Generating Interpretable Prototype Networks by Comprehensive Compression for Multi-Layered Neural Networks

Ryotaro Kamimura
Tokai University
Hiratsuka, Kanagawa, 259-1292, Japan
email: ryotarokami@gmail.com

***Abstract*—The present paper aims to propose a method for creating an interpretable prototype network that is hidden within original multi-layered neural networks. The interpretation of the inference mechanism of neural networks has received much attention in recent times, leading to the development of various methods. However, these methods have focused on interpreting specific internal representations created by neural networks. There is an urgent need to propose an interpretation method that considers not only specific representations but also all internal representations created by a neural network, aiming for a more unified understanding of the fundamental properties of the inference mechanism. To address this problem, we propose the introduction of a prototype network that is hidden within the original multi-layered neural network. This is achieved through an interpretation method called "comprehensive compression," which aims to replace the process of interpretation for finding a simple and interpretable prototype network. The method was applied to the analysis of customer data sets. The experimental results demonstrate that interpretable compression can simplify multi-layered neural networks and unify all obtained representations. It enables the detection and interpretation of non-linear as well as corresponding linear relations. The proposed method of the prototype network makes it possible to interpret not only specific instances but also a number of different instances. It helps us uncover the fundamental inference mechanism that is deeply hidden within neural networks.**

***Keywords*—prototype network; comprehensive compression; interpretable compression; layer compression; collective compression; stabilizing compression; total de-compression; selective compression**

I. Introduction

A. Problems of Interpretation

As neural networks have been applied in many fields because of their improved generalization performance, the problem of difficulty in interpreting their internal representations has received much attention these days [1], due to reliability [2] and ethical problems [3] [4]. Though much progress has been made in the field of convolutional neural networks (CNN), where the intuitive interpretation of image datasets is fairly easy [5]–[8], the interpretation methods developed so far are far from being acceptable, particularly when dealing with more abstract objects in social and human sciences, where the intuitive interpretation of data sets is almost impossible.

One of the main problems is that the majority of interpretation methods have tried to interpret an instance of the final results obtained by learning. A slight consideration of these types of methods leads us to doubt their validity because neural networks have had a bad reputation from the beginning of research that the final results can be easily changed by modifying initial conditions, input patterns, parameters, network configurations, etc., which have recently received much attention in the name of adversarial attacks [9]–[13]. In addition, one of the more fundamental problems is that while we should aim to explain why and how a method can reach its conclusion as rigorously as possible, it only tries to describe the phenomena occurring during learning in a very specific manner. This specific interpretation is naturally not enough to understand the main inference mechanism of neural networks, even if it can explicitly and fully explain the inference.

Naturally, extensive studies have been conducted on the so-called "global" interpretation [14] [15], and these studies seem to deal with the inference mechanism broadly. However, they do not necessarily explain the real global properties of neural networks. As mentioned above, one of the main properties of neural networks lies in their productive property, where a neural network can generate a large number of different internal representations by changing learning conditions. Although many of these representations may not be understandable to observers, they should still be interpreted because networks without interpretable representations can still produce the appropriate final conclusions. This productive property has not been fully discussed in the field of neural networks, with only the acknowledgment that many different representations can be generated with different initial conditions and situations. In this framework of neural network productivity, the so-called "global" methods seem to be limited to specific instances among many, which should be called "specific" or "local" interpretation.

Furthermore, considering the fact that neural networks were developed to oppose rule-based systems, many global interpretation methods with logical and semantic-based interpretation [16]–[18] seem to have serious problems from the beginning. We should move away from explanations based on logical rules and explore different methods for understanding the productive property of neural networks.

B. Prototype Network

In this context, we aim to introduce a new type of interpretation method to clarify the objective of interpretation. In this method, the interpretation is based on detecting a prototype and interpretable network, which is assumed to be hidden within actual multi-layered neural networks. In existing

 ISBN: 978-1-68558-046-9

interpretation methods, the prototype is used to facilitate the interpretation process, where neural networks attempt to classify input images based on the prototypes created in the prototype layers [19]. However, we propose that behind many multi-layered neural networks, there always exists a simple prototype network. These multi-layered neural networks are assumed to be generated by transformation rules from the prototype network, and the productive property can be explained by this transformational generation. Furthermore, because the prototype network is assumed to be simple enough to interpret the meaning of any components within it, the interpretation lies not in directly interpreting the original multi-layered neural networks but in uncovering the hidden prototype network. Thus, the abstract and intuitive process of interpretation can be replaced by a more concrete process of finding a prototype network.

C. Interpretable and Stabilizing Compression

The prototype network is supposed to be as simple as possible. For example, it can be imagined that the prototype network can be represented by a network without hidden layers, and the interpretation method should aim to find this prototype network without hidden layers for interpretation. This network is realized by simplifying multi-layered neural networks to the extreme. Thus, interpretation, in the first place, is about simplifying multi-layered neural networks as much as possible to approximate the prototype network from a technical viewpoint. To find the prototype network, we need to compress multi-layered neural networks to the simplest ones without hidden layers. Additionally, we need to compress all representations created by learning with different initial conditions, input patterns, parameters, and network configurations for interpretation. These types of compression can be called "interpretable compression." It should be stressed that interpretable compression is different from conventional compression methods [20]–[22] in that the original information in internal representations should be preserved as much as possible in the compressed ones.

