International Journal of Advances in Intelligent Informatics
Vol. 5, No. 3, November 2019, pp. 179-192
ISSN 2442-6571

Internal and collective interpretation for improving human interpretability of multi-layered neural networks

Ryotaro Kamimura [a,b,1,*]

[a] Kumamoto Drone Technology and Development Foundation, Techno Research Park, Techno Lab 203, 1155-12, Japan
[b] IT Education Center, Tokai Univerisity, 4-1-1 Kitakaname, Hiratsuka, Kanagawa 259-1292, Japan
[1] ryotarokami@gmail.com
* corresponding author

ARTICLE INFO

Article history
Received July 1, 2019
Revised October 21, 2019
Accepted October 29, 2019
Available online October 29, 2019

Keywords
Mutual information
Internal interpretation
Collective interpretation
Inference mechanism
Generalization

ABSTRACT

The present paper aims to propose a new type of information-theoretic method to interpret the inference mechanism of neural networks. We interpret the internal inference mechanism for itself without any external methods such as symbolic or fuzzy rules. In addition, we make interpretation processes as stable as possible. This means that we interpret the inference mechanism, considering all internal representations, created by those different conditions and patterns. To make the internal interpretation possible, we try to compress multi-layered neural networks into the simplest ones without hidden layers. Then, the natural information loss in the process of compression is complemented by the introduction of a mutual information augmentation component. The method was applied to two data sets, namely, the glass data set and the pregnancy data set. In both data sets, information augmentation and compression methods could improve generalization performance. In addition, compressed or collective weights from the multi-layered networks tended to produce weights, ironically, similar to the linear correlation coefficients between inputs and targets, while the conventional methods such as the logistic regression analysis failed to do so.

1. Introduction

Machine learning has been used in many areas of our daily life, causing some troubles in our life. As the techniques inside become larger and more complex, it becomes harder to interpret the main inference mechanism and to explain why and how the decisions made by the machine learning techniques reach their final conclusion. Because the methods have had serious influences over our safety [1], and the users of the techniques should have the right to receive an explanation of how the decisions are made, there has been an urgent need to develop methods to interpret and explain the main inference mechanism of the machine learning techniques [2].

Thus, many types of methods for interpretation have been developed in machine learning, which can be classified into two types: internal and external interpretation. In the internal interpretation, the methods aim to produce models whose components can be directly inspected and interpreted [3]–[5]. On the contrary, in the external interpretation, the models are considered as black-box ones, and try to interpret the inference mechanism externally [6]–[8]. In the neural networks, similarly as for the machine learning techniques, the interpretation methods have been classified as “decompositional” or “pedagogic” [9]. The pedagogic model is the black-box model and tries to infer the relations between inputs and outputs only by inspecting the inputs and outputs externally. The decompositional approach

tries to analyze the components such as connection weights and neuron activations directly. Thus, the method can be considered as the above-mentioned internal interpretation. However, usually, in the decompositional approach, many external methods, such as symbolic rules, fuzzy rules, decision trees, have been used to analyze and represent the components [10]–[12]. In addition, to extract the rules, many techniques, such as digitization of inputs and outputs for extracting rules, have been applied [9]. Thus, those methods cannot be called "internal interpretation" methods, but they have tried to interpret the final results by some external methods, and it is more appropriate call them "external interpretation." As is known, the objective of the interpretation is two-fold. First, and naturally, the interpretation method can be used to explain the inference mechanism in human intelligible ways. In addition, the clarification of the inference mechanism can be used to improve the general property, such as generalization performance, of neural networks. Considering two important aspects behind the interpretation, the interpretation methods so far developed have been dependent on methods not related to the real inference mechanism of neural networks. Thus, when we need to improve further the performance of neural networks, it is necessary to interpret internally the main inference mechanism.

In addition to the external interpretation, we have faced another problem, that of instable interpretation. Ordinarily, machine learning, as well as neural networks, are trained with many different data sets and initial conditions, in particular, in evaluating generalization performance. Thus, even for the same data set, we can have completely different internal representations due to different initial conditions. The problem is selecting which representation among many we should interpret. One of the practical solutions is to interpret a representation with the best generalization performance. This means that we try to see the ability of neural networks only from one aspect of improved generalization. We think that all representations created by different data sets and initial conditions should be taken into account for uncovering the fundamental properties of data sets. Then, for the problem of instability of interpretation, we should collectively interpret all internal representations created by learning, where each representation should have equal importance. It seems to us that the problem of collective interpretation has not been fully examined in machine learning as well as neural networks except in some exceptional cases with the ensemble methods [9], [13]. In this context, the present paper proposes a new type of interpretation called "collective interpretation," in which all representations from neural networks should be taken into account with equal importance.

We have shown that interpretable neural networks should be internally interpreted and all different types of internal representations should be collectively interpreted. Let us consider how to create neural networks with those properties for interpretation. As mentioned, in neural networks, there have been many types of interpretation methods, and the majority of those methods have been based on the simplification of network complexity. This is based upon the assumption that, as networks become simpler, their interpretation becomes simpler. Since the beginning of research on this matter, there have been many types of simplification methods proposed [14]–[17]. However, it cannot be said that those methods can be applied successfully to the interpretation problem because they have been mainly used to improve generalization performance. In addition, we can say that simplified networks are not necessarily easily interpreted, because it happens that too much information may be condensed in simpler components. In particular, in applying them to the interpretation problems, multiple solutions, by different initial conditions and with different choices of parameter for the regularization terms, have prevented us from interpreting the final results.

Recently, more powerful methods of simplification, namely, model compression methods, have been proposed. The model compression methods aim to compress complex and larger networks into smaller ones by transferring knowledge in larger networks to smaller ones [18]–[20]. However, the methods are considered principally as black-box ones, in which the behaviors of larger networks or teacher networks are imitated by the corresponding smaller student networks using the soft targets from the larger networks. In addition, the methods have been used only to improve generalization performance. So, even if we succeed in interpreting the main mechanism of student networks, we cannot say that the methods can interpret the original and teacher networks internally.

Though the model compression can be used to produce the simple models for practical applications as well as hardware implementations, the present paper assumes that the internal interpretation should be needed for improving the interpretation performance, and even for improving generalization performance. For using the model compression methods for the internal interpretation, we have developed an information compression method called "neural compressor" [21]. The neural compressor tries to compress all connection weights in all layers into the simplest ones. Because the compressed weights or collective weights should directly inherit the main characteristics of connection weights from the larger networks, the method eventually tries to interpret the inference mechanism of the original teacher networks internally. Besides, the collective interpretation is realized by averaging all compressed weights over all different types of data sets and initial conditions.

