to prune the extracted rules, which showed that the uncertainty problem of rules can be improved.

Introduction

The artificial neural network evolves into deep learning recently and shows excellent performance in the areas of image and speech recognition and translation. However, in spite of these excellent performances, there is a problem that it is difficult for a person to understand what exactly the trained knowledge of artificial neural network and there is still a risk that the decision error may be fatal for use in medical fields or military fields requiring high-reliability verification (Hailesilassie 2016). Various methods for solving the problems of the artificial neural network have been studied. One of them is a method of extracting rules that can be understood by human from the artificial neural network. This extracted rules make it possible to provide

CONTACT Hyeoncheol Kim harrykim@korea.ac.kr Korea University, Lyceum Bldg, 145 Anam-ro, Seongbuk-gu, Seoul 02841, Korea

This article has been republished with minor changes. These changes do not impact the academic content of the article.
© 2021 Taylor & Francis

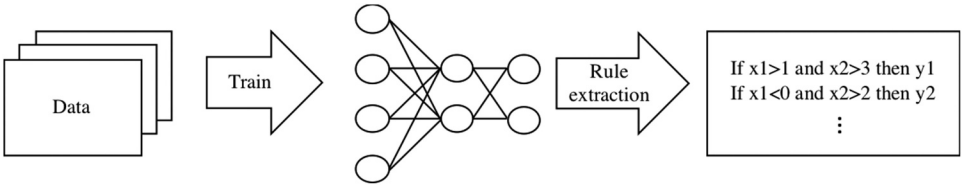

Figure 1. Rule extraction from trained artificial neural network.

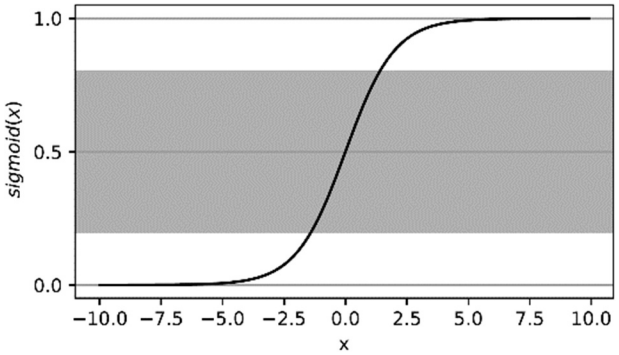

Figure 2. Intermediate value of sigmoid function.

information in a form that can be understood by human beings about decisions of artificial neural networks.

Rule extraction has three approaches: decompositional/pedagogical/eclectic (Andrews, Diederich, and Tickle 1995; Hailesilassie 2016). In this study, we investigated what rules are extracted when we use the ordered-attribute search (OAS) algorithm (Kim 2000), which is one of the decompositional approaches. The decompositional approach is an approach to extract the rules by viewing the artificial neural network as a white box. One of the disadvantages of the decompositional approach, such as the OAS algorithm, is that uncertainty problems can occur in the process of binarizing the output from sigmoid, which is an activation function of the hidden unit.

In order to solve this problem, we analyze the output value of the hidden unit and analyze the uncertainty problem of the rule extracted from the artificial neural network in detail. And we applied hidden unit clarification (Ishikawa 1996) to the process of training the artificial neural network and confirmed how much improvement was made in the extracted rule by using the OAS algorithm through experiments. In addition, we have confirmed through experiments that it is helpful to improve the quality of rules when pruning extracted rules using CDRPs (Critical Data Routing Paths).

The main contribution of this paper is to analyze the uncertainty problem of rules, one of the problems of decompositional approach rule extraction, and to

show that we can improve this problem to some extent by using hidden unit clarification and CDRPs.

The composition of this paper is as follows. In Section 2, we introduce each rule extraction algorithm, hidden unit clarification algorithm, and CDRPs. Section 3 describes the data and data preprocessing used in the study, the OAS algorithm and the hidden unit clarification algorithm, and the CDRPs. In In Section 4, we describe the performance indicator, the analysis of experimental results, and the effect of hidden unit clarification and CDRPs on rule extraction. Section 5 summarizes the results of the study and describes the direction of future work.

Related Work

Rule Extraction Algorithm

As shown in Table 1, there are three approaches to extract rules from trained neural networks: decompositional approach, pedagogical approach, and eclectic approach.

Decompositional Approach

The decompositional approach is an approach to extract the rules by viewing the artificial neural network as a white box. The decompositional approach is computationally expensive and uses a lot of search space but better than other approaches in terms of transparency because it looks at the all units in each layer and extracts the rules. The decompositional approach involves the following steps (Kim 2000).

(1) Extract intermediate rules at the level of individual units of an artificial neural network.
(2) In the rewrite step, the symbols of the intermediate rule extracted from each unit are removed. During the rewriting process, rules that overlap with other rules, are included in other rules, or are inconsistent with each other are removed.