In addition, the number and intensity of components in a neural network should be reduced in order to simplify and stabilize the interpretable compression. This reduction aims to restrict the flexibility of components. This compression is referred to as "stabilizing compression." It is closely related to conventional regularization methods such as weight decay [23] [24] and pruning [25] [26] as it aims to restrict the productive property of neural networks. However, because the obtained prototype network is expected to generate a considerable number of multi-layered networks, which is directly related to the productive property at the surface level, it should aim to simplify representations while preserving the productive property as much as possible. To achieve this, we introduce a process of compression accompanied by decompression, which controls the productivity of neural networks for smooth learning.

D. Main Contributions

Main contributions are summarized as follows:

- The present paper proposes a new interpretation method that replaces the interpretation process for finding a prototype and interpretable network.
- This network is obtained through interpretable compression by simplifying or compressing as many multi-layered neural networks as possible.
- Although this interpretable compression naturally entails some instability due to the productive property of neural networks, it can be mitigated by introducing stabilizing compression.
- The paper demonstrates how our method successfully produces a prototype network with several important inputs that were considered unimportant by conventional methods, serving as an example of interpretation.

E. The Structure of the Paper

In Section 2, we explain how to compress multi-layered neural networks for interpretation, using layer compression and stabilizing compression. Layer compression gradually eliminates hidden layers, while stabilizing compression includes both total de-compression to control the strength of weights and selective compression to select strong weights. In Section 3, we present the experimental results on the interpretation of a customer data set. We demonstrate how layer and stabilizing compression can be used to clarify relations between inputs and outputs and to distinguish between linear and non-linear relations.

II. Theory and Computational Methods

A. Concept of Compression

In this paper, the interpretation aims to detect a prototype network supposed to be hidden in multi-layered neural networks. Figure 1 shows our basic framework of interpretation. Firstly, it is supposed that an actual multi-layered neural network is generated from the corresponding prototype network by decompression or by deploying it to a multi-layered network, as shown on the left side of Figure 1. One of the major challenges is to find this prototype network, which we refer to as the "interpretation," because the prototype network is supposed to be as simple as possible. To find this prototype network, we must compress an actual multi-layered neural network to the extreme point as shown on the right side of Figure 1, because simplicity must be one of the most important inherent properties of the prototype network. The present paper focuses on this compression to find a prototype network. For actual implementation, this compression should be elaborated further to deal with actual and practical learning processes.

We introduce a framework shown in Figure 2 to make the basic framework in Figure 1 practically applicable. One of the main differences is the introduction of comprehensive compression instead of simple compression. In the comprehensive compression component in Figure 2, two types of compression are introduced: interpretable (a) and stabilizing (b) compression. In the interpretable compression, two

 ISBN: 978-1-68558-046-9

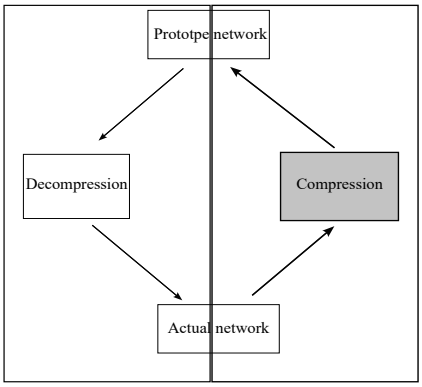

Figure 1. A framework to find a prototype network, which is decompressed into an actual network configuration, while an actual network is compressed into the prototype network.

types of compression, layer and collective compression, are introduced. In layer compression, hidden layers are gradually eliminated to find a network without hidden layers. Collective compression is used to compress all representations created by a neural network with different initial conditions, input patterns, parameters, and network configurations. Then, stabilizing compression is introduced to simplify and stabilize multi-layered neural networks for producing an easily interpretable network. In this stabilizing compression, we introduce total de-compression and selective compression. In total de-compression, the strength or magnitude of total weights is mainly decreased, and to stabilize this process, we introduce the decompression of the strength of total weights, meaning that the reduced strength is restored. Thus, we call this process "de-compression" to combine a process of compression and decompression. The selective compression aims to simplify a network configuration by selecting important connection weights in the corresponding hidden layers.

B. Interpretable Compression

1) Compression for Interpretation: The prototype network can be estimated by producing and compressing many different types of multi-layered neural networks with different initial conditions, input patterns, parameters, and so on, which is called "interpretable compression," aiming to produce the simplest network or collective network to approximate the prototype network. In interpretable compression, we have two types of compression: collective compression and layer compression. Layer compression is used to reduce the number of hidden layers, and collective compression aims to unify all possible representations created by learning.

Figure 3 shows the concept of interpretable compression. In the first place, we try to produce many different types of networks with different initial conditions (a), input patterns (b), and different parameter values (c), among others. This is necessary because the fundamental property related to the prototype network can be estimated by viewing data sets from many different viewpoints. The obtained estimated networks are collectively unified or averaged in the collective

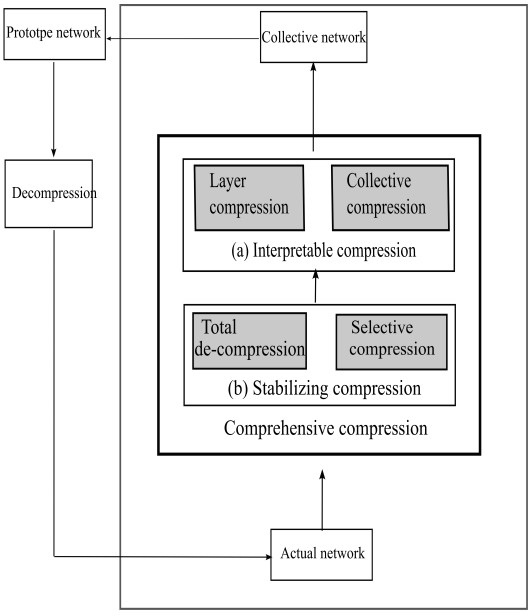

Figure 2. Actual compression components in which simple compression is substituted for comprehensive compression in the practical implementation with stabilizing and interpretable compression inside.

compression (d) to create collective networks (e) which are used to estimate the corresponding prototype network (f). The difference between the two networks can be detected, albeit roughly, by the ratio (u/z) (e) of the actual absolute coefficients of the collective network (u) to the absolute coefficients (z) of the prototype network. The absolute values are used to roughly understand the meaning of prototype networks at the present stage. Since it is actually impossible to use the coefficients of the prototype network (z), we should employ other measures such as linear correlation coefficients between inputs and targets obtained through regression analysis.