However, when we tried to apply the neural compressor to actual problems, we observed that the generalization performance of networks with compressed weights tended to degrade considerably. We encountered a problem of whether compressed weights really represented the characteristics of weights of the original networks due to poor generalization performance. By examining carefully information flow in compression processes, we found that compression processes were naturally accompanied by the loss of information content in the original connection weights. In the worst case, information considered important, could be lost in a process of compression. The problem of this lost information forced us to develop a method to increase information content on inputs as much as possible before compression. Thus, the present paper aims to examine the performance of multi-layered neural networks with information augmentation components introduced against the information loss. In addition, we try to evaluate the utility of compressed or collective weights from multi-layered neural networks to train the simplest networks with no hidden layers.

2. Method

2.1. Collective Interpretation and Information Compression

We try to show here how to compress multi-layered neural networks, as shown in Fig. 1. The collective or compressed weights are obtained by multiplying and summing all connection weights and by averaging those weights for all different initial conditions and different input patterns. Thus, the final collective weights tend to represent the overall characteristics of connection weights between inputs and outputs.

For simplicity's sake, we deal here with connection weights for different initial conditions. Now, let us compute step by step collective weights between inputs and outputs for different initial conditions. For illustration purposes, we use the five-layered neural networks shown in Fig. 1, where the number of layers is changed from one (1) to five (5), and the first layer is the input and the final layer is the output layer by definition. First, suppose that for the t^{th} initial condition (t = 1,2,...,r), connection weights are obtained after training. Then, two connection weights of the final two layers, namely, ${}^{t}w_{ij'}^{(5)}$ and ${}^{t}w_{j'j}^{(4)}$, are combined into

$$ {}^{t}w_{ij}^{(5*4)} = \sum_{j'=1}^{n_4} {}^{t}w_{ij'}^{(5)} \, {}^{t}w_{j'j}^{(4)} \tag{1} $$

Then, these collective weights are further combined as

$$ {}^{t}w_{ij}^{(5*3)} = \sum_{j'=1}^{n_3} {}^{t}w_{ij'}^{(5*4)} \, {}^{t}w_{j'j}^{(3)} \tag{2} $$

Finally, we have the final collective weights

$$ {}^{t}w_{ik}^{(5*2)} = \sum_{j=1}^{n_2} {}^{t}w_{ij}^{(5*3)} \, {}^{t}w_{jk}^{(2)} \tag{3} $$

With collective weights, we train two-layered neural networks as shown in Fig. 1(b) and (c). In Fig. 1(c), two-layered neural networks are initialized with collective weights ${}^{t}w_{ik}^{(5*2)}$. Then, these two-layered neural networks are trained to produce the targets, but this time, the learning is performed with information from multi-layered neural networks. Finally, for interpretation, the collective weights can be obtained by averaging all connection weights

$$\overline{w}_{ik}^{(5*2)} = \frac{1}{r} \sum_{t=1}^{r} {}^{t}w_{ij}^{(5*2)} \tag{4}$$

Thus, these compressed and collective weights can be used to interpret relations between inputs and outputs.

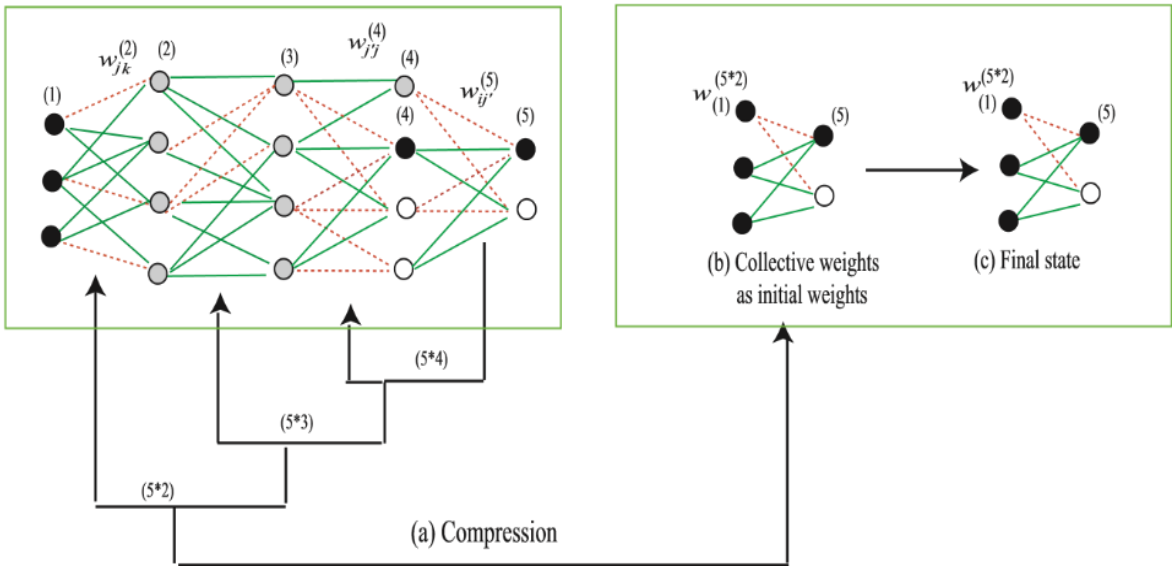


Fig. 1. Network architecture with a process to obtain collective weights from an initial multi-layered neural network (a). The final two-layered network (b) was trained with collective weights as initial weights to produce the final state (c).

2.2. Mutual Information Augmentation

In the information augmentation component in Fig. 2, we try to increase mutual information between input patterns and neurons in the second layer, denoted by (2). For computing mutual information, we must compute distance between an input x_k^s of the sth input pattern (s = 1,2,...,q) to the kth input neuron (k = 1,2,...,n_1) and the corresponding connection weight $w_{jk}^{(2)}$ from the k^{th} input neuron to the j^{th} hidden neuron of the second layer (j = 1,2,...,n_2)

$${}^{s}z_j^{(2)} = \sum_{k=1}^{n_1} \left({}^{s}x_k - w_{jk}^{(2)}\right)^2 \tag{5}$$

The output from the jth neuron of the second layer is computed by

$${}^{s}v_j^{(2)} = \exp\left(-\frac{{}^{s}z_j^{(2)}}{\sigma}\right) \tag{6}$$

where the parameter σ represents the width of the distribution. The firing probability is computed by normalizing the output

$$p^{(2)}(j|s) = \frac{{}^{s}v_j^{(2)}}{\sum_{j'=1}^{n_2} {}^{s}v_{j'}^{(2)}} \tag{7}$$

By this normalized output, mutual information is defined by

$$I^{(2)} = -\sum_{j=1}^{n_2} p^{(2)}(j) \log p^{(2)}(j) + \sum_{s=1}^{q} \sum_{j=1}^{n_2} p(s)\, p^{(2)}(j|s) \log p^{(2)}(j|s) \quad (8)$$

When this mutual information is maximized, all neurons fire equally on average, while each neuron responds to a specific group of neurons. This means that all neurons respond very specifically to input patterns. Thus, when mutual information increases, each neuron plays a specific role.