Table 1. Rule extraction algorithms.

Approach	Rule extraction algorithm
Decompositional	KT(knowledgetron) (Fu 1994, 1991, 1993)
	OAS(ordered-attribute search) (Kim 2000)
Pedagogical	VIA(validity interval analysis(Thrun 1995)
	BIO-RE(binarized input-output rule extraction) (Taha and Ghosh 1999)
	ANN-DT(artificial neural-network decision tree algorithm) (Schmitz, Aldrich, and Gouws 1999)
Ecletic	MofN(Towell and Shavlik 1993)
	FERNN(fast extraction of rules from neural networks) (Setiono and Leow 2000)

The knowledgetron(KT) algorithm proposed by Fu (1993) is one of the initially proposed decompositional approaches (Fu 1994, 1991, 1993). KT algorithm is a rule extraciton method by using the structure of artificial neural network as the most intuitive method. The KT algorithm represents all neurons as If-Then rules, and finds a combination of input values that exceeds the threshold of a neuron. There are two kinds of rules generated by the KT algorithm: confirm rule/disconfirm rule. In the case of confirm rule, output is activated and disconfirm rule is opposite.

The OAS algorithm used in this study is also based on a decompositional approach. One of the problems of existing decompositional approaches is that it is costly to calculate and the rule search space exponentially increases according to the number of input attributes. The OAS algorithm reduces the search space of rules and proposes a computationally efficient method.

When the rule is extracted through the OAS algorithm, the output value of each node passes through the sigmoid. At this time, there is no problem when the output value is close to 0 or 1, which is clearly activated or deactivated. However, due to the nature of the sigmoid function, it also has an intermediate value between them as shown in Figure 3, which causes problems such as uncertainty of the rule.

Pedagogical Approach

The pedagogical approach is an approach to obtain the rules through the relationship between input and output only by considering the artificial neural network as a black box without considering the internal structure of the artificial neural network, unlike the decompositional approach described above. The pedagogical approach is more efficient in terms of search space and computation time than the decompositional approach in which algorithms are individually applied to all layers and then rewritten. However,

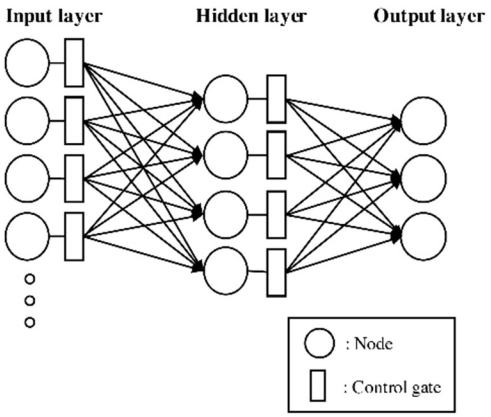

Figure 3. Artificial neural network with control gates.

since the artificial neural network is regarded as a black box and the relation between input and output is only used, the transparency is relatively lower than the decompositional approach (Hailesilassie 2016).

There are mmethods, suchas VIA (validity interval analysis) (Thrun 1995), binarized input-output rule extraction (BIO-RE) (Taha and Ghosh 1999), and ANN-DT (Schmitz, Aldrich, and Gouws 1999). In the case of BIO-RE, only binary values or binary attribute values can be used as inputs, and all possible combinations of attributes are input as artificial neural network input values, and create a truth table.

Eclectic Approach

The eclectic approach is an approach that uses both the decompositional and pedagogical approaches described above. For example, MofN (Towell and Shavlik 1993) and FERNN (Setiono and Leow 2000) exist.

Hidden Unit Clarification

As mentioned above, there is a problem that the output value of each node of the trained neural network cannot be binarized clearly, and the uncertainty problem of rules may occur when extracting rules by applying OAS algorithm. In order to solve this problem, in this study, we tried to improve this problem by applying the hidden unit clarification algorithm (Ishikawa 1996) as follows.

$$J_h = J + c\sum_i \min\{1 - h_i, h_i\} \tag{1}$$

The hidden unit clarification algorithm is one of the three algorithms mentioned in structural learning with forgetting (Ishikawa 1996), which trains the artificial neural network in such a way that the output value of the hidden unit is maximally activated or deactivated as much as possible. The training using the hidden unit clarification can be expressed as follows. In the above equation, J is a cost function, a cross entropy function, and h_i is an output value of an i-th hidden node having a value between 0 and 1. The value added to the cost function is the penalty term and c is the weight of this penalty term. The penalty term has a small value when the value of all h_i approaches zero or one. The artificial neural network is trained so that the output value of each hidden unit is close to full activation or deactivation due to the new cost function J_h added with the penalty term.