2) Layer Compression: Let us illustrate the process of layer compression in Figure 4(a)-(e). First, we compress the connection weights from the first to the second layer, denoted by (1,2), and from the second to the third layer (2,3) for an initial condition and a subset of a dataset. Then, we obtain the compressed weights between the first and the third layer, denoted by (1,3).

$$w_{ik}^{(1,3)} = \sum_j w_{ij}^{(1,2)} w_{jk}^{(2,3)} \tag{1}$$

These compressed weights are further combined with the weights from the third to the fourth layer (3,4), resulting in the compressed weights between the first and the fourth layer (1,4).

$$w_{il}^{(1,4)} = \sum_k w_{ik}^{(1,3)} w_{kl}^{(3,4)} \tag{2}$$

By repeating these processes, we obtain the compressed weights between the first and the fifth layer, denoted by $w_{iq}^{(1,5)}$.

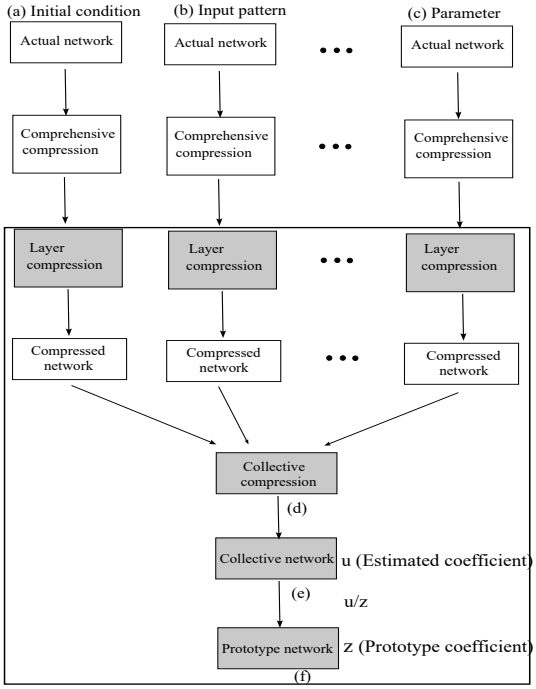

Figure 3. Compression from multi-layered networks with different initial conditions (a), input patterns (b), and parameters (c) to the simplest collective network (e) to estimate the corresponding prototype network (f).

Using these connection weights, we have the final and fully compressed weights (1,6).

$$w_{ir}^{(1,6)} = \sum_{q} w_{iq}^{(1,5)} w_{qr}^{(6,7)} \tag{3}$$

3) Collective Compression: Figure 4(d) shows that all the finally compressed weights are averaged to obtain the final collective weights. Then, c represents the real coefficients of a prototype network in Figure 4(e). Similarly, z denotes the absolute and individual compression coefficients of the prototype network.

$$z_{ir} = \mid c_{ir} \mid \tag{4}$$

These coefficients should be compared with the corresponding coefficients or weights of the collective weights.

$$u_{ir} = \mid w_{ir} \mid \tag{5}$$

Then, we define the collective compression coefficient as follows:

$$c_{ir} = \frac{u_{ir}}{z_{ir}} \tag{6}$$

Here, the denominator represents the absolute values of the basic relations between inputs and targets in a prototype network. The use of absolute values allows us to intuitively observe the overall relations without considering the sign information. This coefficient aims to indicate which inputs are larger than the corresponding basic relations. Introducing this coefficient is crucial for interpreting the finally compressed weights since interpreting the compressed weights on their own is not possible. Interpretation can only be achieved by comparing them to other interpretation results. Several possibilities for these basic relations include the correlation coefficient between inputs and targets of the original data set, regression coefficients from regression analysis, weights obtained through conventional methods, and so on. When the collective weights significantly deviate from these relations obtained by conventional methods, it becomes necessary to provide an explanation for the deviation.

