sciendo

JAISCR, 2019, Vol. 9, No. 2, pp. 123 – 147
10.2478/jaiscr-2018-0029

SUPPOSED MAXIMUM MUTUAL INFORMATION FOR IMPROVING GENERALIZATION AND INTERPRETATION OF MULTI-LAYERED NEURAL NETWORKS

Ryotaro Kamimura

IT Education Center, Tokai University
4-1-1 Kitakaname, Hiratsuka, Kanagawa 259-1292, Japan

E-mail: ryo@keyaki.cc.u-tokai.ac.jp

Submitted: 6th February 2018; Accepted: 13th August 2018

Abstract

The present paper[1] aims to propose a new type of information-theoretic method to maximize mutual information between inputs and outputs. The importance of mutual information in neural networks is well known, but the actual implementation of mutual information maximization has been quite difficult to undertake. In addition, mutual information has not extensively been used in neural networks, meaning that its applicability is very limited. To overcome the shortcoming of mutual information maximization, we present it here in a very simplified manner by supposing that mutual information is already maximized before learning, or at least at the beginning of learning. The method was applied to three data sets (crab data set, wholesale data set, and human resources data set) and examined in terms of generalization performance and connection weights. The results showed that by disentangling connection weights, maximizing mutual information made it possible to explicitly interpret the relations between inputs and outputs.

Keywords: mutual information; disentanglement; generalization; interpretation

1 Introduction

The present paper proposes a new type of information-theoretic method to maximize mutual information in neural networks. Mutual information has played an important role in neural networks since Linsker proposed his maximum information presentation or maximum mutual information preservation principle [2, 3, 4, 5] to describe neural networks in visual systems. Though many methods have been proposed since his original proposal, mutual information maximization has not necessarily been successfully applied to neural networks [6, 7, 8, 9, 10, 11, 12]. For example, Linsker's pioneering works were mainly concerned with the optimization of entropy, which is only one component of mutual information maximization. In addition, the majority of information-theoretic methods, such as the independent component analysis [13, 14, 15, 16], blind source separation [17, 18], and factorial coding [19, 20], among others, are mainly concerned with entropy maximization or mutual information minimization. This means that these methods are very passive in extracting important features. For example, they are used to make components as statistically independent as possible, expecting that independent components will eventually detect important features.

[1] This paper is an extended version of the paper presented during the proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI 2017)[1]

However, this is not always true, because the extraction of independent components does not naturally include the operation of extracting features [19, 20]. These methods implicitly assume that the property of independence is accompanied by the extraction of important features. Thus, it is necessary to incorporate the operation of feature extraction in addition to the extraction of independent components. For this, mutual information maximization is relatively active in relating components to their corresponding features. This is because all components should be equally used in maximizing mutual information, which implies that neurons are independent. However, the use of mutual information maximization has not been widely pursued, save for the very important and sophisticated contribution to the practical implementation of mutual information by Principe et al. [10, 11]. This is because there has been some difficulty in its implementation due to its complex computational procedures, in particular, the computation of conditional probability and the corresponding conditional entropy.

Thus, there is an urgent need to simplify the computational procedures of mutual information. To do so, we have so far introduced an information-theoretic method called "potential learning" to maximize information content on neurons [21, 22, 23, 24]. By supposing the importance or potentiality of connection weights or neurons before learning, the computational procedures of mutual information are greatly simplified. In addition, with potential learning we can specify which neurons should be fired. However, it has been shown that these methods nevertheless have some difficulty in dealing with mutual information. For example, in [25], mutual information was described by producing multiple and independent neural networks and then considering all produced neural networks. This sort of method has basically aimed to reduce the number of important neurons or connection weights as much as possible by maximizing information content in neurons. As was mentioned, mutual information does not simply imply a reduction in the number of neurons; rather, it implies that all neurons are used equally on average. Thus, at the present stage of potential learning, it is impossible to describe the states produced by mutual information maximization. Because much difficulty is involved in computing mutual information, some simplification is certainly warranted.

Thus, we propose here a more simplified method, where mutual information is supposed to be already maximized. This means that a network configuration to attain maximum mutual information is first created in the early stages of learning, and it is then enhanced in the later stages. More concretely, we suppose that each neuron is connected with different neurons, while all neurons are equally used on average. One of the main merits of the present method is that any tedious computation of mutual information is completely eliminated. Thus, the method can realize mutual information maximization in the most simplified way.

This paper is organized as follows. In Section 2, we present how to formulate mutual information by using the absolute values of weights or potentiality. We also present the supposed maximum mutual information. Next, we present how to assimilate the potentiality in connection weights. Finally, we briefly present how to unify weights in multi-layered neural networks, namely, collective weights. In Section 3, we present three experimental results of the method's being applied to the crab data set, wholesale data set, and human resources data set. In all experimental results, we show that mutual information could be increased by the simplified method. Specifically, the number of strong connection weights was reduced and led to better generalization performance. Then, by the collective weights, multi-layered neural networks were simplified into ones without hidden layers, which made it possible to interpret final connection weights by logistic regression analysis.

2 Theory and Computational Methods

2.1 Mutual Information

Mutual information maximization fundamentally aims to create a network with ideal connection weights, as portrayed in Figure 1(b). Let us compute mutual information by using connection weights from the input to the first hidden layer, shown in Figure 1(a). The same procedures can be applied for higher layers as well. Connection weights from the j_0th input node ($j_0 = 1, 2, ..., J_0$) to the j_1th hidden neuron ($j_1 = 1, 2, ..., J_1$) of the first hidden layer are represented by $w_{j_1 j_0}$. Follow-

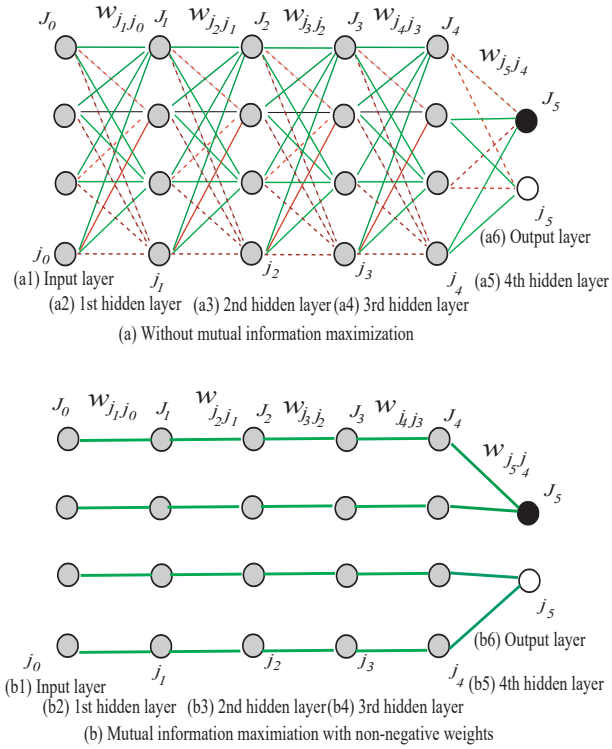

Figure 1. Neural networks without maximum mutual information (a) and with maximum mutual information (b)

ing potential learning, we try to determine the importance or potentiality of weights by the absolute values of weights [26]. This means that when the absolute values of weights become larger, the potentiality is also higher. Then, the potentiality is computed by

$$u_{j_1 j_0} = \left| w_{j_1 j_0} \right|. \tag{1}$$

Then, we should compute the probability of neurons with higher potentiality. By normalizing the potentiality, we have the probability for a neuron to have higher conditional potentiality.

$$p(j_1|j_0) = \frac{u_{j_1 j_0}}{\sum_{j_1=1}^{J_1} u_{j_1 j_0}} \tag{2}$$

The probability of the j_1th hidden neuron for higher potentiality is obtained by

$$p(j_1) = \sum_{j_0=1}^{J_0} p(j_0) p(j_1|j_0). \tag{3}$$

Using these equations, mutual information between the inputs and outputs in the first hidden layer is defined by

$$\begin{aligned} I_{10} &= \sum_{j_1=1}^{J_1} \sum_{j_0=1}^{J_0} p(j_0)\ p(j_1|j_0) \log \frac{p(j_1|j_0)}{p(j_1)} \\ &= -\sum_{j_1=1}^{J_1} p(j_1) \log p(j_1) \\ &+ \sum_{j_1=1}^{J_1} \sum_{j_0=1}^{J_0} p(j_0)\ p(j_1|j_0) \log p(j_1|j_0) \\ &= H_1 - H_{10}. \end{aligned} \tag{4}$$

When this mutual information is maximized, all the neurons of the first hidden layer tend to have the same potentiality on average, and at the same time, each node of the input layer has higher conditional potentiality for a specific neuron in the first hidden layer, which is close to the network shown in Figure 1(b).