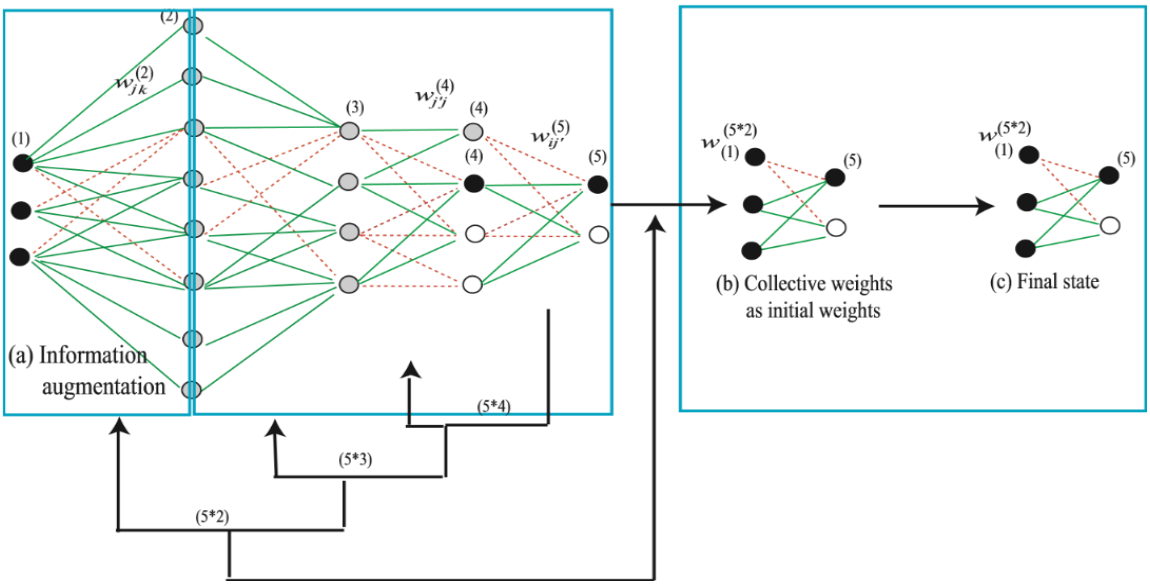


Fig. 2. Network architecture composed of information augmentation (a) with a process to obtain collective weights from an initial multi-layered neural network. The final two-layered network (b) was trained with collective weights as initial weights to produce the final state (c).

2.3. Computational Methods for Information Augmentation

Mutual information is defined between inputs and input patterns, having two properties, namely, the equal use of neurons and specific responses to inputs. This means that all neurons should be equally used and respond to specific input patterns. In terms of interpretation, mutual information maximization has the effect of disentangling complex features into simpler and distinct features. Thus, those disentangled features can be simple enough for easy interpretation, when the number of neurons increases. The importance of mutual information has been stressed since Linsker's pioneer work on maximum information preservation [22]–[24]. Since then, there have been many attempts to use mutual information in neural networks [25]–[27]. However, one of the main problems in mutual information is the computational complexity of computing mutual information, composed of entropy and conditional entropy of neurons with respect to input patterns.

The present paper uses competitive learning to realize mutual information in which the computation of mutual information is greatly simplified [28]. The competitive learning [29] tries to detect a winner, closest to specific input patterns, and all competitive neurons tend to respond to distinct input patterns in an ideal state. This property of competitive learning is close to the objective of mutual information maximization. As has been well known, competitive learning tries to determine a winner ${}^{s}c$ for the sth input pattern

$${}^{s}c = argmax_j\, p(j|s) \quad (9)$$

Thus, the winner tries to imitate the corresponding input patterns as much as possible. In addition, competitive learning supposes that all winning neurons represent distinct input patterns in an ideal state. This process of winning neurons tries to minimize the conditional entropy, the second term of mutual information. In addition, the equal use of all neurons corresponds to maximizing the entropy, the first term of mutual information. However, the equal use of all competitive neurons cannot be easily realized. In actual competitive processes, some neurons can be dead, meaning that they do not respond to any

input patterns. Though many methods to solve the dead neuron problem have been proposed [30], [31], the problem has not yet been solved. To weaken the problem of dead neurons, we use here one variant of competitive learning, namely, the self-organizing map (SOM) [32]. The SOM determines a winner, and it tries to imitate specific input patterns as much as possible. One of the main differences is that, in the SOM, not only a winner but also other neurons, located in the vicinity of the winner, try to imitate input patterns, depending on the distance from the winner. Thus, the SOM is more suited for a situation where the number of neurons tends to increase, as is the case with the method proposed in this paper.

In addition, mutual information can be realized by increasing the number of neurons. To see how mutual information can be increased by increasing the number of neurons, we consider maximum mutual information content. As has been well known, maximum mutual information is obtained by

$$I_{max}^{(2)} = \log n_2 \tag{10}$$

where n_2 represents the number of neurons in the second layer. As can be seen in the equation, mutual information can be increased by just increasing the number of neurons. Thus, all we have to do to increase the mutual information is to increase the dimensionality n_2 as much as possible.