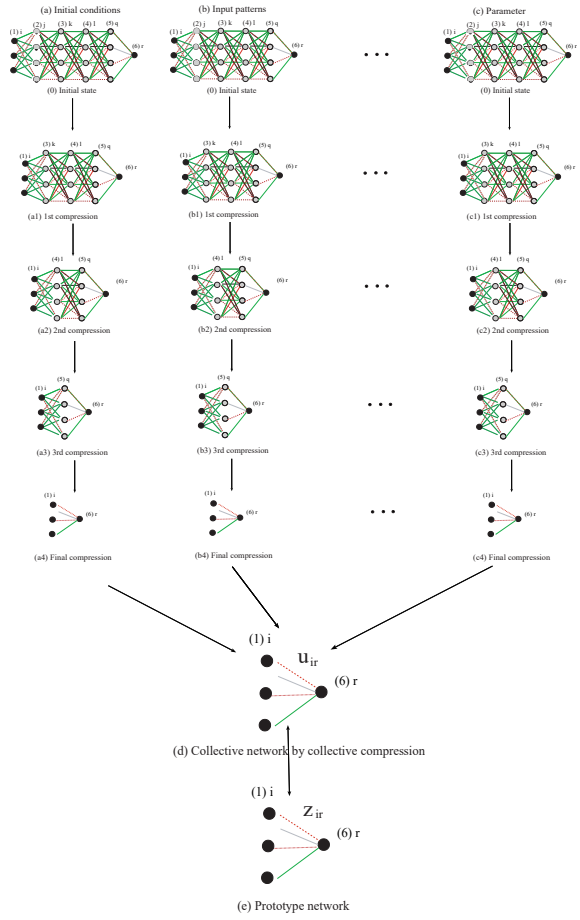

Figure 4. Layer compression from multi-layered networks (1) to the simplest networks (4) with different initial conditions (a), different inputs (b), and different parameters (c), and collective network (d) and the corresponding prototype network (e).

C. Stabilizing Compression

1) Stabilizing Concept: We will explain stabilizing compression within the framework of comprehensive compression. It is necessary to simplify the actual network to an extreme point in order to find the prototype network. This simplification and restriction of possible network configurations are associated with the stability of the produced networks and their internal representations.

In compression, we can identify two types: total decompression (compression and decompression) and selective compression. Total compression aims to reduce the strength of

total weights, while total decompression aims to increase the strength. The purpose of total decompression is to enhance the effectiveness of total compression. On the other hand, selective compression is used to minimize the number of connection weights for simplification. These types of compression should be performed simultaneously when using the conventional learning approach. However, these types of compression and error minimization are contradictory. For instance, total compression and decompression are completely contradictory to each other, and it is impossible to perform them simultaneously.

For solving this type of contradiction, we introduce the serially disentangled stabilizing compression. In this method, all compression procedures and error minimization are completely disentangled, and they are independently applied. Firstly, the total compression is applied to reduce the strength of total weights, and then errors between outputs and targets are reduced. Then, the selective compression is applied, followed by the corresponding error minimization. Finally, the total decompression is used to weaken the effects of total compression, followed by the corresponding error minimization. In this way, completely contradictory terms such as total compression and decompression can coexist, though seemingly.

2) Total de-compression: Let us define total decompression by considering the strength of weights. Note that the compression in this paper is different from or contrary to the conventional compression methods using the compression rate. The total compression can be defined by measuring the strength of the absolute weights. When the absolute weights become smaller, the actual degree of compression becomes larger because information should be represented by the smaller strength of weights, where the flexibility of weights becomes smaller.

On the contrary, when the absolute strength becomes larger, the actual degree of compression becomes smaller because larger weights have a possibility to represent many different types of weight configurations.

For simplicity, we consider weights from the tth hidden layer to the $t+1$th hidden layer, represented by $(t, t+1)$, and the absolute weight from the jth neuron to the kth neuron is defined by

$$u_{jk}^{(t,t+1)} = \mid w_{jk}^{(t,t+1)} \mid \tag{7}$$

We need to use the total weight strength, summed over all connection weights in all hidden layers. Then, averaging this total weight strength, we have the average total compression coefficient, computed by

$$\bar{U} = \frac{\theta}{h \cdot n_t \cdot n_{t+1}} \sum_{t}^{h} \sum_{j}^{n_t} \sum_{k}^{n_{t+1}} u_{j,k}^{(t,t+1)} \tag{8}$$

where h denotes the number of hidden layers minus one, and n_t is the number of neurons in the tth hidden layer. In the following experimental results, the parameter θ should be positive, and it should be decreased gradually.

3) Selective Compression: Then, we should count the number of strong weights in a layer as an index for representing the compression in a hidden layer, meaning that we must select important connection weights in the selective compression. Here, we also consider the absolute strength of weights, but they are normalized by the corresponding maximum absolute weight. This relative strength can be computed by

$${}^{\gamma}g_{jk}^{(t,t+1)} = \left[\frac{u_{jk}^{(t,t+1)}}{\max_{j'k'} u_{j'k'}^{(t',t'+1)}} \right]^{\gamma} \tag{9}$$

where the max operation is over all connection weights in the layer, and the parameter γ should be a small positive value for stabilizing learning. When this equation is applied to connection weights, a winning connection weight in terms of weight strength remains the same, while all the other weights are pushed toward smaller ones. In an extreme case, only one winning weight has some strength, while all the others become zero, which is the well-known hard type WTA (winner-take-all).

By using the relative strength over all hidden layers, we have the average selective compression coefficient, defined by

$${}^{\gamma}\bar{G} = \frac{1}{h \cdot n_t \cdot n_{t+1}} \sum_{t}^{h} \sum_{j}^{n_t} \sum_{k}^{n_{t+1}} \left[\frac{u_{jk}^{(t,t+1)}}{\max_{j'k't'} u_{j'k't'}^{(t,t+1)}} \right]^{\gamma} \tag{10}$$

This average selective compression coefficient becomes maximum when all connection weights have the same values. When only one connection weight is larger than zero, while all the others are zero, the corresponding coefficient becomes minimum, supposing that at least one weight should be larger than zero.

III. Results and Discussion

A. Customer Data Set

The experiments were performed, using a data set with a sport gymnasium's customers. We tried to discriminate between genders, and to infer the characteristics of female customers [27]. The data set was composed of 2104 gymnasium customers, and the inputs were composed of eight variables. We used networks with ten hidden layers with ten neurons for each hidden layer. The scikit-learn neural network package was used with default parameter values except the activation function (set to the tangent-hyperbolic) and the number of learning steps (set to 600 learning steps) for the easy reproduction of the experimental results.