2.2 Supposed Mutual Information

Because it is almost impossible to compute mutual information directly, we propose here a simplified method, where mutual information is supposed to be already maximized. In the equation (4), to ob-

tain maximum mutual information, the entropy H_1 should be maximized, and at the same time the conditional entropy H_{10} should be minimized. For simplicity's sake, the first term of entropy, H_1, is supposed to be maximized, meaning that all neurons' potentialities are equal on average. In addition, the j_0th input node also takes equiprobability; then, the equation can be

$$I_{10} = \log J_1 + \frac{1}{J_0} \sum_{j_1=1}^{J_1} \sum_{j_0=1}^{J_0} p(j_1|j_0) \log p(j_1|j_0). \quad (5)$$

Following this, all we have to do to maximize mutual information is minimize conditional entropy, meaning that each neuron should respond to a specific neuron in a higher layer. In addition, we try to force connection weights to be as positive as possible for better interpretation, as shown in Figure 1(b). Thus, in computing conditional probability, we use the connection weights themselves to realize mutual information maximization. Because for each input node j_0, a specific hidden neuron j_1 has higher conditional probability, we have the following equation to minimize conditional entropy

$$p(j_1^* \mid j_0) = \begin{cases} 1 & j_1^* = \text{argmax}_{j_1} w_{j_1 j_0} \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

This means that a weight should be retained only when it is to a specific hidden neuron, has the highest value and all others are zero. In actual implementation, the strict application of this conditional probability tends to cause some trouble, particularly when the conditional probability is close to zero. Thus, we need to add some constants to the conditional potentiality as $p(j_1 \mid j_0) + \varepsilon$. In the experiments presented here, ε is set to the default small value in Matlab.

2.3 Mutual Information Assimilation

In Figure 2(a), In the first epoch of the first learning step denoted by the superscript (1,1), we apply the conventional learning method. The final output from the first hidden layer is computed by

$${}^{(1,1)}v_{j_1}^s = \text{tansig}\left(\sum_{j_0=1}^{J_0} {}^{(1,1)}w_{j_1 j_0} x_{j_0}^s \right), \quad (7)$$

where ${}^{(1,1)}w_{j_1 j_0}$ denote weights from the j_0th input node to the j_1th hidden neuron and $x_{j_0}^s$ is the sth input $(s = 1, 2, ..., S)$ for the j_0th input node. For the other intermediate layers, the same procedures are applied, and the final output is computed by

$${}^{(1,1)}o_{j_5}^s = \text{softmax}\left(\sum_{j_5=1}^{J_5} {}^{(1,1)}w_{j_5 j_4} {}^{(1,1)}v_{j_5}^s \right), \quad (8)$$

where "softmax" denotes the function defined by $\exp(x)/\sum\exp(x)$. The cost function is the cross-entropy function

$${}^{(1,1)}E = -\sum_{s=1}^{S} \sum_{j_5=1}^{J_5} t_{j_5}^s \log {}^{(1,1)}o_{j_5}^s, \quad (9)$$

where $t_{j_5}^s$ are the targets for the corresponding outputs. Then, these procedures continue up to the T_1th epoch in Figure 2(a2), and we compute the conditional probability

$${}^{(1,T_1)}p(j_1^* \mid j_0) = \begin{cases} 1 & j_1^* = \text{argmax}_{j_1} {}^{(1,T_1)}w_{j_1 j_0} \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

This conditional probability or potentiality is assimilated in the first epoch of the second step denoted by (2,1) in Figure 2(b1) as

$${}^{(2,1)}w_{j_1^* j_0} = {}^{(1,T_l)}p(j_1^* \mid j_0) {}^{(1,T_1)}w_{j_1^* j_0}. \quad (11)$$

These steps continue to the T_2th epoch in Figure 2(b2), and finally, the conditional probability is again computed and the final epoch produces the conditional probability for the next learning step and so on.

2.4 Collective Interpretation

Finally, we should note the well-known interpretation problem, which is addressed by the new method of collective weights. Since early research on neural networks, it has been very hard to interpret the final results obtained by learning [27, 28, 29, 30, 31]. In the age of multi-layered neural networks, this problem has become more serious [32, 33]. To address this problem, in the present paper we introduce collective weights, where all hidden weights are treated collectively. In the end, multi-layered neural networks can be reduced to the most simplified ones without any hidden layers. Then, it becomes possible to interpret these simplified neural networks. This method aims to interpret relations between inputs and outputs as well as the overall inference mechanisms of multi-layered neural networks.

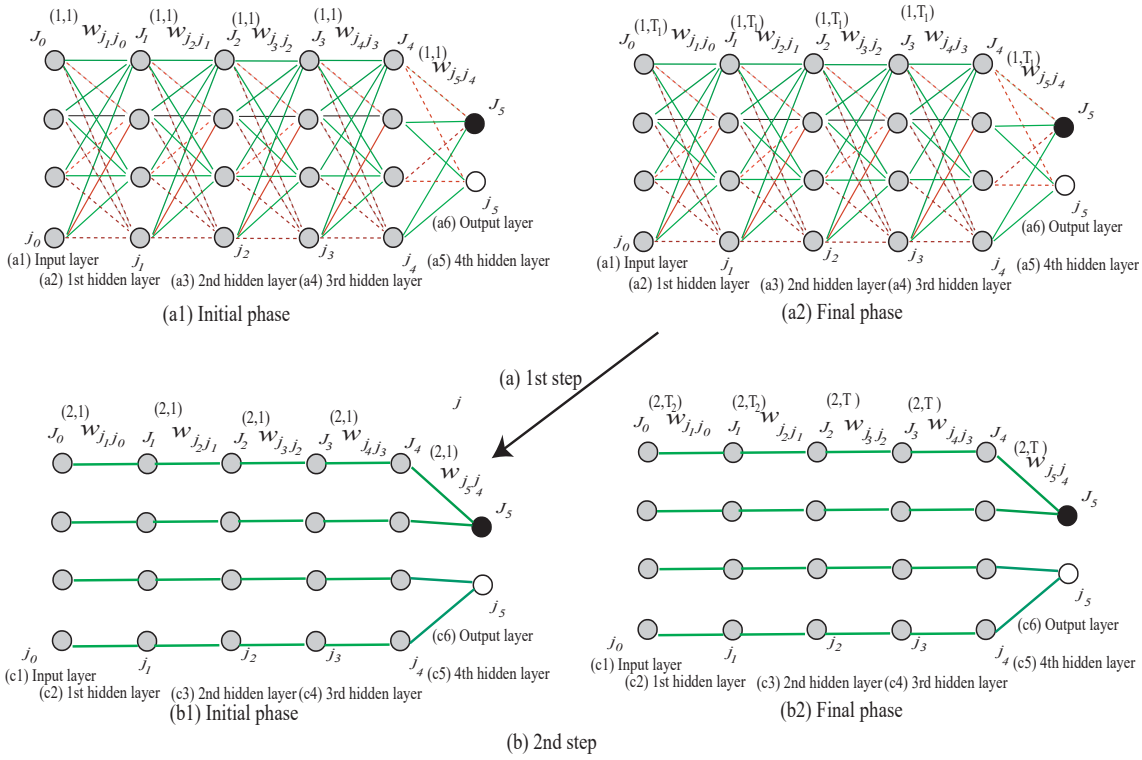

Figure 2. Computational procedures for mutual information assimilation with four hidden layers.