Critical Data Routing Paths

Critical Data Routing Paths (CDRPs) are a way to find critical nodes on the data routing path for each input sample (Wang et al. 2018). This method uses

the distillation guided routing method that puts a control gate on the output channel of each layer in the trained neural network model, and trains the control gate so that the output value becomes similar to output values of the original model. The extracted CDRPs can tell which nodes are important in artificial neural networks for each input data.

The optimization of the group of control gates in a network with K layers is done using the cost function of the following equation (Wang et al. 2018).

$$\min_{\Lambda} L(f_\theta(x, f_\theta(x;\Lambda)) + \gamma \sum_k |\lambda_k|_1 \tag{2}$$

$$s.t. \lambda_k \geq 0, k = 1, 2, \ldots, K \tag{3}$$

$$\Lambda = \lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_K \tag{4}$$

In the above equations, L is the cross-entropy function between the original model and the control-gate combined model and γ is the balance parameter and we used L1 normalization to make λ sparse. If the model with the control gate is trained by the above cost function, the unnecessary node becomes closer to 0 and the critical node becomes closer to 1. The trained control gate is binarized with a threshold value to finally obtain CDRPs information.

Model

In this study, artificial neural network with one input layer and one hidden layer is used. The number of nodes in each layer is shown in Table 2, and the activation function of the hidden layer and the output layer is sigmoid, and the cross entropy is used as the cost function.

In order to apply the hidden unit clarification algorithm to the trained artificial neural network, the trained artificial neural network was further trained by adding a penalty term to the cost function as shown in Equation (1).

And performed additional training of artificial neural network that combined the control gate in order to extract CDRPs. The structure of the artificial neural network with the control gate is shown in Figure 3 and the application of CDRPs proceed in the following order.

(1) Training artificial neural network using datasets
(2) Copying the model and then combining the control gates to each node
(3) Training model combining control gates

Table 2. Structure of artificial neural network.

	Input layer	Hidden layer	Output layer
The number of nodes	12	4	3

(4) Obtain CDRPs information by binarizing the value of the control gates
(5) Extract rules from artificial neural networks
(6) Using CDRPs information to prune rules that include unnecessary nodes

Experiments

In this chapter, we describe how the uncertainty problem of rules is improved by experimenting with applying hidden unit clarification and applying of CDRPs.

Section 4.1 explains what data was used and what preprocessing was done. Section 4.2 describes the process of model training. Finally, Section 4.3 describes the results of applying hidden unit clarification and CDRPs.

Data Preprocessing

In this study, IRIS domain, which is relatively simple public data, is applied for the experiment. The IRIS domain is data from a statistician, Fisher, which shows the width and length of sepals and petals for three species of iris. The reason for using a simple data set is that,

(1) The computational complexity of the OAS is high and it takes a long time until the result is obtained when using a complicated data set.
(2) The main purpose of this study is that check how the rule extraction is affected when applying hidden unit clarification and CDRPs.

In order to apply the OAS algorithm to IRIS data, input attribute with continuous value is transformed into data with 12 attributes by discretizing input attribute with 3 intervals as in Table 3 (Kim 2000).

Model Training

To make sure that the rules made by applying the OAS algorithm to artificial neural networks are improved through the application of the hidden unit clarification and CDRPs, made and trained the following model.

First, an artificial neural network composed of an input layer, a hidden layer, and an output layer was constructed to study IRIS data. As shown in Table 1, the artificial neural network consists of 12 nodes to have 12 discretized attributes as input values, and the hidden layer has 4 nodes and the output layer has 3 nodes. The activation function of the hidden layer and the output layer is sigmoid, and the cross entropy is used as the cost function.

Second, we applied the hidden unit clarification method to insert penalty term into the cost function as shown in Equation (1). Then, the artificial neural

Table 3. Discretized IRIS data.

Attribute	Discretized Attribute
sepal-length	sepal-length ≤ 5.4
	5.4 < sepal-length ≤ 6.3
	6.3 < sepal-legnth
sepal-width	sepal-width ≤ 2.8
	2.8 < sepal-width ≤ 3.1
	3.1 < sepal-width
petal-length	petal-length ≤ 2.7
	2.7 < petal-length ≤ 5
	5 < petal-length
petal-width	petal-width ≤ 0.7
	0.7 < petal-width ≤ 1.6
	1.6 < petal-length

network was further trained to see how the hidden layer output value and rule results improved.

Third, we trained the model using the distillation guided routing method, and extracted CDRPs. The original method for extracting CDRPs extracts CDRPs for each input sample, but in this study, we extracted the CDRPs for each category by dividing the data into categories without obtaining the CDRPs for each input sample. We also experimented how changes in the CDRPs threshold which determine a critical or non-critical node affect the rules.

Finally, using the binarized CDRPs information based on the CDRPs threshold, rules containing negligible nodes were regarded as unnecessary rules and were removed from the list of rules.