In the experiment, a multi-layered neural network with ten hidden layers are compressed for interpretation with the stabilizing compression, composed of total de-compression and selective compression. This network is compressed (layer compression) into networks without hidden layer for all learning steps. All those compressed networks are collectively compressed in the collective compression. Finally, we tried to compare this collective network with the corresponding prototype network to examine the characteristics obtained by our compression method.

 ISBN: 978-1-68558-046-9

B. Interpretable Compression

The interpretable compression consists of layer compression and collective compression. In the layer compression, the network with ten hidden layers is compressed gradually into the corresponding network without hidden layers. The connection weights or coefficients of the estimated network were compared to those of the prototype network. We here used the correlation coefficients between inputs and outputs as an example of coefficients of the prototype network. The comparative study between the estimated network and the prototype network in terms of correlation coefficients clarified several important inputs that could not be identified by the simple linear correlation coefficients.

1) Layer and Collective Compression: Figure 5 shows collective weights (1), ratios of absolute collective weights (2) to the corresponding absolute correlation coefficients between inputs and targets (3) by the conventional method (a) and by new methods where the parameter θ decreased from 1.0 (b) to 0.6 (f). As shown on the left side of Figure 5, the input No.8 (daytime use) was the largest by all the methods, while all the others were considerably smaller in terms of correlation coefficients and connection weights.

Then, we examined the ratios of absolute connection weights to the corresponding correlation coefficients, represented in the middle of Figure 5. The correlation coefficients between the ith input to the corresponding target were represented by v_i. Then, the absolute and individual compression coefficients of the prototype network were represented by z_i:

$$z_i = \mid v_i \mid \tag{11}$$

These values should be compared with the weights of collective weights. Then, we have the ratio of the absolute connection weights to the corresponding correlation coefficients as:

$$c_i = \frac{u_i}{z_i} \tag{12}$$

The ratios showed that when the inputs on the left side became relatively larger, these relations between the inputs and targets could not be extracted by linear correlation coefficients. However, the ratios slightly differed when the parameter was decreased from 1 (b) to 0.6 (f). When the parameter decreased from 1.0 (b) to 0.6 (f) in Figure 5, gradually the input No.2 became larger than the others. Finally, we should average compressed weights for really collective meaning of the connection weights, which was the final round of interpretable compression.

Figure 6 shows the final collective weights averaged over all weights obtained by different parameter values, shown in Figure 5. As seen in the figure, the female clients tended to use the gym during daytime (input No.8) but relatively irregularly (input No.2) in terms of negative values, and the usage period tended to be longer (input No.3). Our compression method could clarify the characteristics of female clients. In addition to the linear relation represented in the input No.8 (daytime use), non-linear relations were detected in the input No.2 (irregularity) and No.3 (usage period). This means that the female clients tend to use longer the gym in daytime, but in an irregular manner.

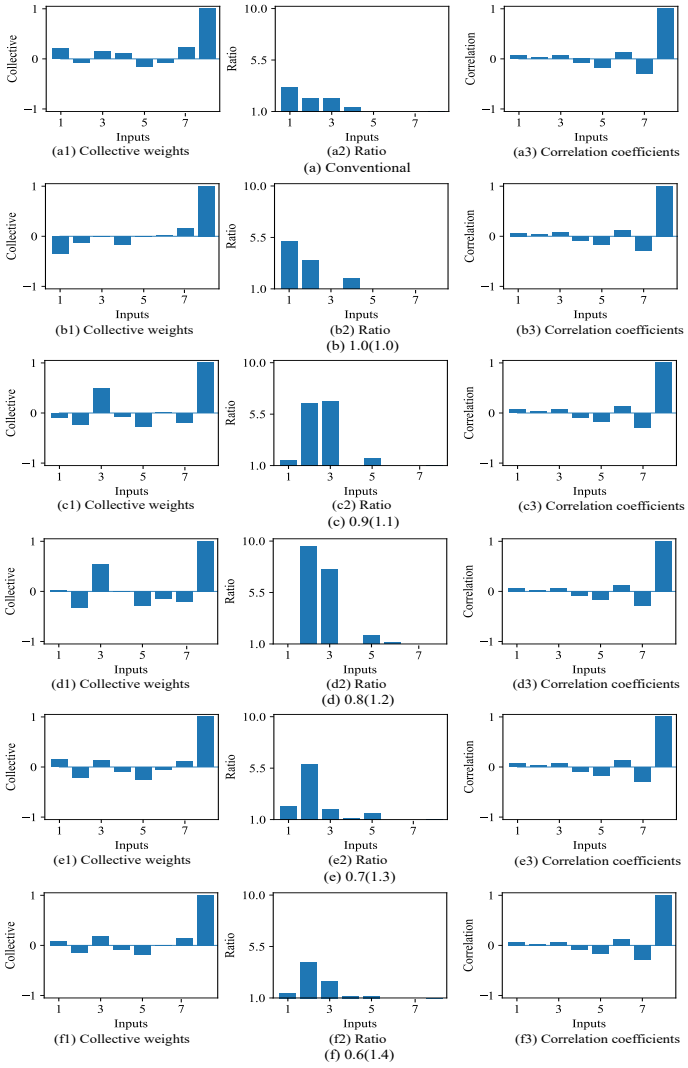

Figure 5. Collective weights (1), ratio (2) and correlation coefficients (3) by the conventional method (a), and by decreasing the parameter θ from 1.0 (b) to 0.4 (f) for the customer data set.