Then, the collective weights for networks with four hidden layers are computed by summing up all weights in the intermediate layers. The collective weight from the input to the output layer is computed by

$$w_{j_5 j_0} = \tag{12}$$

$$\sum_{j_1=1}^{J_1}\left(\sum_{j_2=1}^{J_2}\left(\sum_{j_3=1}^{J_3}\left(\sum_{j_4=1}^{J_4} w_{j_5 j_4} w_{j_4 3}\right) w_{j_3 j_2}\right) w_{j_2 j_1}\right) w_{j_1 j_0}.$$

As shown in Figure 3(a), all hidden layers are collectively treated by the equation (12). Then, the multi-layered neural networks are reduced to their most simplified forms, as shown in Figure 3(b1). Then, we can interpret relations between inputs and outputs, just by examining collective weights. However, as shown in Figure 3(b1), there are still many connection weights, which makes it hard to interpret all connection weights. At this point, mutual information maximization is introduced in Figure 3(a2), where the entanglement of connection weights is eliminated and the weights are simplified. Then, from these connection weights in Figure 3(a2), we can obtain simplified and disentangled collective weights in Figure 3(b2) for better interpretation.

3 Results and Discussion

3.1 Crab Data Set

3.1.1 Experimental Outline

The well-known crab data set was used to demonstrate the performance of our method[2]. The number of variables and patterns were 6 and 200, respectively. The variables represented six identifying attributes of crabs, such as species, frontal lip, rear width, length, width, and depth. The goal of neural networks for this data set is to infer the gender of the crabs (male or female) based on the variables. Half of the data set was used for training the neural networks, and the remaining half was evenly divided into the validation and testing data sets. For easy reproduction of the present results, we used the default parameter values of the Matlab neural

[2]https://jp.mathworks.com/help/nnet/gs/neural-network-toolbox-sample-data-sets.html?lang=en

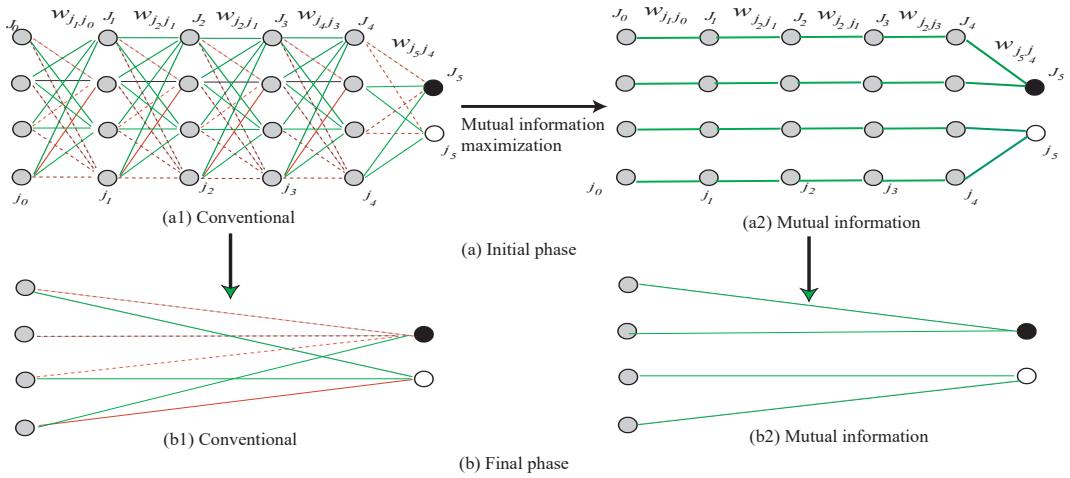

Figure 3. Collective weights

network package.

3.1.2 Mutual Information Maximization

Mutual information is composed of entropy and conditional entropy. Thus, we show here that entropy kept its relatively high values, while conditional entropy gradually deceased. Then, mutual information increased when the number of steps increased. Figure 4 shows entropy (1), conditional entropy (2), and mutual information (3) for the first hidden layer (a) through to the output layer (g). Entropy for the weights for the first layer in Figure 4(a) gradually decreased when the number of steps increased. However, decrease in the entropy became slower for the connection weights to the third hidden layer in Figure 4(b) and to the weights to the eleventh layer in Figure 4(f). When we closely examined mutual information, we could see that it was relatively larger for hidden to hidden layers compared to input to output layers. This is because the input and output layers are directly linked to inputs and outputs. In addition, mutual information slightly fluctuated for the output layer, because it was most explicitly related to the error minimization between the targets and outputs. On the other hand, conditional entropy decreased constantly for all layers in Figure 4(a)-(f). Since the mutual information was obtained by subtracting conditional entropy from entropy, naturally, mutual information increased gradually in Figure 4(a)-(f).

The results show that the present simplified method could greatly increase mutual information by supposing that entropy (H_1) is already maximized. All this leaves for us to do is to minimize conditional entropy (H). The present results confirmed the validity of this presumption. In all layers, entropy tended to be stable with relatively high values.

3.1.3 Connection Weights and Generalization

Figure 5 shows connection weights to the first hidden layer (a) through to the output layer (g) for the first step (1) and the final step (4). As can be seen in the figure, in the second step, connection weights close to those at the final step in Figure 5(4) were already obtained, though minor weights were observed. For the third step, minor connection weights were gradually eliminated, and from this step on, connection weights became stable and changed little. Figure 6 shows connection weights from the input node to the first hidden layer in the Hinton diagram and in the ordinary diagram. In the ordinary diagram in Figure 6(b), only the strongest connection weights between two layers were given. As can be seen in the figure, all the strongest weights were separately connected with