Evaluation and Analysis

Performance Indicators

Each rule extracted by applying the OAS algorithm evaluates the performance through accuracy and coverage (Kim 2000).

$$Coverage\ of\ each\ rule = (pos + neg)/total \tag{5}$$

$$The\ accuracy\ of\ each\ rule = pos/(pos + neg) \tag{6}$$

Where pos is a positive sample number when applying the experimental data set to any individual rule, neg is the negative sample number, and (pos + neg) is the number of samples covered by the rule. total is the number of all samples in the data set. The calculated coverage indicates how many samples are in each rule, and the accuracy indicates how much of the sample is positive. The above performance indicator is an performance indicator for the individual rules. The accuracy and coverage of the entire ruleset are calculated as follows.

$$Coverage\ of\ the\ entire\ rule\ set = (pos + neg)/total \tag{7}$$

$$Classification\ accuracy\ of\ entire\ rule\ set = pos/total \tag{8}$$

Here, pos and neg calculate the number of samples for all rulesets.

Analysis of Trained Artificial Neural Network

Figure 4 shows the distribution of output values of the hidden layer of the artificial neural network when train 2000 epoch with IRIS data. Since the activation function is sigmoid, the output value of the hidden layer has a value between 0 and 1. It can be confirmed that most of the values are between 0 and 0.15 and between 0.7 and 1, but there are also values between them.

Analysis of Result of OAS Algorithm

We applied the OAS algorithm to the trained artificial neural network to extract the rules. A total of 100 experiments were repeated and the results of repeated experiments were averaged. As a result, an average of 82.91 rules were extracted and the extracted rule showed an accuracy of 0.967.

The rules shown in Table 3 are some of the results obtained by applying the OAS algorithm to the IRIS domain and extracting them using the If-Then rule. In order to find ways to improve the above results, we classified rules as high and low accuracy. And we analyzed the rule which less accurate rule. The rule shown below is a rule whose accuracy is only 20%.

If (2.7 < petal-length < = 5) and (1.6 < petal-length) then versicolor 0.2

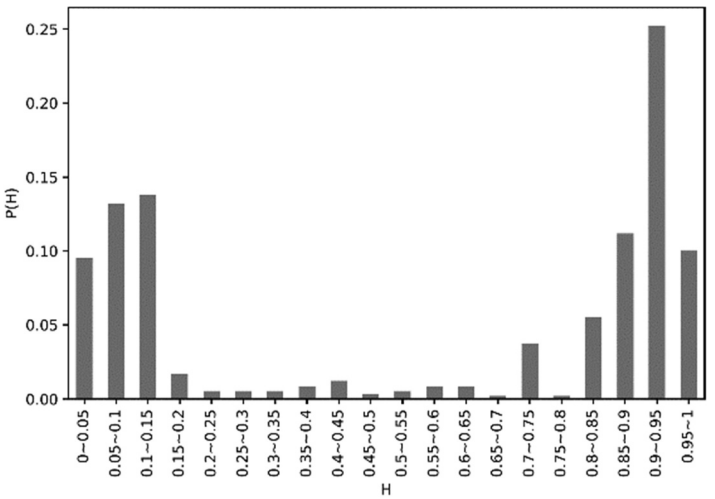

Figure 4. Distribution of output value of hidden layer.

When the data corresponding to the above rule passes through the artificial neural network, the output value of the hidden layer is obtained as the result shown in Table 4.

Table 5 shows that there are many values corresponding to 0.3∼0.7. It can be assumed that this leads to uncertainty problems of the rules and that the accuracy of the rules may be low. Next, we looked at the rule with 100% accuracy to see the opposite case.

If (6.3 < sepal-length) and (2.7 < petal-length < = 5) and not (1.6 < petal-length) then versicolor 1.00

Table 6 shows the output value of the hidden layer when the data corresponding to the above rule passes through the artificial neural network. From the output value, it can be confirmed that there is no corresponding value between 0.3 and 0.7.

In order to confirm the above results more precisely, the output value of the hidden layer when the data corresponding to all the rules with the accuracy less than 75% passes through the artificial neural network and the output value of the opposite case are shown as distribution in Figure 5.

Figure 5 shows that the output value is mainly in the range of 0.3∼0.7 while the value between 0.3 ∼ 0.7 is completely excluded in the Figure 6. As a result of this experiment, it was confirmed that if the output value of the hidden layer

Table 4. Structure of artificial neural network.