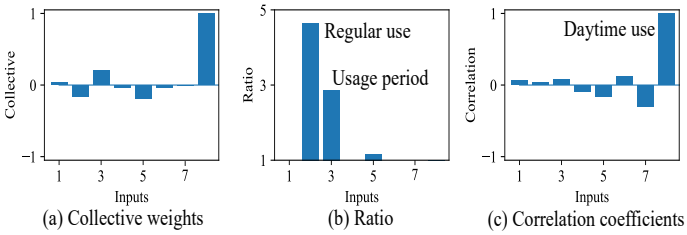

Figure 6. Collective weights (a), ratio (b) and correlation coefficients (c) by averaging all the collective weights for the customer data set.

C. Stabilizing Compression

1) Stability of Selective Compression: By decreasing the parameter θ in total compression, accompanied by increasing

the parameter θ in total decompression, average total compression coefficients decreased with large fluctuations. However, the selective compression coefficients decreased almost without fluctuations to have an effect to simplify the corresponding network configurations in terms of number of connection weights. This means that the stabilized simplification was realized by this compression in spite of large variations of total compression coefficients.

Figure 7 shows total (1) and selective (2) compression coefficients. Figure 7(a) shows the results by the conventional methods without compression. As can be seen in the figure, total and selective compression coefficients remained to be unchanged throughout all learning steps. When the parameter θ decreased from 1.0 (b) to 0.6 (f), two compression coefficients gradually decreased. In spite of decrease in the parameter to have an effect to decrease the strength of weights, the intensity of each compression coefficient tended to increase, when the parameter decreased from 1.0 (b) to 0.6 (f). This effect could be explained by the fact that the parameter for decompression was inversely set to $2 - \theta$. Total compression coefficients decreased with large fluctuations by the effects of compression and decompression in the total de-compression. However, in spite of these fluctuations, the selective compression coefficients decreased almost without fluctuations.

The results show that total de-compression had the effect to prevent connection weights from being too small by increasing the strength of connection weights by total decompression. In addition, the large fluctuations by the effects of total de-compression had no effects on the process of decreasing the selective compression coefficients. In sum, the selective compression can be effective in decreasing the strength of connection weights, which was performed very smoothly by the effect of total de-compression. The smooth decrease in the selective information by the help of total de-compression can be used to realize actually the simplicity of network configurations in terms of the number of connection weights.

2) Weight Stability : The results showed that an decrease in the parameter θ was related to the stable acceleration of learning. This explains why networks with different parameter values could produce similar performance in terms of generalization.

Figure 8 shows connection weights, obtained with 50 learning steps using the conventional method (a), when the parameter decreased from 1.0 (b) to 0.6 (f). In Figure 8(a), when using the conventional method, almost all weights were random, making it impossible to discern any regularity inside. When the parameter decreased from 1.0 (b) to 0.6 (f), the characteristics became clearer. Upon closer examination, the characteristics observed in connection weights with a parameter of 1.0 (b), appeared to gradually unfold, as the parameter decreased from 0.9 (c) to 0.6 (f). This means that the change in the parameter did not have any influences on the main characteristics of connection weights, but it only accelerated the learning in terms of clarifying the characteristics of connection weights.

Figure 9 shows connection weights at the final stage of learning steps. The results using the conventional method in

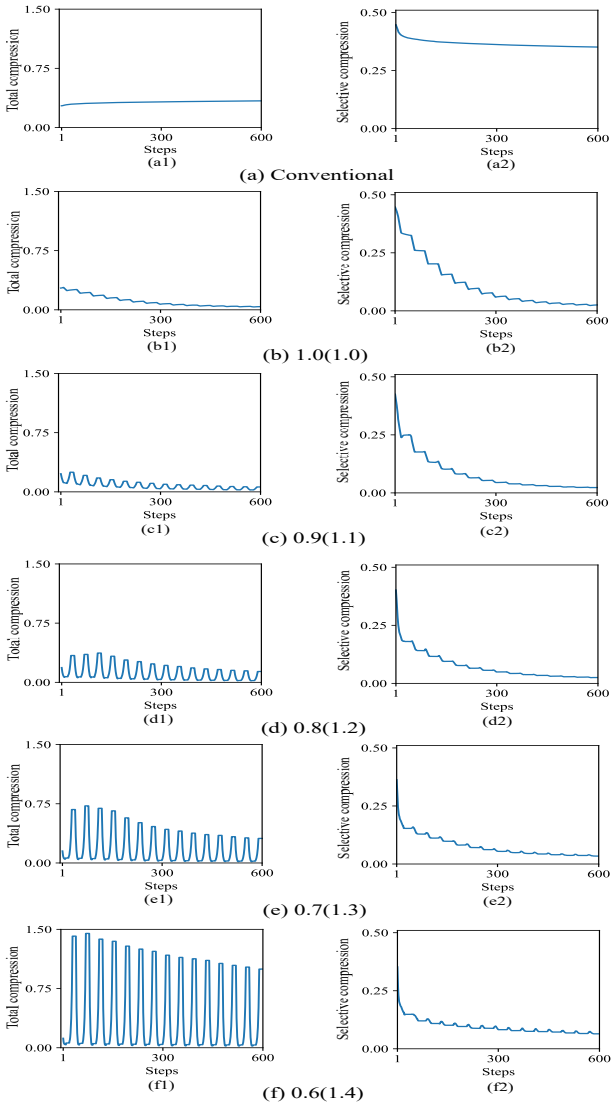

Figure 7. Average total (1) and selective (2) compression coefficients by the conventional method (a), and by changing the parameter θ from 1.0 (b) to 0.6 (f) for the customer data set.