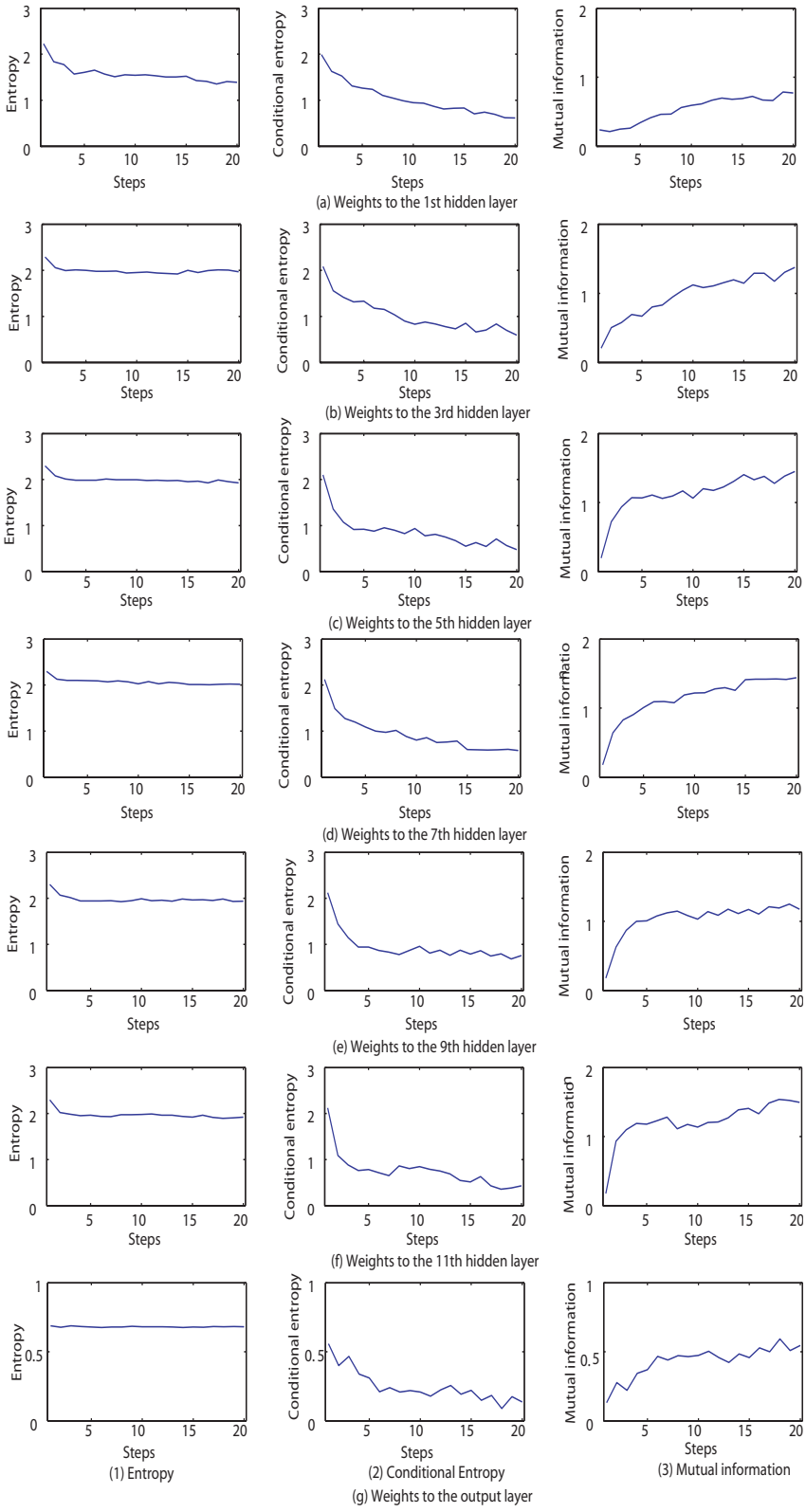

Figure 4. Entropy (1), conditional entropy (2), and mutual information (3) for weights to the first hidden layer (a) through to the output layer (f) for the crab data set.

different neurons. Thus, disentanglement was almost accomplished.

Figure 7(a) shows generalization errors as a function of the number of hidden layers. Errors by the mutual information method gradually decreased when the number of layers gradually increased. On the other hand, generalization errors by the non-mutual information method increased when the number of layers increased. In particular, when the number of layers became higher, generalization errors rapidly increased. Figure 7(b) shows the number of steps with minimum validation errors as a function of the number of layers. As can be seen, the number of steps by mutual information maximization was much higher than that by the non-mutual information method. This means that over-training was restricted by the mutual information method.

In addition, in the field of deep learning, much recent attention has been paid to multi-layered neural networks. Then, mutual information should play more important roles in increasing the performance of multi-layered neural networks. For example, in multi-layered neural networks, many layers, and neurons are used, and we must face the problem of entanglement [34], where many neurons or connection weights tend to be entangled with each other. The entanglement is one of the main causes of difficulty in improving generalization and interpreting connection weights. As mentioned earlier, it is not sufficient to make components as independent as possible; rather, we need to relate the components with their corresponding ones. To disentangle connection weights or neurons, we introduced mutual information between neurons here. When mutual information is maximized, all neurons should on average be used equally. On the other hands, each neuron should be explicitly connected with different neurons. Thus, all neurons are used on average, and all neurons are connected differently with the other neurons. Thus, we can say that all the neurons can be disentangled and be expected to transmit information on inputs and errors explicitly and efficiently.

3.1.4 Interpreting Collective Weights

Figure 8(a) shows the collective weights for the first output neuron, where input neuron No.3 was the largest. For the second output neuron in Figure 8(b), input neuron No.3 had the largest absolute value, but it was negative. By the non-mutual information method, the collective weights in Figure 8(b) were almost the same as that by the mutual information method. Mutual information focused on the first neuron, while non-mutual information used both output neurons. In addition, we can see that the correlation coefficients between connection weights to two output neurons were slightly reduced to -0.9458 by the mutual information method from -0.9831 by non-mutual information. The regression coefficients by the logistic regression analysis showed that variable No.3 had the largest value, but some other coefficients had relatively large values as well. Thus, the present method shows the relations between inputs and outputs more explicitly. The results show that the collective weights were effective in extracting relations between inputs and inputs. In particular, mutual information maximization seemed to make important relations more explicit.

3.2 Wholesale Data Set

3.2.1 Experimental Outline

The dataset refers to clients of a wholesale distributor. It includes the annual spending in monetary units on diverse product categories [35]. The number of input patterns was 440 and the number of input variables were seven. We examined which variables contributed to the channel differences, namely, the hotel and coffee shops channel and the retail channel. The number of hidden layers was increased from one to ten layers. Fifty percent of the data set was used for training, 25 percent for validation and the remaining 25 percent for testing. We used the Matlab neural network package with all default parameter values for easy reproduction of the results presented here.

3.2.2 Mutual Information Maximization

The experimental results show that in the lower layers, conditional entropy decreased sufficiently, and thus mutual information increased gradually. For the higher layers, conditional entropy decreased rapidly, and correspondingly, mutual information rapidly increased and remained constant. Figure 9 shows entropy, conditional entropy and mutual information for the weights to the first hidden layer