3. Results and Discussion

3.1. Glass Data Set

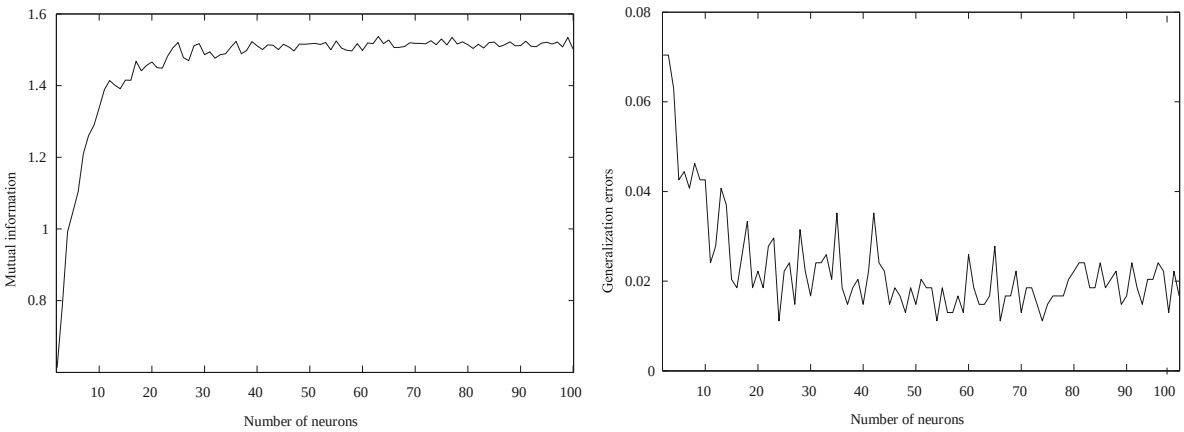
3.1.1. Experimental Outline

The first experiment used the well-known and easily accessible glass data set (glass dataset), where we tried to classify glass either as window or non-window, depending on the glass chemistry. The number of inputs was nine (refractive index, sodium, magnesium, aluminum, silicon, potassium, calcium, barium, and iron). The number of patterns was 214, which was increased so as to make the window and non-window glass as equal as possible for easy evaluation of the final results by using the well-known minority up-sampling method SMOTE [33]. As explained in the section above, the maximum number of layers was set to five, because we could not obtain better results when the number of layers was further increased. We used the Matlab neural network package with all default values except the number of neurons in the second layers. This is because the use of the default setting makes it possible to easily reproduce the present results. Naturally, if we try to tune the parameters, we can expect to obtain results better than those discussed in this paper. Only the number of neurons in the second layer was determined to produce the best possible generalization performance.

3.1.2 Information and Generalization

Fig. 3(a) shows mutual information when the number of neurons in the second layer increased from 2 to 100. Mutual information increased sharply at the beginning and then increased very slowly. Fig. 3(b) shows generalization errors as a function of the number of neurons in the second layer. We could see that generalization errors decreased rapidly at the beginning and then decreased slowly when the number of neurons increased. This result shows that, when mutual information increased, generalization errors correspondingly decreased gradually.

Table 1 shows a summary of generalization errors. As can be seen in the table, the best average error of 0.0074 was obtained by the present method with information augmentation (IA) with three and four layers. In addition, the present method showed the best standard deviation (0.0096), minimum error (0), and maximum error (0.0185). The error was much lower than that of 0.0358 obtained by the logistic regression analysis. The error was even much lower than that of 0.0204 obtained by the bagging ensemble method [34], [35], which is close to the collective operation in the present method. Also, the well-known AdaBoostM1 method [36]–[39] could not produce reasonably low errors.



(a) Mutual information (b) Generalization errors

Fig. 3. Mutual information (a) and generalization errors (b) as a function of the number of neurons for five-layered neural networks for the glass data set.

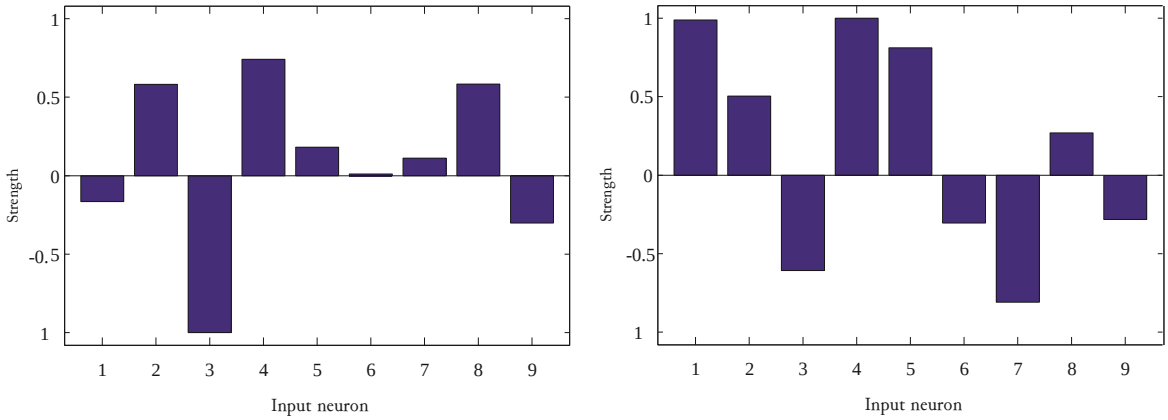
Table 1. Summary of experimental results on generalization performance for the glass data set. The numbers of layers denoted by 3-2, 4-2, and 5-2 means that two-layered networks were trained with collective weights from three-, four-, and five-layered networks.

Layers	Methods	Neurons	Avg	Std dev	Min	Max
	Logistic		0.0358	0.0139	0.0189	0.0566
	AdaBoostM1		0.4074	0.0611	0.3148	0.5000
	Bagging		0.0204	0.0222	0.0000	0.0556
2	Simple		0.0370	0.0195	0.0000	0.0741
3	Simple		0.0167	0.0184	0.0000	0.0556
3	IA	49	**0.0074**	**0.0096**	**0.0000**	**0.0185**
4	Simple		0.0278	0.0251	0.0000	0.0741
4	IA	37	**0.0074**	**0.0096**	**0.0000**	**0.0185**
5	Simple		0.0204	0.0204	0.0000	0.0556
5	IA	24	0.0111	0.0156	0.0000	0.0370
3-2	Simple		0.0407	0.0259	**0.0000**	0.0926
3-2	IAC	49	0.0352	0.0254	**0.0000**	0.0926
4-2	Simple		0.0444	0.0265	0.0185	0.1111
4-2	IAC	37	0.0352	0.0295	**0.0000**	0.1111
5-2	Simple		0.0519	0.0273	0.0185	0.1111
5-2	IAC	24	0.0333	0.0244	**0.0000**	0.0926

Then, we compressed connection weights to get two-layered neural networks. When compressing from the simple multi-layered neural networks without an information augmentation component, generalization errors increased from 0.0407 (three layers, denoted by 3-2 simple in Table 1) to 0.0519 (five layers, 5-2 simple). These errors were larger than those of 0.0370 by the simple two-layered neural networks without collective weights and 0.0358 by the logistic regression analysis. With collective weights from multi-layered networks with information augmentation components, the error decreased from 0.0352 (three layers, 3-2 IAC) to 0.0333(five layers, 5-2 IAC). We could see that those errors were lower than that of 0.0358 by the logistic regression analysis, though slightly. Thus, the information augmentation and compression method could improve generalization, and also, the collective weights from multi-layered neural networks with information augmentation could be used to increase generalization performance of two-layered neural networks without hidden layers.