Figure 9(a) still produced random connection weights. When the parameter decreased from 1.0 (b) to 0.6 (f), the number of strong weights diminished considerably compared with the results with 50 learning steps in Figure 8. Additionally, even when the parameter decreased to 0.6, the number of relatively stronger weights did not decrease significantly. This implied that even with an extremely decreased parameter, similar connection weights could still be obtained, which can be explained by the effect of total decompression. Furthermore, the connection weights between the first and the second hidden layer, and those between the ninth and tenth hidden layer were different from each other. This means that the connection weights were quite similar to each other in the initial stage of learning, and slight changes occurred in the later stages of learning, in particular, for connection weights between hidden layers, closer to the input and output layer.

 ISBN: 978-1-68558-046-9

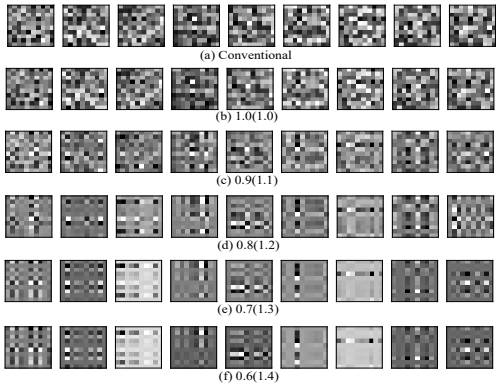

Figure 8. Connection weights, obtained after 50 learning steps by the conventional method (a) and by changing the parameter θ from 1.0 (b) to 0.6 (f) for the customer data set.

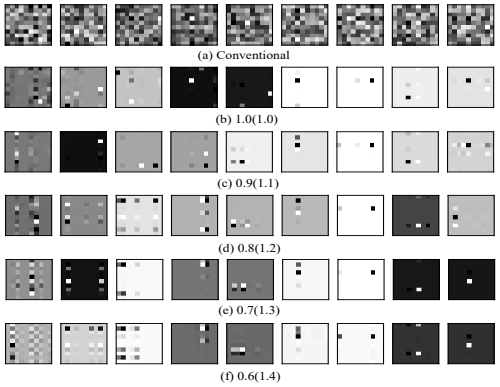

Figure 9. Connection weights, obtained at the final step of learning by the conventional method (a) and by changing the parameter θ from 1.0 (b) to 0.6 (f) for the customer data set.

The results show that the connection weights could produce similar and stable characteristics even if the parameter was forced to be decreased considerably, which was the effect of the total decompression. Minor differences could be obtained by the connection weights close to the input and output layer. These stable connection weights are certainly related to the stability of final performance, particularly in terms of generalization, as shown in Table I.

3) Layer-wise Selective Compression: We attempted to determine which hidden layers were primarily used in learning by computing the average layer compression coefficients for each hidden layer. By examing the average total compression coefficients, we observed that the new method applied less intense compression to connection weights closer to the input and output layer. This indicates that connection weights near to the input and output layer were not be easily compressible due to the abundance of information from inputs and outputs. However, when the parameter was excessively increased, only connection weights closest to the input tended to play an important role.

To clarify the characteristics of hidden layers, we plotted the normalized layer compression rates in Figure 10. We observed that the first hidden layer and the last hidden layer had larger compression coefficients. However, when the parameter decreased to 0.6 (f), only the first hidden layer tended to have larger compression coefficients. Finally, when using the conventional methods, the compression coefficients were larger for all hidden layers.

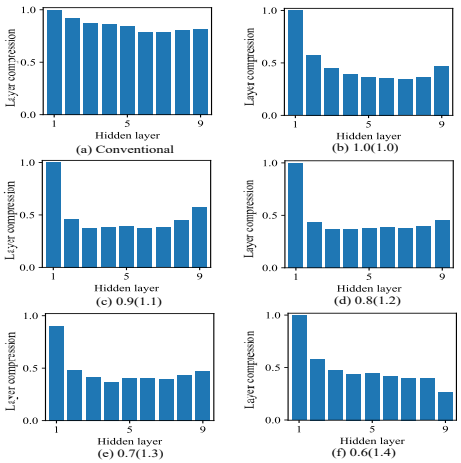

Figure 10. Normalized layer compression coefficients by the conventional method (a) by changing the parameter θ from 1.0 (b) to 0.6 (f) for the customer data set.

D. Summary of Numerical Results

Let us summarize the experimental results in terms of correlation coefficients and generalization performance. Better generalization was achieved through compression method, and particularly the new method was able to increase generalization while improving correlation coefficients.

Table I presents the summary of generalization and correlation coefficients. When the parameter was set to 1.0, the generalization accuracy was 0.684, but the correlation coefficient was the second lowest (0.807). As the parameter decreased from 0.9 to 0.6, all generalization accuracies exceeded 0.690. Additionally, the correlation coefficients were higher than those obtained all the other methods, except for the parameter $\theta = 0.8$. The similarity in generalization and correlation coefficients can be attributed to the stability of learning, as explained in the above experimental results on the stabilizing compression. The logistic regression analysis yielded accuracy of only 0.678, the second lowest one, and even the correlation coefficients was 0.855, which was not particularly large compared with those using the present method. Finally, the random forest model produced the lowest accuracy and correlation coefficient, as it could not handle negative effects.

One of the most important findings is that when the generalization accuracy was the largest ($\theta = 0.9$), the correlation coefficient was also the highest. This means that the present method tried to increase generalization by using connections weights as independently as possible, and by making the weights as linearly as possible.