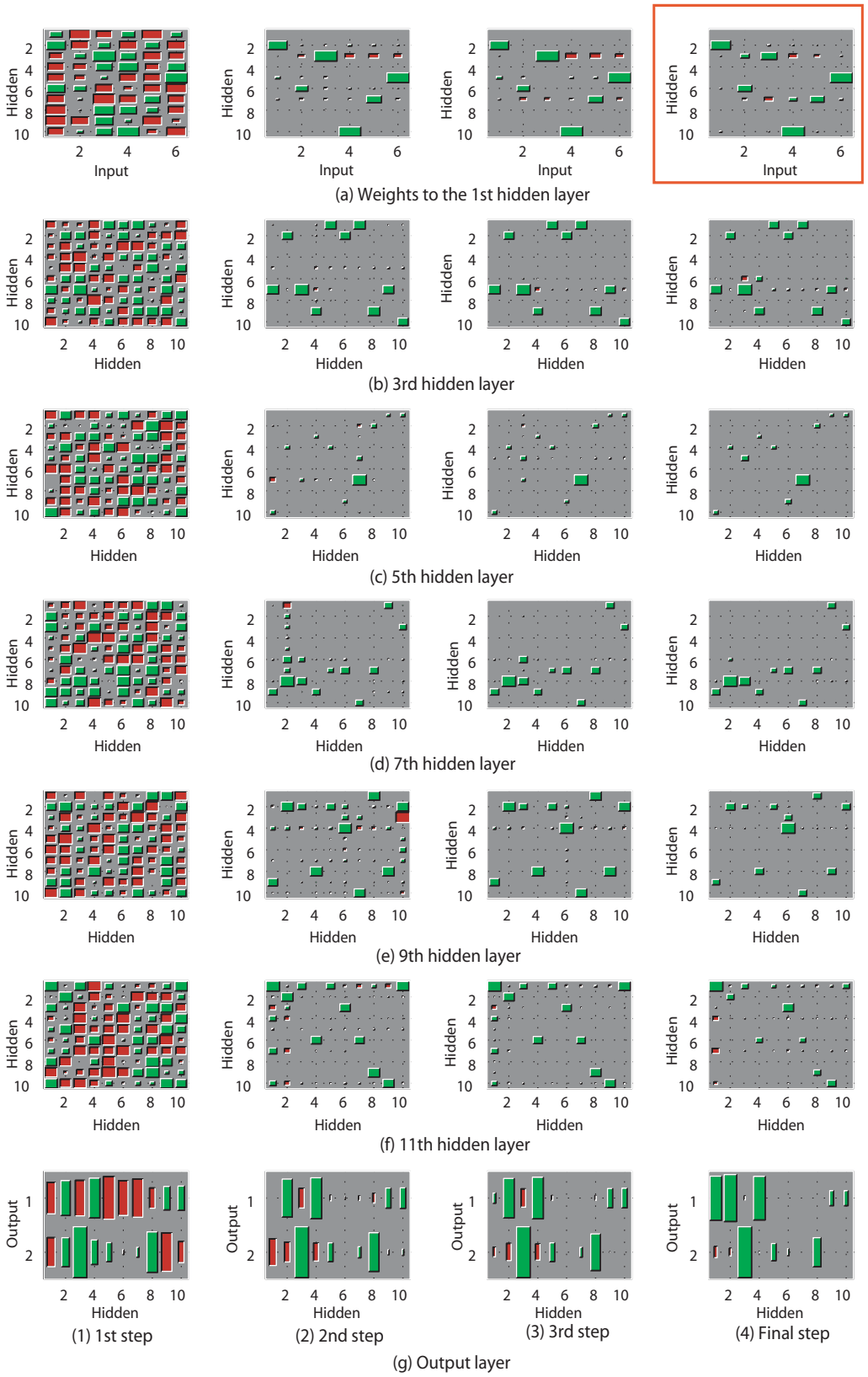

Figure 5. Connection weights from the first (a) to output (f) layer, and for the first step (1) to the final step (4) for the crab data set.

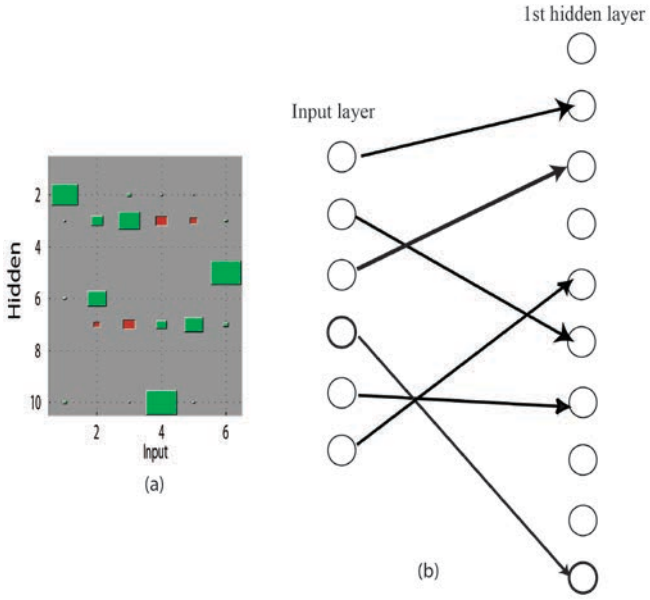

Figure 6. Connection weights from the inputs to the first hidden layer in a Hinton diagram (a) and in a conventional diagram for the weights surrounded by a red square in Figure 5 for the crab data set.

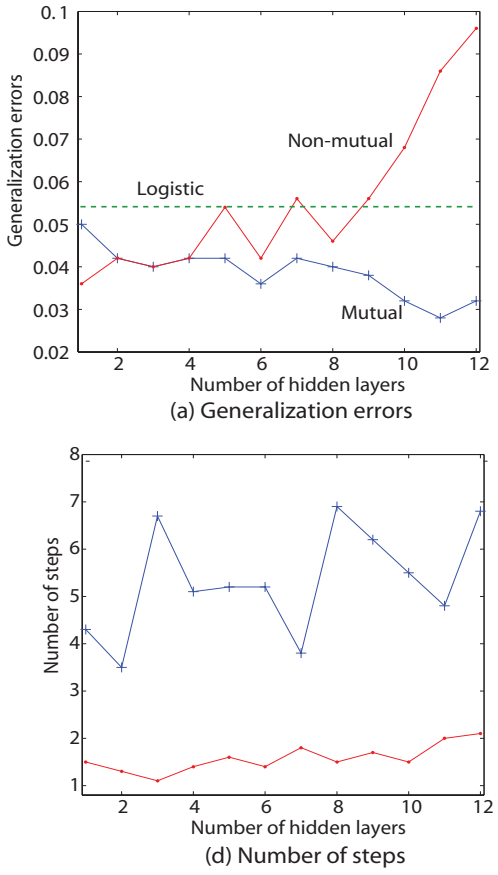

Figure 7. Generalization errors (a) and the number of steps to reach the minimum validation errors (b) by mutual, non-mutual, and logistic regression for the crab data set.

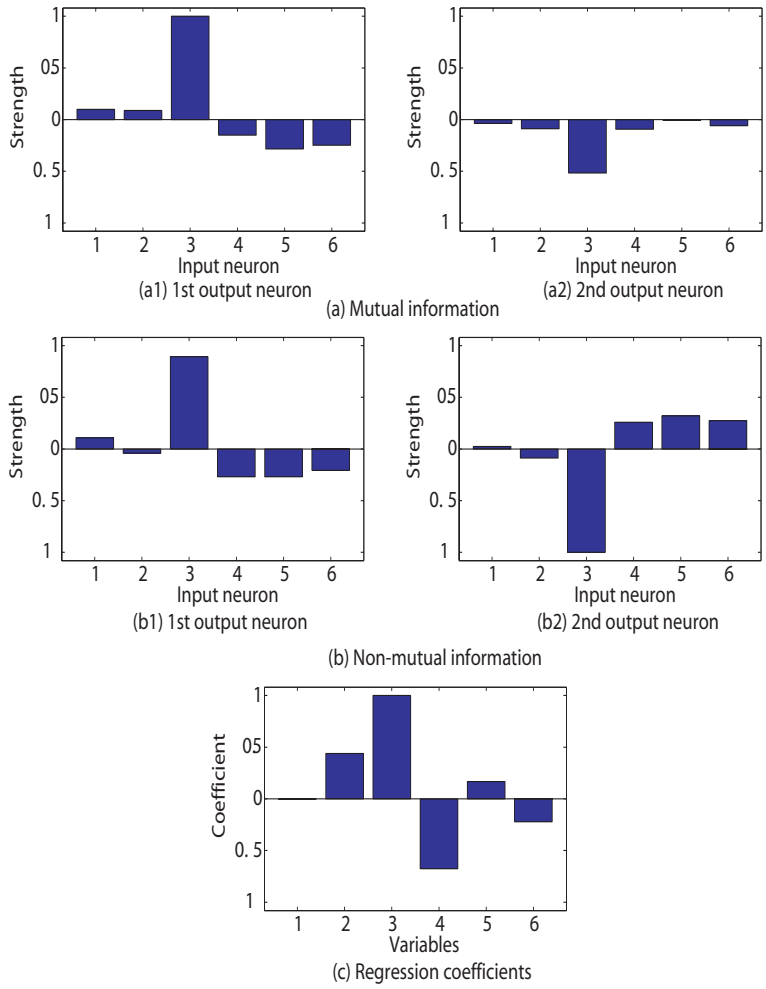

Figure 8. Collective weights into the first (1) and the second (2) output neuron by mutual (a) and non-mutual (b) information method and regression coefficients (c) by the logistic regression analysis for the crab data set. The correlation coefficients between connection weights to two output neurons were -0.9458 and -0.9831 by the mutual information and non-mutual information method, respectively.

(a) through to the output layer (f). As can be seen in Figure 9(a), entropy decreased gradually for the weight to the first hidden layer, but conditional entropy deceased more rapidly. Then, mutual information gradually increased in the end. For the weights to the third hidden layer in Figure 9(b), entropy decreased at the beginning and remained to be almost constant, while conditional entropy decreased gradually. Then, mutual information gradually increased. For the weights to the fifth hidden layer in Figure 9(c) to those to the output layer in Figure 9(f), conditional entropy decreased immediately and remained almost constant. Correspondingly, mutual information increased rapidly and remained constant.