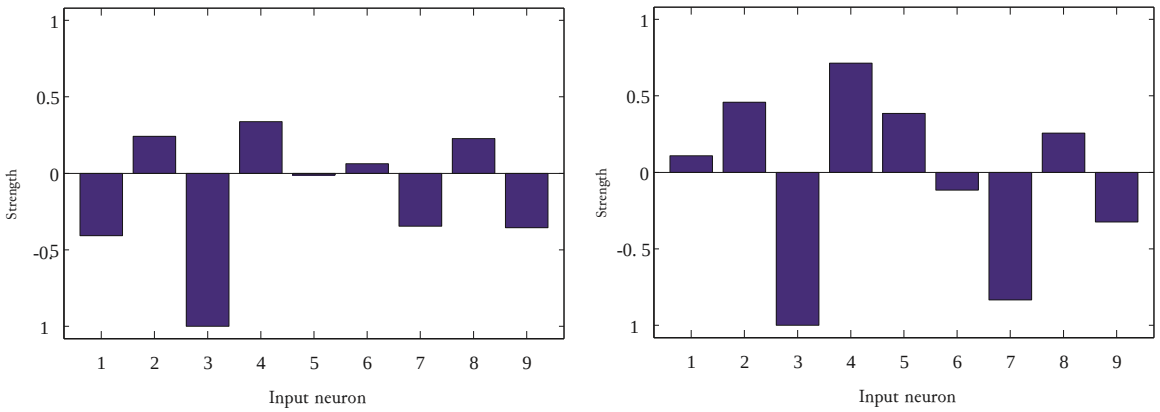
3.1.3 Collective Interpretation

Fig. 4(a) shows correlation coefficients between inputs and targets, where input No.3 had the largest absolute value. For easy interpretation and comparison, the maximum absolute value was adjusted to one. Fig. 4(b) shows connection weights from the simple two-layered neural networks without using collective weights. As can be seen in Fig. 4, connection weights were considerably different from the correlation coefficients. For example, in the connection weights, inputs No.1 and No.4 had larger absolute values, while input No.3, which had the largest absolute values in the correlation coefficients, was not so strong. Fig. 5(a) shows collective weights from five-layered neural networks with the best generalization performance. As can be seen in Fig. 5(a), the collective weights were considerably close to the correlation coefficients in Fig. 4(a), though the positive collective weights were weaker than the correlation coefficients. Fig. 5(b) shows connection weights by the two-layered neural networks initialized with collective weights. Collective weights and connection weights were similar to each other because the two-layered neural networks were initialized with the collective weights. However, in connection weights, the positive weights were stronger, and in addition, input No.7 had a stronger absolute value, which was different from the correlation coefficients.



(a) Correlation coefficients between inputs and targets (b) Connection weights by simple 2 layered networks

Fig. 4. Correlation coefficients between inputs and targets (a) and connection weights for two-layered networks with no hidden layers (b) for the glass data set.



(a) Collective weight from 5 layered neural networks (b) Connection weights by 2 layered networks with collective weights

Fig. 5. Collective weights from the five-layered neural networks (a) and connection weights for two-layered network initialized with the collective weights from the five-layered neural networks (b) for the glass data set.

Fig. 6(a) shows the prediction importance by the bagging ensemble method. The importance seems to be different from the correlation coefficients. However, considering that the importance could not distinguish between positive and negative effects, the importance considered input No.3 the most

important one, which was confirmed by the correlation coefficients. However, Fig. 6(b) shows the regression coefficients by the logistic regression analysis. The logistic regression could not detect simple linear correlations between inputs and targets. This is because the logistic regression analysis tended to produce different coefficients by slightly different training patterns. Since the results presented here were the average of all ten runs, the final results became different from the correlation coefficients.

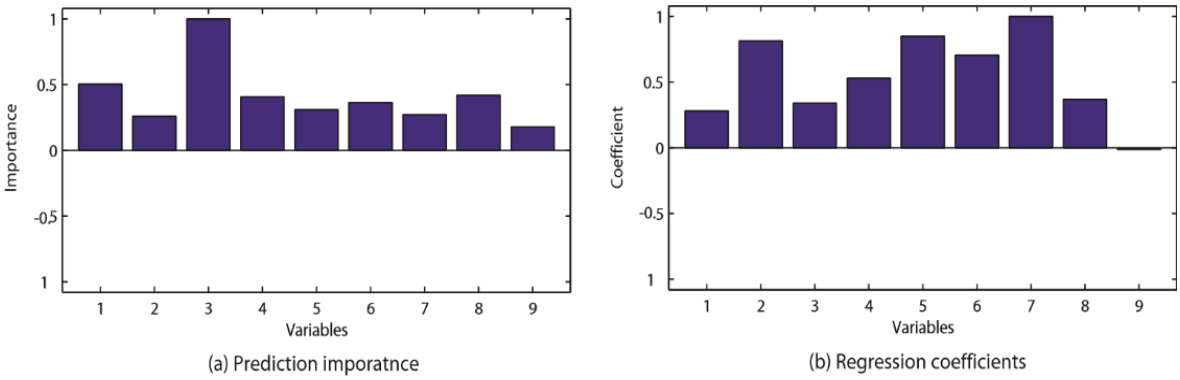


Fig. 6. Prediction importance by the bagging method (a) and regression coefficients by the logistic regression analysis (b) for the glass data set.

The results show that the information augmentation and compression tended to produce collective weights and connection weights close to the correlation coefficients. On the other hand, the simple multi-layered and two-layered neural networks and logistic regression produced connection weights, collective weights, and regression coefficients different from the correlation coefficients. These results show that the simple neural networks try to detect complex patterns as themselves, while the present method with information augmentation seems to disentangle complex features into simpler ones as much as possible, leading to the detection of linear relations. The logistic regression failed to detect even the linear relations, presumably because of the existence of multiple collinearities.