Extracted rules using OAS algorithm
If (petal-length ≤ 2.7) and (petal-width ≤ 0.7) then setosa 1.000
If (sepal-length ≤ 5.4) and not (2.7 < petal-length ≤ 5) and (petal-width ≤ 0.7) then setosa 1.000
If not (6.3 < sepal-length) and not (2.7 < petal-length ≤ 5) and (petal-width ≤ 0.7) then setosa 1.000
If not (2.7 < petal-length ≤ 5) and not (5 < petal-length) and (petal-width ≤ 0.7) then setosa 1.000
If (2.7 < petal-length ≤ 5) and (0.7 < petal-width ≤ 1.6) then versicolor 0.979
If (2.7 < petal-length ≤ 5) and not (petal-width ≤ 0.7) and not (1.6 < petal-length) then versicolor 0.979
If not (sepal-length ≤ 5.4) and (2.7 < petal-length ≤ 5) and not (petal-width ≤ 0.7) then versicolor 0.843
If not (petal-length ≤ 2.7) and not (5 < petal-length) and (0.7 < petal-width ≤ 1.6) then versicolor 0.979
If (5 < petal-length) and (1.6 < petal-length) then virginica 1.000
If (sepal-width ≤ 2.8) and (1.6 < petal-length) then virginica 1.000
If not (petal-length ≤ 2.7) and not (2.7 < petal-length ≤ 5) and (1.6 < petal-length) then virginica 1.000
If not (3.1 < sepal-width) and not (2.7 < petal-length ≤ 5) and (1.6 < petal-length) then virginica 1.000

Table 5. Hidden layer output value of data corresponding to rule with accuracy of 20%.

Node 1	Node 2	Node 3	Node 4
0.103	0.423	0.539	0.485
0.059	0.295	0.692	0.538
0.128	0.340	0.674	0.317
0.05	0.229	0.785	0.421
0.05	0.229	0.785	0.421
0.05	0.229	0.785	0.421
0.05	0.229	0.785	0.421
0.069	0.310	0.691	0.458
0.069	0.310	0.691	0.458
0.05	0.229	0.785	0.421

Table 6. Hidden layer output value of data corresponding to rule with accuracy of 100%.

Node 1	Node 2	Node 3	Node 4
0.067	0.859	0.087	0.950
0.067	0.859	0.087	0.950
0.032	0.788	0.154	0.944
0.044	0.711	0.229	0.936
0.032	0.788	0.154	0.944
0.044	0.788	0.154	0.944
0.044	0.788	0.154	0.944
0.044	0.788	0.154	0.944
0.032	0.711	0.229	0.936
0.044	0.788	0.154	0.944

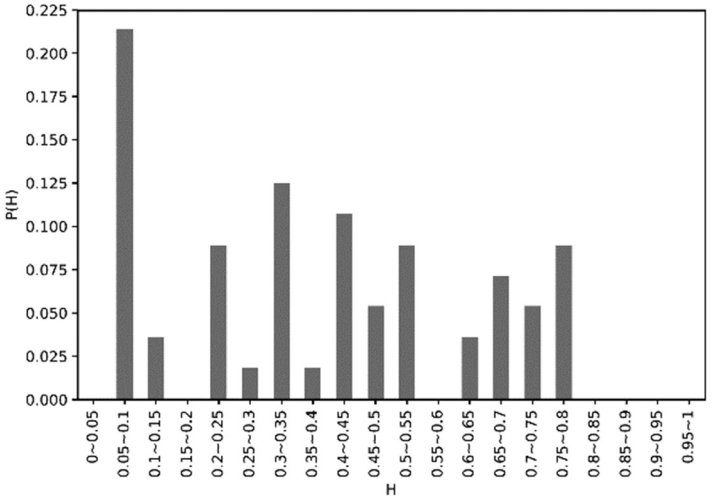

Figure 5. Distribution of hidden unit output values of rule with accuracy less than 75%.

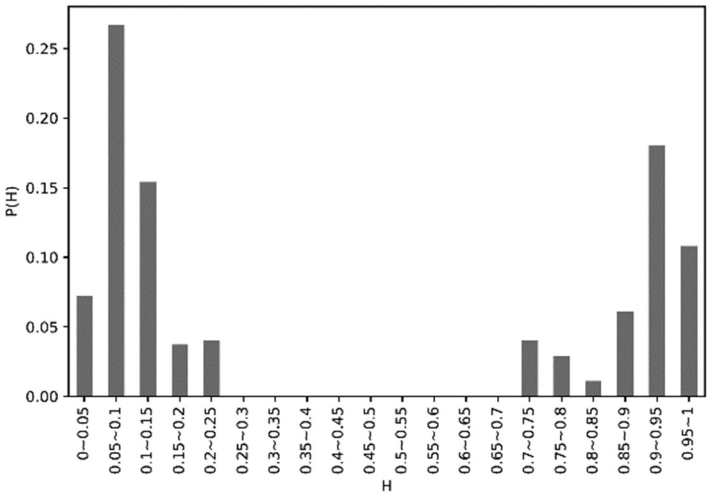

Figure 6. Distribution of hidden unit output values of rule with accuracy more than 75%.

is not binarized clearly, the probability of extracting the wrong rule due to the uncertainty problem of the rule is high, and in the opposite case, it is confirmed that the rule with high accuracy is extracted.