The results show that connection weights from the input nodes to the first hidden layer in Figure 9(a) and weights to the output layer in Figure 9(f) were not easily controlled by mutual information maximization. For the weights to the first hidden layer in Figure 9(a), entropy tended to decrease, while the entropy tended to be constant for the other cases. This shows that connection weights to the first hidden layer were more strongly influenced by inputs. On the other hand, for the connection weights to the output layer in Figure 9(f), the values of entropy, conditional entropy, and mutual information were relatively smaller, because they were used to decrease errors between targets and outputs. In addition, the number of output neurons was only two, while the number of hidden neurons was ten; this made it impossible to increase mutual information. The results showed that mutual information could be increased more flexible in the intermediate layers.

3.2.3 Comparison of Connection Weights

The results show that, by supposing maximum mutual information, many strong weights in the first step disappeared, and only in the third step, sparse and positive weights were obtained by the present method. In addition, it was confirmed that the majority of neurons were connected with different neurons, disentangling neurons and connection weights. Figure 10 shows weights from the first hidden layer (a) through to the output layer (f). In the first step in Figure 10(1), because no mutual information was applied, many strong positive and negative weights were observed. In the second step in Figure 10(2), the number of strong connection weights was reduced greatly, but some smaller positive and negative weights could be seen. In the third step in Figure 10(3), the majority of small connection weights disappeared, and only a small number of connection weights remained strong. For the final step in Figure 10(4), the number of minor connection weights, and in particular, negative weights, were further reduced, and almost all connection weights were positive. These results show how that three learning steps were enough to reach a state close to the final state. In the later steps, connection weights were fine-tuned to eliminate minor weights and to make connection weights positive.

Figure 11 shows connection weights from the input layer to the first hidden layer in Figure 10(a4). The first input node was not explicitly connected with the hidden neurons, and the fourth and the sixth input nodes were connected with the same sixth hidden neuron. All the other input nodes were connected with different hidden neurons. This means that unnecessary neurons or nodes could be automatically eliminated.

3.2.4 Interpreting Collective Weights

Collective weights by the present method produced quite clear characteristics than the by the method without information maximization. In addition, it turned out that the collective weights were quite similar to the regression coefficients by the logistic regression analysis. Figure 12 shows the collective weights for the wholesale data set. By mutual information maximization, the input variable No.6 had the largest positive value for the first output neurons and negatively for the second output neuron. On the other hand, without mutual information maximization, in addition to variable No.6, several other variables gained some importance. Then, the correlation coefficient between connection weights to two output neurons was reduced from -0.9862 by the non-mutual information method to -0.8754 by mutual information and the non-mutual information method, meaning that connection weights by mutual information maximization were slightly less correlated. Finally, the regression analysis showed the same tendency (that variable No.6 was the most important). These results show that the multi-layered neural networks with mutual information maximization produced al-

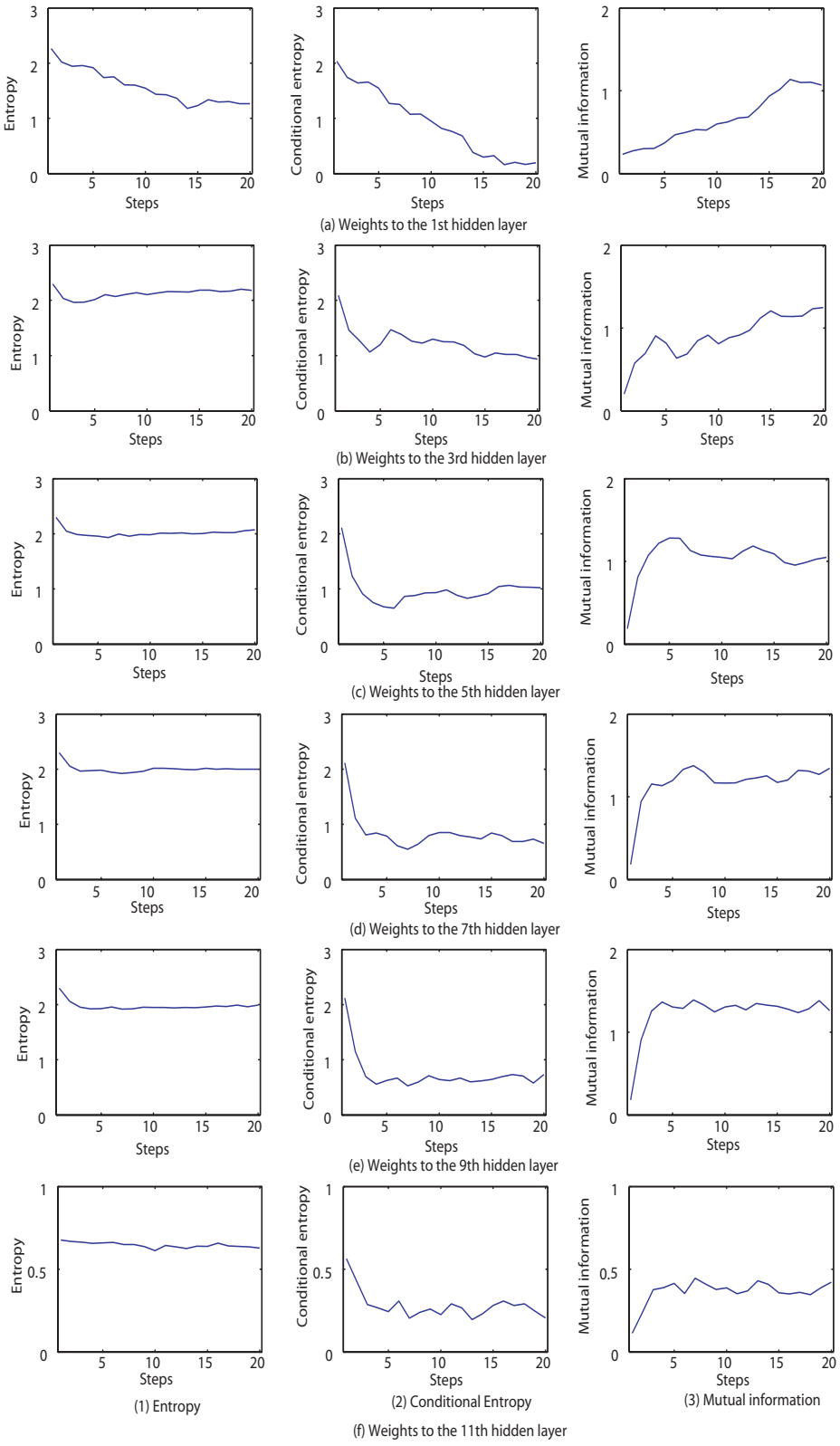

Figure 9. Entropy (1), conditional entropy (2), and mutual information (3) for weights to the first hidden layer (a) through to the output layer (f) for the wholesale data set.