3.2. Pregnancy Data Set

3.2.1. Experimental Outline

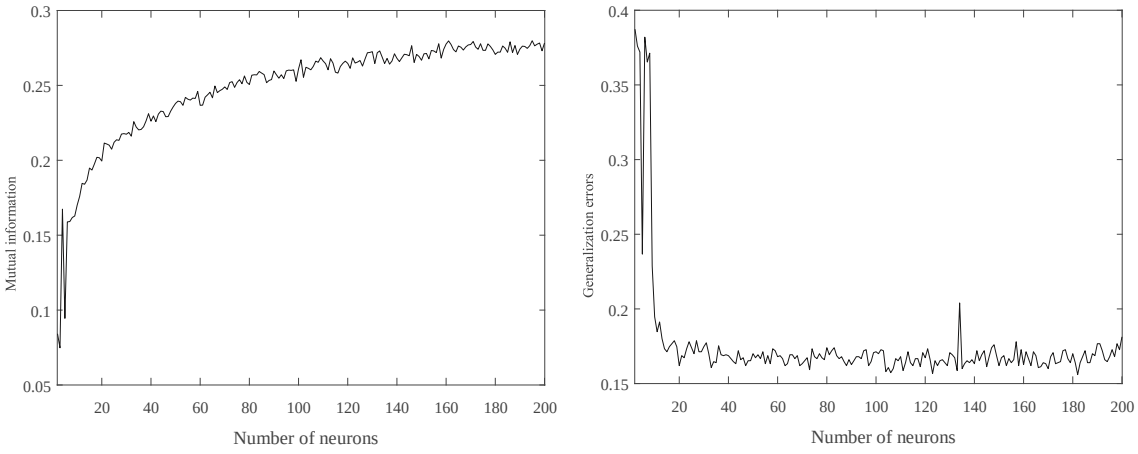
The second data set was used to predict the pregnancy of customers for the purpose of recommending pregnancy-related products to them [40]. The number of customers was 1,000, and the number of input variables was 25, representing gender (No.1) to maternity clothes (No.19). The number of neurons in the second layer increased from two to 100, and the number of neurons in the third and fourth layer were set to 10 (default values in the Matlab). Seventy percent of the data set was used for training, and the remainder was evenly divided into validation and testing data sets. For the sake of page limitation, we only present formal results on improved generalization and interpretation here.

3.2.2. Information and Generalization

Fig. 7 shows mutual information (a) and generalization errors (b) as a function of the number of neurons in the second layer of the five-layered neural networks for the pregnancy data set. As can be seen in Fig. 7(a), mutual information constantly increased when the number of neurons increased. On the other hand, the generalization errors decreased considerably in the first place, and then they slowly decreased. In the end, with 182 neurons, the smallest generalization error was obtained. This means that the number of neurons became much larger than the actual number of input variables for obtaining the best generalization performance. Then, fixing the number of neurons at 182 with the best generalization performance, we tried to control the spread parameter to increase mutual information for the two-layered neural networks with collective weights.

Table 2 shows a summary of generalization errors. The four-layered neural networks with information augmentation produced the best average error of 0.1553. With the same four-layered neural networks but without information augmentation, the average error increased to 0.1867, which was the maximum

error obtained by the four-layered networks with information augmentation. For all layers from three to five, generalization errors decreased when we used the information augmentation.



(a) Mutual information (b) Generalization errors

Fig. 7. Mutual information (a) and generalization errors (b) as a function of the number of neurons of the second layer for the pregnancy data set.

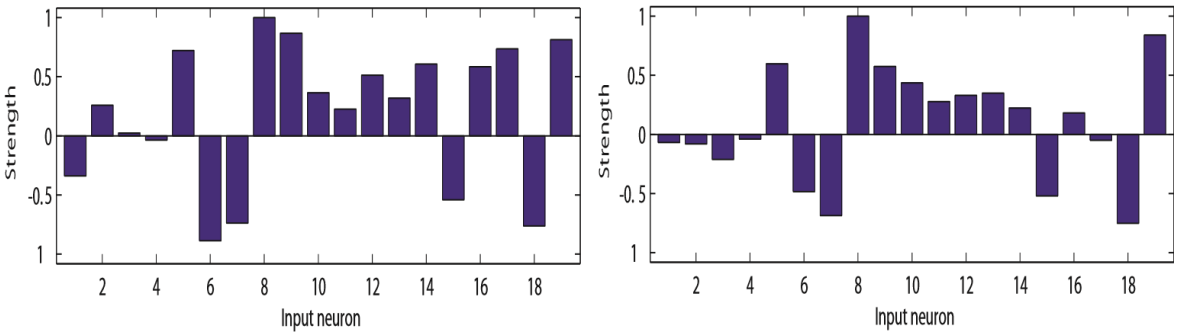
Table 2. Summary of experimental results on generalization performance for the pregnancy data set.

Layers	Methods	Neurons	Avg	Std dev	Min	Max
	Logistic		0.1627	0.0186	**0.1267**	0.1867
	AdaBoostM1		0.1773	0.0189	0.1533	0.2067
	Bagging		0.1847	0.0160	0.1533	0.2067
2	Simple		0.1880	0.0361	0.1333	0.2400
3	Simple		0.1767	**0.0145**	0.1600	0.2067
3	IA	55	0.1600	0.0208	**0.1267**	0.1867
4	Simple		0.1867	0.0340	**0.1267**	0.2467
4	IA	111	**0.1553**	0.0183	0.1333	0.1867
5	Simple		0.1780	0.0199	0.1467	0.2067
5	IA	182	0.1560	0.0225	**0.1267**	0.1867
3-2	Simple		0.1653	0.0239	**0.1267**	0.2133
3-2	IAC	55	0.1573	0.0167	0.1333	**0.1800**
4-2	Simple		0.1667	0.0249	**0.1267**	0.2133
4-2	IAC	111	0.1600	0.0204	**0.1267**	0.2000
5-2	Simple		0.1640	0.0178	0.1333	0.1867
5-2	IAC	182	0.1580	0.0166	0.1400	0.1933

Then, we could examine how the errors of two-layered networks changed with collective weights. When two-layered networks were trained with collective weights from three-layered networks with information augmentation (3-2, IAC), the error decreased from 0.1653 (simple) to 0.1573. By the collective weights from four-layered networks (4-2, IAC), the error decreased from 0.1667 to 0.1600. Finally, two-layered networks with collective weights from five-layered network (5-2, IAC) produced the error of 0.1580, decreased from 0.1640 by the simple networks. All generalization errors of two-layered networks by information augmentation and compression were lower than that of 0.1627 by the logistic regression analysis. Contrary to our expectation of the well-known AdaBoostM1 and bagging ensemble methods, which are close to our methods, they produced higher errors of 0.1773 and 0.1847. These results show that the present method with information augmentation and compression could produce collective weights by which two-layered neural networks showed better generalization performance than the conventional methods.