The Result of Applying the Hidden Unit Clarification

The results show that the hidden layer output value affects the accuracy of the rule due to the uncertainty problem of the rules, and it is found that improving this part is an important part of rule extraction. Next, we have experimented with applying hidden function clarification to improve this problem. We applied the hidden unit clarification to the trained neural network and trained more 2000 epoch. As a result, the distribution of the hidden layer output value changed as shown in Figure 7 without affecting the performance of the artificial neural network.

The distribution of Figure 7 shows that the value of 0.3∼0.7 corresponding to the median value is reduced compared to the previous result, and the value is shifted toward the complete activation or complete deactivation, and the binarization is better than before.

Next, we observed how the number of rules extracted and the accuracy change with c value while changing parameter c, which is the penalty term weight of the cost function. As shown in Figure 8, when the parameter c is 0, it is the value when the hidden unit clarification is not applied. The graph shows that the number of rules gradually decreases as c increases. When c was 1.6, the number of rules was reduced from 88.12 to 61.62, which is about 30% less than when the hidden unit clarification was not applied. If the accuracy and coverage decrease as the number of rules decreases, the experimental results may

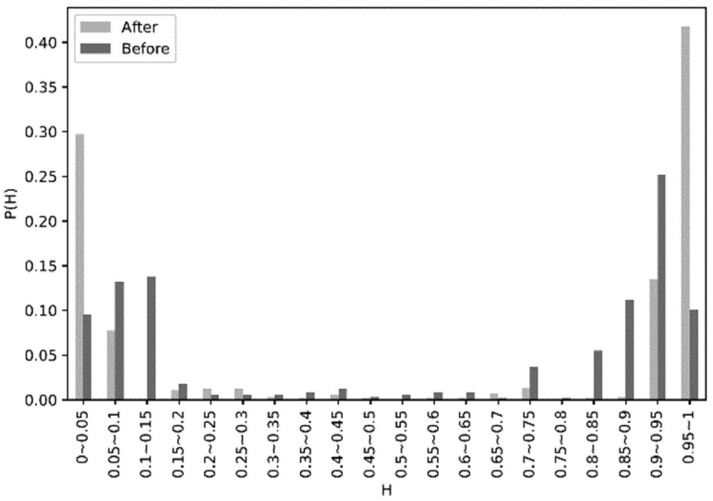

Figure 7. Comparison before and after application of hidden unit clarification.

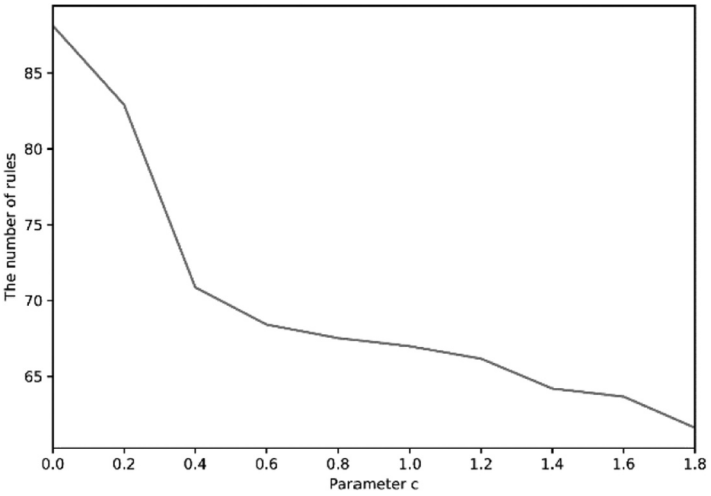

Figure 8. Graph of change in number of rules according to change of c.

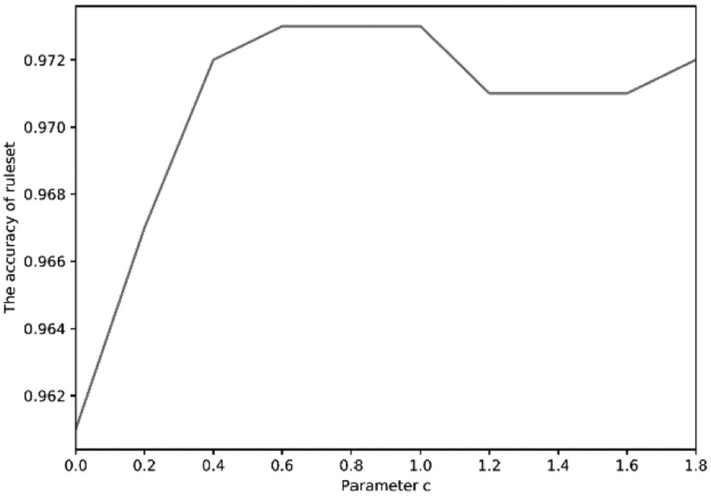

Figure 9. Graph of change in accuracy of ruleset according to change of c.

not be meaningful, but the coverage does not change at 100%, and the accuracy improves as c increases as shown Figure 9.