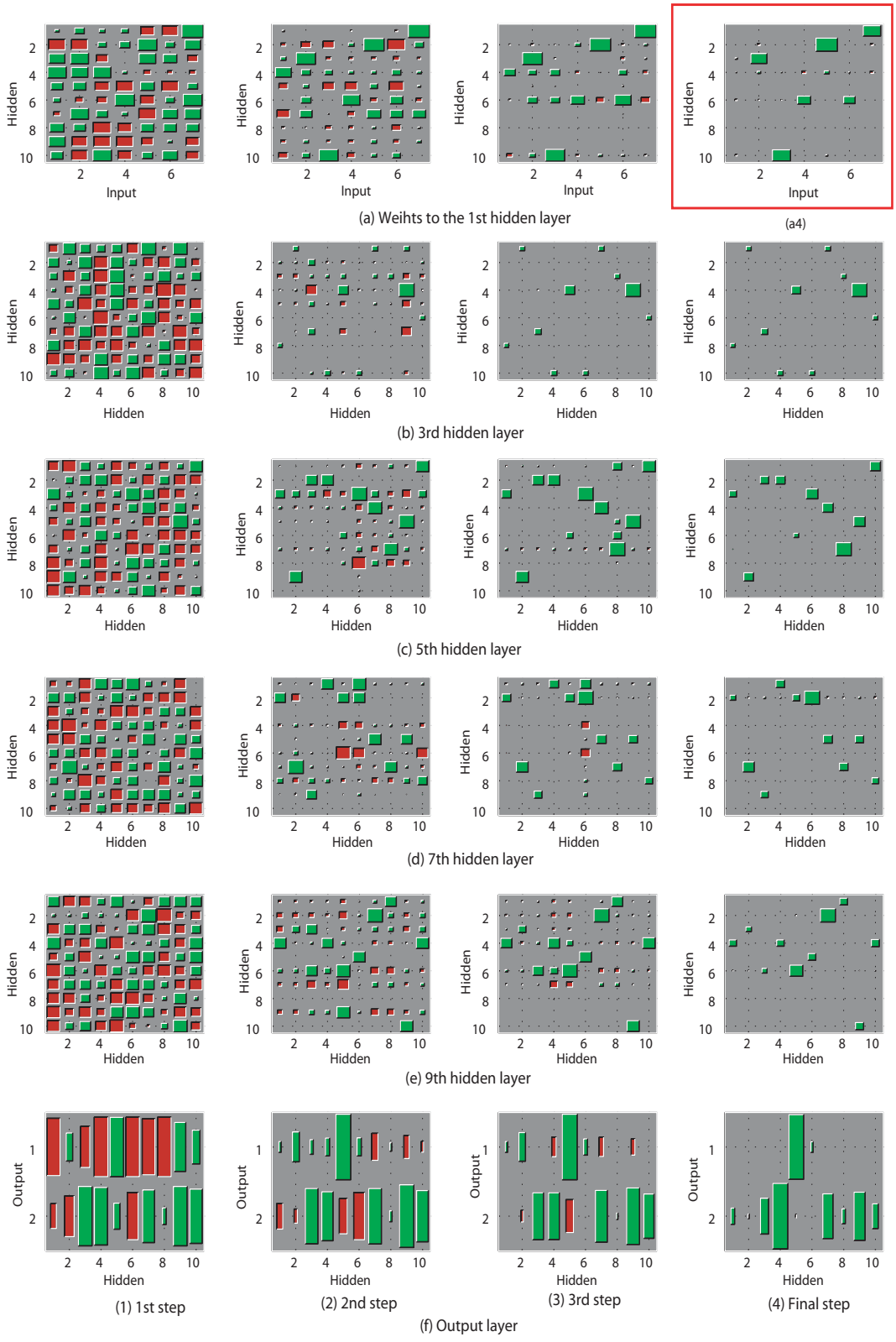

Figure 10. Connection weights from the first (a) through to the output (f) layer and for the first step (1) to the final step (4) for the wholesale data set.

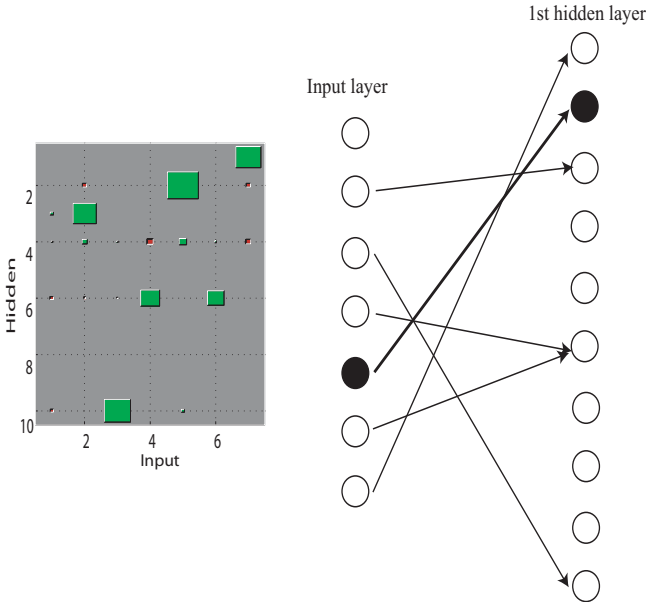

Figure 11. Connection weights in Hinton diagram (a) and the conventional diagram (b), corresponding to those in Figure 10(a4) for the wholesale data set.

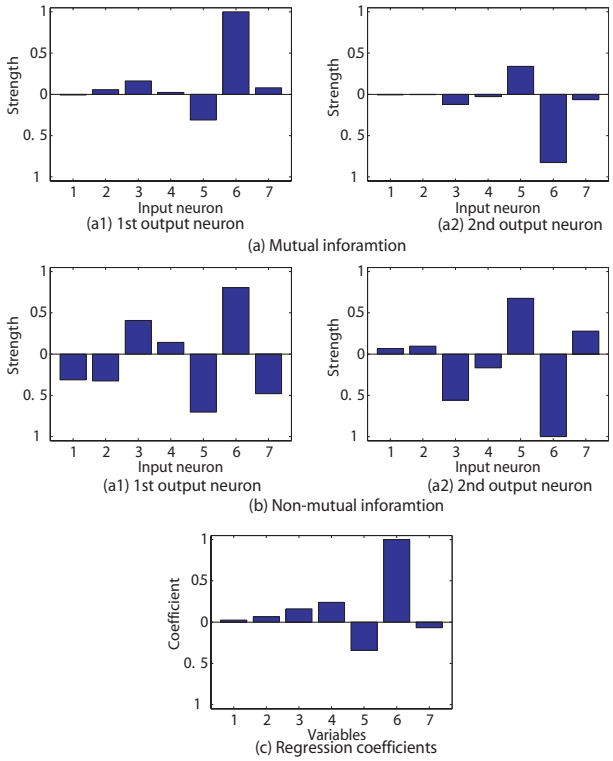

Figure 12. Collective weights into the first (1) and the second (2) output neuron by the mutual (a) and non-mutual (b) information method, and regression coefficients (c) by the logistic regression analysis for the wholesale data set. The correlation coefficient between connection weights to two output neurons were -0.8754 and -0.9862 by mutual information and the non-mutual information method.

most the same input-output relations as that by the regression analysis. In addition, mutual information maximization had the effect of making weights clearer.

3.2.5 Generalization Comparison

Mutual information maximization produced the best generalization performance in terms of average, minimum, and maximum values. Table 1 shows the summary of generalization errors by the three methods. The lowest generalization error of 0.0909 was obtained by mutual information maximization with six hidden layers. This method also obtained the lowest standard deviation of 0.0178 with ten hidden layers. The lowest minimum error of 0.0545 was obtained with two hidden layers by mutual and non-mutual information. The lowest maximum value of 0.1182 was obtained by information maximization with six and ten hidden layers. The logistic regression analysis produced the second-best average error of 0.0982. One of the main differences between mutual information and non-mutual information was that, for mutual information, the number of steps to reach the minimum validation error was always larger than that for non-mutual information. For example, without mutual information, the number of steps to reach the lowest validation error ranged between 1.1 (two hidden layers) and 2.7 (ten hidden layers). On the other hand, by mutual information maximization, the number of steps for the lowest validation errors increased from 4.5 with four hidden layers to 9.3 with ten hidden layers. This means that mutual information maximization could improve generalization performance by increasing the number of steps without over-training. The over-training was strictly controlled by mutual information maximization.