3.2.3. Collective Interpretation

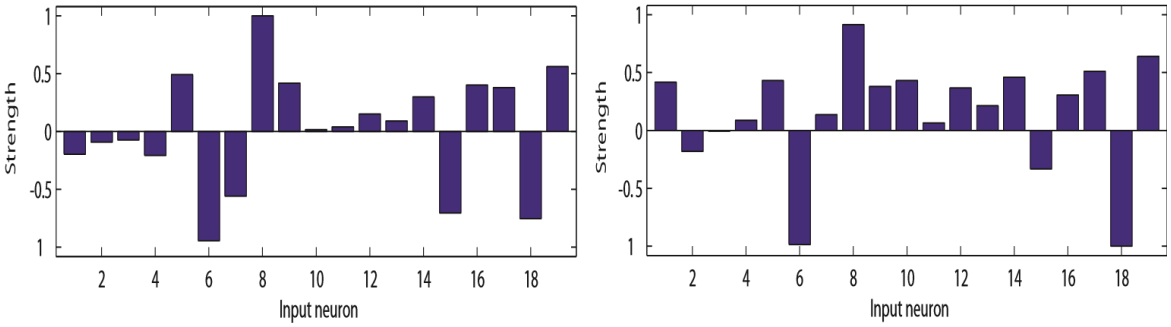
Fig. 8(a) shows correlation coefficients between inputs and targets for the pregnancy data set. Fig. 8(b) shows connection weights by the simple two-layered neural networks. Connection weights and correlation coefficients were similar to each other, though some different minor points in the middle could be seen. For example, inputs No. 16 and 17 were weaker, while the correlation coefficients between those inputs and the corresponding targets were relatively larger.



(a) Correlation coefficients between inputs and targets (b) Connection weights of simple 2 layered networks

Fig. 8. Correlation coefficients between inputs and targets (a) and weights for the simple two-layered neural networks (b) for the pregnancy data set.

On the other hand, for the collective weights, the stronger weights were quite similar to the corresponding correlations, but the minor correlations were weaker. Fig. 9(a) shows collective weights from four-layered neural networks with information augmentation components. The collective weights were similar to the correlation coefficients, but some minor weights, for example, weights from input No. 10 to 13, were weaker in the collective weights. Fig. 9(b) shows connection weights by the two-layered neural networks initialized with the collective weights. The connection weights inherited the major strong collective weights in Fig. 9(a), but inputs No. 6 and 18 became much stronger.



(a) Collective weights from 5 layered networks (b) Connection weights of 2 layered networks with collective weights

Fig. 9. Collective weights by the three-layered neural networks (a) and weights by the two-layered with the collective weight of multi-layered neural networks (b) for the pregnancy data set.

The results show that the present method with information augmentation tried to produce collective weights similar to the correlation coefficients, but it tried to select the important inputs from among inputs with higher correlation coefficients. Finally, we should note that the logistic regression analysis produced regression coefficients completely different from the correlation coefficients. This means that not the logistic regression analysis but the present method with information augmentation tried to detect linear correlations between input and targets.

4. Conclusion

The present paper aimed to propose a new interpretation method, which tries to interpret the inference mechanism internally and collectively. The internal interpretation aims to interpret the inference mechanism itself without any external methods. In addition, collective interpretation tries to

interpret all internal representations produced by different initial conditions and different input patterns, because all created representations are considered equal in importance. The method tries to compress multi-layered neural networks into the simplest networks without hidden layers, which can be used to interpret the relations between inputs and outputs, as is the case with the conventional regression analysis. However, the compression is accompanied by information loss on input patterns in a process of compressing multi-layered neural networks. Thus, we proposed a method to augment information content on input patterns before compressing multi-layered neural networks. Actually, the mutual information between input patterns and neurons in the information augmentation component is forced to be increased as much as possible. This increased mutual information has the effect of disentangling complex information in input patterns into a set of simple features. However, we faced difficulty in computing mutual information directly, and we used competitive learning, more exactly, SOM, for increasing mutual information. In addition, when the number of neurons increases, naturally mutual information can be increased. Thus, we tried to increase the number of neurons as much as possible in the information augmentation component.

The method was applied to two data sets namely, the well-known glass data set and the more practical pregnancy data set. In both data sets, the present method could improve generalization performance, and the performance was considerably better than that of other conventional methods, in particular, ensemble methods close to our method, though the main focus of our method was on the interpretation. One of the main findings was that information compression with information augmentation tried to extract relations between inputs and outputs, described in the correlation coefficients between input and targets. On the other hand, by the conventional methods, final weights seem to be different from their correlation coefficients. Ironically, our method lies in the extraction of linear relations between inputs and outputs, while the conventional methods such as regression analysis could not detect those relations, probably due to the multiple collinearities.

One of the main problems is related to computational methods for increasing mutual information. The present paper used competitive learning, or more exactly, SOM, for increasing mutual information. This is because competitive learning tries to use all neurons equally on average, responding to specific input patterns, which conforms to the objective of mutual information maximization in this paper. However, competitive learning has had the well-known shortcoming of dead neurons, where some neurons cannot be used in learning. In mutual information maximization, all neurons should be equally used, which is not exactly achieved by the present method of competitive learning. We think that more powerful methods will be needed to make all neurons activated equally for the further development of the method.

Finally, for future direction, we should examine to what extent the method can be used to clarify the actual meaning of data sets. For example, in the second pregnancy data set, we only examined formal relations between the connection weights and correlation coefficients on inputs and targets. We should examine whether the obtained connection weights can explain the actual meaning of the data set. In addition, we should examine in what points the present method is different from the linear correlations between inputs and targets, and explain why and how the difference can be produced for the further development of neural networks.

References

[1] K. R. Varshney and H. Alemzadeh, "On the Safety of Machine Learning: Cyber-Physical Systems, Decision Sciences, and Data Products," *Big Data*, vol. 5, no. 3, pp. 246–255, Sep. 2017, doi: 10.1089/big.2016.0051.

[2] B. Goodman and S. Flaxman, "European Union Regulations on Algorithmic Decision-Making and a 'Right to Explanation,'" *AI Mag.*, vol. 38, no. 3, pp. 50–57, Oct. 2017, doi: 10.1609/aimag.v38i3.2741.

[3] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan, "Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model," *Ann. Appl. Stat.*, vol. 9, no. 3, pp. 1350–1371, Sep. 2015, doi: 10.1214/15-AOAS848.

[4] F. Wang and C. Rudin, "Falling rule lists," in *Artificial Intelligence and Statistics*, 2015, pp. 1013–1022, available at : http://proceedings.mlr.press/v38/wang15a.pdf.

[5] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible Models for HealthCare," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15*, 2015, pp. 1721–1730, doi: 10.1145/2783258.2788613.

[6] M. Craven and J. W. Shavlik, "Extracting tree-structured representations of trained networks," in *Advances in neural information processing systems*, 1996, pp. 24–30, available at : Google Scholar.