Figure 9 shows that the accuracy is increased when applying the hidden unit clarification. Based on the experimental results, it was confirmed from the experimental results that it is possible to make more precise and good rules by using additional training through hidden unit clarification when OAS algorithm is used.

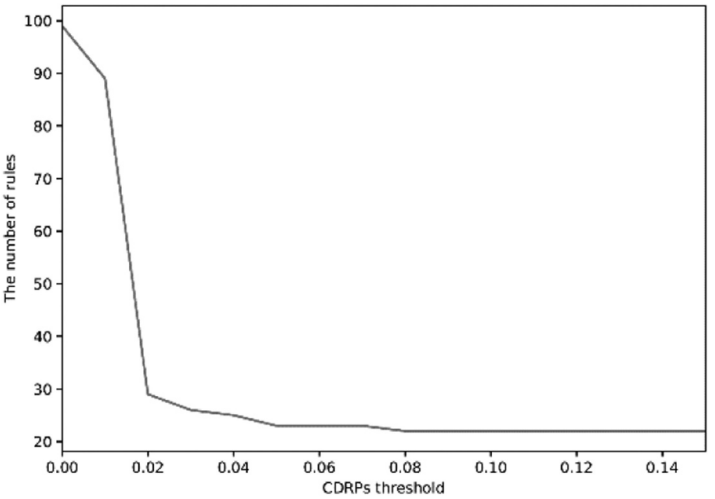

Figure 10. Graph of change in number of rules according to CDRPs threshold.

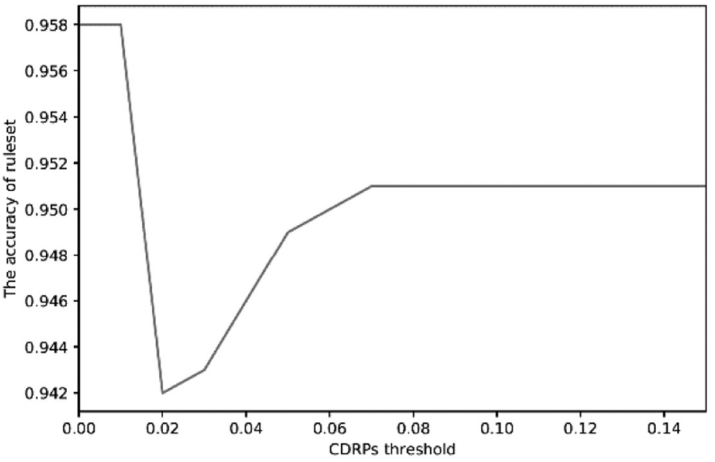

Figure 11. Graph of change in accuracy of rules with changes in CDRP threshold.

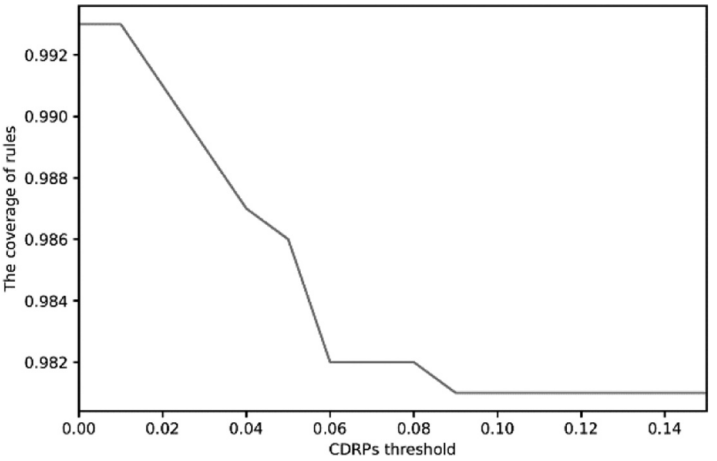

Figure 12. Graph of change in coverage of rules with changes in CDRP threshold.

The Result of Applying CDRPs

The application of CDRPs experiment was repeated 100 times and the mean value was calculated. We increased CDRPs threshold from 0 to 0.15 by 0.01. When the CDRPs threshold was zero, CDRPs were not used.

Figure 10 shows that the number of rules is reduced from 99 to 22 as the CDRPs threshold increases. The decrease in the number of rules was highest when the CDRPs threshold changed from 0.01 to 0.02.