 ISBN: 978-1-68558-046-9

TABLE I
SUMMARY OF EXPERIMENTAL RESULTS ON AVERAGED CORRELATION COEFFICIENTS AND GENERALIZATION PERFORMANCE BY OUR METHODS, COMPARING WITH THOSE BY THE CONVENTIONAL METHODS FOR THE CUSTOMER DATA SET. BOLD TYPE LETTERS INDICATE THE MAXIMUM VALUES.

Methods	Param$\theta(2-\theta)$	Accuracy	Correlation
Compression	1.0(1.0)	0.684	0.807
	0.9(1.1)	**0.692**	**0.885**
	0.8(1.2)	0.691	0.833
	0.7(1.3)	0.690	0.878
	0.6(1.4)	0.690	**0.885**
Conventional		0.683	0.833
Logistic		0.678	0.855
Random		0.613	0.256

IV. CONCLUSION

The present paper proposed a new interpretation method in which a process of interpretation was replaced for finding an interpretable prototype network. The prototype network is supposed to be found by simplifying multi-layered neural networks to the extreme point. The compression method is called "comprehensive compression", which is composed of interpretable and stabilizing compression. In the interpretable compression, multi-layered neural networks are compressed into the simplest ones without hidden layers, and all the internal representations, obtained by different initial conditions, inputs, parameters, learning steps, are averaged to produce the final collective weights. Those collective weights are compared with the weights in the prototype network to see relations between inputs and targets more exactly. The method was applied to the analysis of a customer data set, trying to clarify the characteristics of customers. By considering the correlation coefficients between inputs and targets as weights in the supposed prototype network, our method could detect linear relations, as well as non-linear ones to capture clearly the characteristics of customers.

One of the major problems is how to check the validity of our estimated prototype networks. In this paper, we tried to consider the correlation coefficient as an example of coefficients of prototype networks. However, we need to examine more exactly the possibility of the prototype network in addition to the linear correlation coefficients between inputs and outputs. Though some problems should be solved for more practical applications, the present method can certainly contribute to the development of global interpretation methods in neural networks.

REFERENCES

[1] X. Li, H. Xiong, X. Li, X. Wu, X. Zhang, J. Liu, J. Bian, and D. Dou, "Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond," *Knowledge and Information Systems*, pp. 1–38, 2022.

[2] G. Visani, E. Bagli, F. Chesani, A. Poluzzi, and D. Capuzzo, "Statistical stability indices for lime: obtaining reliable explanations for machine learning models," *arXiv preprint arXiv:2001.11757*, 2020.

[3] B. Goodman and S. Flaxman, "European union regulations on algorithmic decision-making and a right to explanation," *arXiv preprint arXiv:1606.08813*, 2016.

[4] G. L. Sanclemente and B. N. Cardozo, "Reliability: understanding cognitive human bias in artificial intelligence for national security and intelligence analysis," *Security Journal*, pp. 1–21, 2021.

[5] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital signal processing*, vol. 73, pp. 1–15, 2018.

[6] B. Zhou, D. Bau, A. Oliva, and A. Torralba, "Interpreting deep visual representations via network dissection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 9, pp. 2131–2145, 2018.

[7] R. Fong and A. Vedaldi, "Explanations for attributing deep neural network predictions," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 149–167, Springer, 2019.

[8] Y. Liang, S. Li, C. Yan, M. Li, and C. Jiang, "Explaining the black-box model: A survey of local interpretation methods for deep neural networks," *Neurocomputing*, vol. 419, pp. 168–182, 2021.

[9] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[10] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[11] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 3–14, 2017.

[12] Y. Liu, Z. Wang, H. Jin, and I. Wassell, "Multi-task adversarial network for disentangled feature learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3743–3751, 2018.

[13] L. Wang, Y. Yan, K. He, Y. Wu, and W. Xu, "Dynamically disentangling social bias from task-oriented representations with adversarial attack," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3740–3750, 2021.

[14] C. Yang, A. Rangarajan, and S. Ranka, "Global model interpretation via recursive partitioning," in *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 1563–1570, IEEE, 2018.

[15] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable ai for trees," *Nature machine intelligence*, vol. 2, no. 1, pp. 2522–5839, 2020.

[16] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[17] T. Wang, "Gaining free or low-cost interpretability with interpretable partial substitute," in *International Conference on Machine Learning*, pp. 6505–6514, PMLR, 2019.

[18] M. Wu, S. Parbhoo, M. Hughes, R. Kindle, L. Celi, M. Zazzi, V. Roth, and F. Doshi-Velez, "Regional tree regularization for interpretability in deep neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 6413–6421, 2020.

[19] O. Li, H. Liu, C. Chen, and C. Rudin, "Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[20] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[21] J. O. Neill, "An overview of neural network compression," *arXiv preprint arXiv:2006.03669*, 2020.

[22] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," 2020.

[23] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[24] Y. Guo, A. Yao, and Y. Chen, "Dynamic network surgery for efficient dnns," *Advances in neural information processing systems*, vol. 29, 2016.

[25] D. Blalock, J. J. Gonzalez Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?," *Proceedings of machine learning and systems*, vol. 2, pp. 129–146, 2020.

[26] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, "Pruning and quantization for deep neural network acceleration: A survey," *Neurocomputing*, vol. 461, pp. 370–403, 2021.

[27] T. Shimoyama, Y. Matsuda, and T. Miki, *Python practical data analysis (in Japanese)*. Tokyo: Shuwa System, 2022.

 ISBN: 978-1-68558-046-9