[7] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. MÃžller, "How to explain individual classification decisions," *J. Mach. Learn. Res.*, vol. 11, no. Jun, pp. 1803–1831, 2010, available at : Google Scholar.

[8] I. Kononenko and others, "An efficient explanation of individual classifications using game theory," *J. Mach. Learn. Res.*, vol. 11, no. Jan, pp. 1–18, 2010, available at : Google Scholar.

[9] G. Bologna, "Is it worth generating rules from neural network ensembles?," *J. Appl. Log.*, vol. 2, no. 3, pp. 325–348, Sep. 2004, doi: 10.1016/j.jal.2004.03.004.

[10] G. G. Towell and J. W. Shavlik, "Extracting refined rules from knowledge-based neural networks," *Mach. Learn.*, vol. 13, no. 1, pp. 71–101, Oct. 1993, doi: 10.1007/BF00993103.

[11] R. Andrews, J. Diederich, and A. B. Tickle, "Survey and critique of techniques for extracting rules from trained artificial neural networks," *Knowledge-Based Syst.*, vol. 8, no. 6, pp. 373–389, Dec. 1995, doi: 10.1016/0950-7051(96)81920-4.

[12] J. L. Castro, C. J. Mantas, and J. M. Benitez, "Interpretation of artificial neural networks by means of fuzzy rules," *IEEE Trans. Neural Networks*, vol. 13, no. 1, pp. 101–116, 2002, doi: 10.1109/72.977279.

[13] R. Wall and P. Cunningham, "Exploring the potential for rule extraction from ensembles of neural networks," in *11th Irish Conference on Artificial Intelligence & Cognitive Science*, 2000, pp. 52–68, available at : Google Scholar.

[14] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in neural information processing systems*, 1990, pp. 598–605, available at : http://papers.nips.cc/paper/250-optimal-brain-damage.pdf.

[15] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Advances in neural information processing systems*, 1992, pp. 950–957, available at : Google Scholar.

[16] S. Srinivas and R. V. Babu, "Data-free Parameter Pruning for Deep Neural Networks," in *Procedings of the British Machine Vision Conference 2015*, 2015, p. 31.1-31.12, doi: 10.5244/C.29.31.

[17] G. G. Oliveira, O. C. Pedrollo, and N. M. R. Castro, "Simplifying artificial neural network models of river basin behaviour by an automated procedure for input variable selection," *Eng. Appl. Artif. Intell.*, vol. 40, pp. 47–61, Apr. 2015, doi: 10.1016/j.engappai.2015.01.001.

[18] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*, 2006, p. 535, doi: 10.1145/1150402.1150464.

[19] J. Ba and R. Caruana, "Do Deep Nets Really Need to be Deep?," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2654–2662, available at: http://papers.nips.cc/paper/5484-do-deep-nets-really-need-to-be-deep.pdf.

[20] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv Prepr. arXiv1503.02531*, 2015, available at: https://arxiv.org/abs/1503.02531.

[21] R. Kamimura, "Neural self-compressor: Collective interpretation by compressing multi-layered neural networks into non-layered networks," *Neurocomputing*, vol. 323, pp. 12–36, Jan. 2019, doi: 10.1016/j.neucom.2018.09.036.

[22] R. Linsker, "Self-organization in a perceptual network," *Computer (Long. Beach. Calif).*, vol. 21, no. 3, pp. 105–117, Mar. 1988, doi: 10.1109/2.36.

[23] R. Linsker, "Local Synaptic Learning Rules Suffice to Maximize Mutual Information in a Linear Network," *Neural Comput.*, vol. 4, no. 5, pp. 691–702, Sep. 1992, doi: 10.1162/neco.1992.4.5.691.

[24] R. Linsker, "Improved local learning rule for information maximization and related applications," *Neural Networks*, vol. 18, no. 3, pp. 261–265, Apr. 2005, doi: 10.1016/j.neunet.2005.01.002.

[25] S. Becker, "Mutual information maximization: models of cortical self-organization," *Netw. Comput. Neural Syst.*, vol. 7, no. 1, pp. 7–31, Jan. 1996, doi: 10.1080/0954898X.1996.11978653.

[26] G. Deco and D. Obradovic, *An information-theoretic approach to neural computing*. Springer Science & Business Media, 2012, available at: Google Scholar .

[27] J. C. Principe, *Information theoretic learning: Renyi's entropy and kernel perspectives*. Springer Science & Business Media, 2010, available at: Google Scholar.

[28] R. Kamimura, "Information-Theoretic Competitive Learning with Inverse Euclidean Distance Output Units," *Neural Process. Lett.*, vol. 18, no. 3, pp. 163–204, Dec. 2003, doi: 10.1023/B:NEPL.0000011136.78760.22.

[29] D. E. Rumelhart and D. Zipser, "Feature Discovery by Competitive Learning*," *Cogn. Sci.*, vol. 9, no. 1, pp. 75–112, Jan. 1985, doi: 10.1207/s15516709cog0901_5.

[30] DeSieno, "Adding a conscience to competitive learning," in *IEEE International Conference on Neural Networks*, 1988, pp. 117–124 vol.1, doi: 10.1109/ICNN.1988.23839.

[31] A. Banerjee and J. Ghosh, "Frequency-Sensitive Competitive Learning for Scalable Balanced Clustering on High-Dimensional Hyperspheres," *IEEE Trans. Neural Networks*, vol. 15, no. 3, pp. 702–719, May 2004, doi: 10.1109/TNN.2004.824416.

[32] T. Kohonen, *Self-Organizing Maps*, 1995, vol. 30, doi: 10.1007/978-3-642-97610-0.

[33] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

[34] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996, doi: 10.1007/BF00058655.

[35] L. Breiman, "Random forests," *Mach. Learn.*, 2001, doi: 10.1023/A:1010933404324.

[36] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, 2001, doi: 10.2307/2699986.

[37] J. Friedman, R. Tibshirani, and T. Hastie, "Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors)," *Ann. Stat.*, vol. 28, no. 2, pp. 337–407, Apr. 2000, doi: 10.1214/aos/1016120463.

[38] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, "Boosting the margin: a new explanation for the effectiveness of voting methods," *Ann. Stat.*, vol. 26, no. 5, pp. 1651–1686, Oct. 1998, doi: 10.1214/aos/1024691352.

[39] R. E. Schapire and Y. Singer, "Improved Boosting Algorithms Using Confidence-rated Predictions," *Mach. Learn.*, vol. 37, no. 3, pp. 297–336, Dec. 1999, doi: 10.1023/A:1007614523901.

[40] J. W. Foreman, *Data smart: Using data science to transform information into insight*. John Wiley & Sons, 2013, available at: Google Scholar.