As shown in Figure 11, as the CDRPs increase, the rule accuracy initially dropped from 0.958 to 0.942, but then rose again to 0.951. The accuracy of the rule was decreased by 0.007 and it was observed that it did not fall much.

As shown in Figure 12, the coverage of the rule is slightly decreased from 0.993 to 0.981 according to the CDRPs threshold, and the decrease is small.

Overall, the accuracy and coverage of the rules decreased slightly according to the CDRPs threshold, but the average number of rules decreased from 99 to 22, so it can be said that it is meaningful results. For the IRIS domain, the best overall result was obtained when the CDRPs threshold was 0.07.

Result and Future Work

In this study, we used the OAS algorithm, which is one of the rule extraction algorithms that make the artificial neural network a rule that can be understood by humans. We extracted the rules from the trained neural network through the OAS algorithm and analyzed the characteristics of the rules that are less accurate. As a result, it was found that the distribution of the output values of each hidden layer is much in the range between 0.3 and 0.7 when the data corresponding to the rule with less accuracy pass through the trained artificial neural network. The presence of a large number of these intermediate values is likely to cause problems when the rules are extracted using a decompositional aapproach, suchas the OAS algorithm because the output values were not binarized and it caused uncertainty problem of the rules.

In order to improve this problem, we applied the hidden unit clarification to the artificial neural network, and experimented how the result of the hidden layer output value changes and the quality of the rule improves. As a result, applying of the hidden unit clarification reduced the uncertainty problem of the rule by making the output value from the hidden unit close to the full activation or deactivation, and more binarized it, reduced the number of unnecessary rules. Based on these results, it was confirmed that the rules can be efficiently extracted when applying the hidden unit clarification, and it is confirmed that the artificial neural network is trained in a form that is more easy to understand.

And we extracted CDRPs from the trained neural network to find non-critical nodes and removed the corresponding rules, as a result, it was possible to reduce the number of rules to 22 on average.

There are two major plans in the future work. First, in this study, we experimented with one hidden layer due to the problem of decompositional approach. In the future, we will try to extract the rule of the artificial neural network with deep hidden layer and will try to find out how to improve this. Second, in this paper, we have experimented with OAS algorithm, hidden unit clarification algorithm, and CDRPs, in the future, we plan to study the possibility of improvement by experimenting with more various combinations of algorithms.

References

Andrews, R., J. Diederich, and A. B. Tickle. 1995. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-based Systems* 8 (6):373–89. doi:10.1016/0950-7051(96)81920-4.

Fu, L. 1991. Rule learning by searching on adapted nets. In *AAAI*. vol. 91, 590–95.

Fu, L. 1994. Rule generation from neural networks. *IEEE Transactions on Systems, Man, and Cybernetics* 24 (8):1114–24. doi:10.1109/21.299696.

Fu, L.-M. 1993. Knowledge-based connectionism for revising domain theories. *IEEE Transactions on Systems, Man, and Cybernetics* 23 (1):173–82. doi:10.1109/21.214775.

Hailesilassie, T. 2016. Rule extraction algorithm for deep neural networks: A review. *(IJCSIS) International Journal of Computer Science and Information Security*, 14(7):376–381.

Ishikawa, M. 1996. Structural learning with forgetting. *Neural Networks* 9 (3):509–21. doi:10.1016/0893-6080(96)83696-3.

Kim, H. (2000). Computationally efficient heuristics for if-then rule extraction from feed-forward neural networks. In *Lecture Notes in Artificial Intelligence 1967 (Discovery Science 2000)*, 170–182. Springer.

Schmitz, G. P., C. Aldrich, and F. S. Gouws. 1999. Ann-dt: An algorithm for extraction of decision trees from artificial neural networks. *IEEE Transactions on Neural Networks* 10 (6):1392–401. doi:10.1109/72.809084.

Setiono, R., and W. K. Leow. 2000. Fernn: An algorithm for fast extraction of rules from neural networks. *Applied Intelligence* 12 (1–2):15–25. doi:10.1023/A:1008307919726.

Taha, I. A., and J. Ghosh. 1999. Symbolic interpretation of artificial neural networks. *IEEE Transactions on Knowledge and Data Engineering* 11 (3):448–63. doi:10.1109/69.774103.

Thrun, S. 1995. Extracting rules from artificial neural networks with distributed representations. In G. Tesauro, D. Touretzky, and T. Leen (Eds.), *Advances in Neural Information Processing Systems*, Volume 7. MIT Press.

Towell, G. G., and J. W. Shavlik. 1993. Extracting refined rules from knowledge-based neural networks. *Machine Learning* 13 (1):71–101. doi:10.1007/BF00993103.

Wang, Y., H. Su, B. Zhang, and X. Hu. 2018. Interpret neural networks by identifying critical data routing paths. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8906–8914.