3.3 Human Resources Data Set

3.3.1 Experimental Outline

The purpose of the Human Resources Data Set was to examine the reason why employees leave companies. [3] The number of inputs was 14,999, and eight variables were used: satisfaction level, last evaluation, number of projects, average monthly hours, time spent at the company, whether they had a work accident, whether they had a promotion in the last five years, and salary. Half of the data set was for training, and the remaining was divided into a 25 percent validation and 25 percent testing data set.

3.3.2 Mutual Information Maximization

The experimental results confirmed that mutual information increased gradually when the number of steps increased because conditional entropy decreased, while entropy values remained high. Figure 13 shows entropy (1), conditional entropy (2), and mutual information (3) as a function of the number of steps. For the weights to the first hidden layer, entropy decreased slightly in Figure 13(a), but conditional entropy decreased further. Thus, mutual information increased gradually. Connection weights directly connected with inputs cannot be easily controlled by mutual information maximization; this is why mutual information for the connection weights to the first hidden layer increased very slowly. For the connection weights to the third hidden layer in Figure 13(b) and to the fifth hidden layer in Figure 13(c), entropy decreased at the beginning and became almost constant, and conditional entropy deceased substantially. Thus, mutual information increased gradually as a function of the number of steps. However, for the connection weights to the seventh hidden layer in Figure 13(d), the entropy was almost constant, but it was smaller than that at the other layers. Because of this, mutual information increased but with relatively smaller values. We think that, in this layer, the number of active hidden neurons became smaller, and several hidden neurons were not actually used. For the connection weights for the weights to the ninth and the output layer in Figures 13(e) and (f), conditional entropy decreased and then remained constant with some fluctuations. Thus, mutual information jumped to certain high values and fluctuated for the later steps of learning. Connection weights directly connected with outputs and close to the output layer cannot be easily changed by mutual information maximization, because of the effect of error minimization is larger than for the other layers.

[3] https://www.kaggle.com/deepakgarg22396/human-resources/data

Table 1. Summary of experimental results on generalization performance for the wholesale data set

Methods	Layers	Steps	Average	Std dev	Min	Max
Logistic			0.0982	0.0186	0.0727	0.1273
Withtout	1	1.2	0.1100	0.0349	0.0636	0.1636
	2	1.1	0.1018	0.0263	**0.0545**	0.1364
	3	2.2	0.1282	0.0480	0.0818	0.2364
	4	1.5	0.1045	0.0243	0.0727	0.1455
	5	1.9	0.1182	0.0231	0.0727	0.1455
	6	2.2	0.1182	0.0291	0.0818	0.1909
	7	1.9	0.1145	0.0291	0.0727	0.1727
	8	1.7	0.1209	0.0261	0.0909	0.1727
	9	2.3	0.1209	0.0231	0.0636	0.1455
	10	2.7	0.1100	0.0295	0.0818	0.1727
With	1	7.7	0.1100	0.0432	0.0636	0.1909
	2	5.3	0.1100	0.0292	**0.0545**	0.1545
	3	6.3	0.1091	0.0187	0.0909	0.1455
	4	4.5	0.1073	0.0249	0.0636	0.1455
	5	7.6	0.1036	0.0202	0.0636	0.1364
	6	5.4	**0.0909**	0.0182	0.0636	**0.1182**
	7	6.9	0.0982	0.0195	0.0727	0.1273
	8	4.7	0.0918	0.0220	0.0636	0.1273
	9	5.8	0.1045	0.0258	0.0818	0.1455
	10	9.3	0.0955	**0.0178**	0.0636	**0.1182**

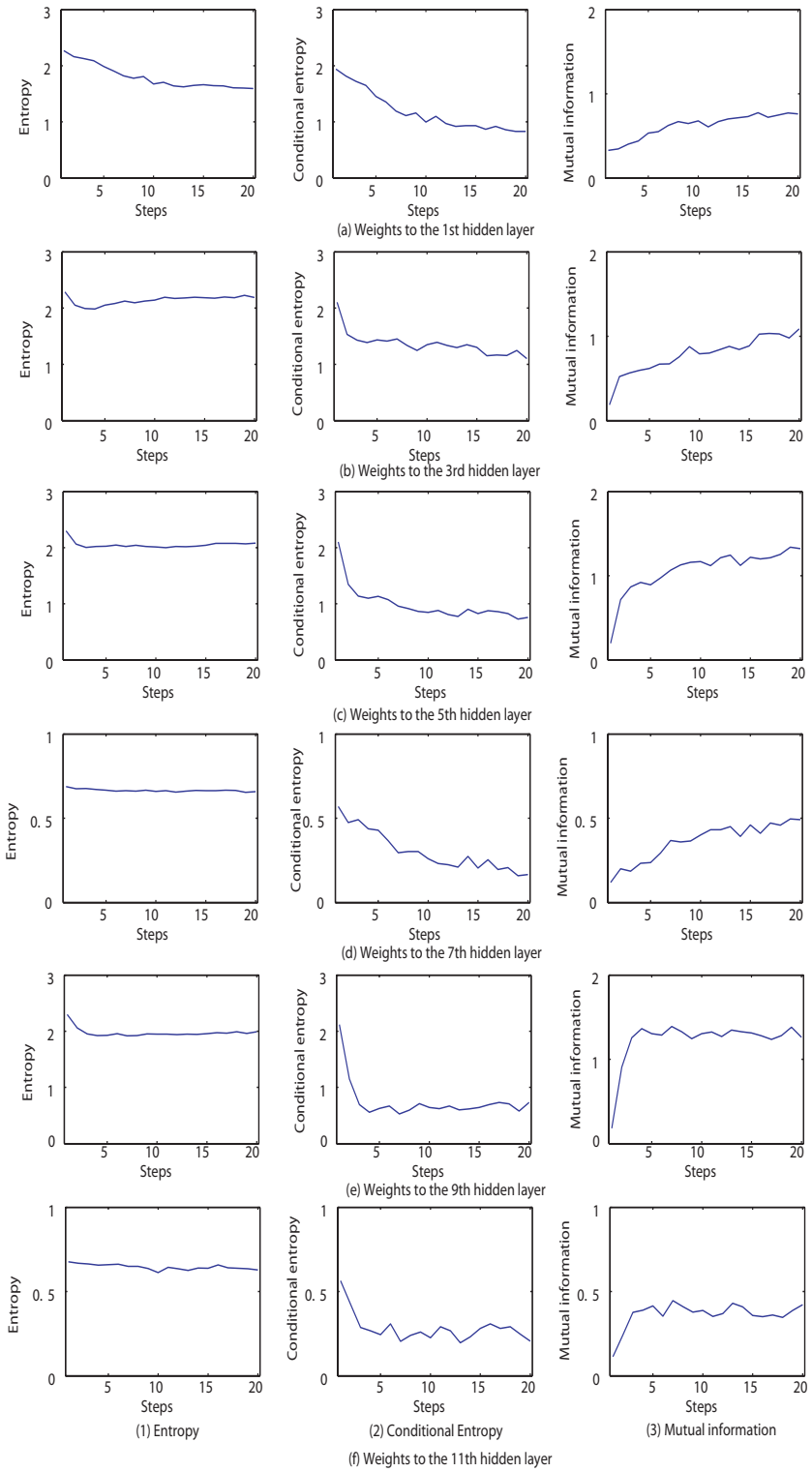

Figure 13. Entropy (1), conditional entropy (2), and mutual information (3) for weights to the first hidden layer (a) through to the output layer (f) for the human resources data set.

3.3.3 Comparison of Connection Weights

Connection weights became close to the final ones with only three steps, where[M1] the number of strong weights was smaller. Figure 14 shows connection weights for the first step (1) to the final step (4). Strong positive and negative weights in Figure 14(1) for the first step became smaller in the second step in Figure 14(2), with some minor negative connection weights. In the third step in Figure 14(3), the minor negative connection weights were almost eliminated, and weights were close to the final connection weights in Figure 14(4). These results show the effectiveness of the present method where, because maximum mutual information is supposed to be already attained from the beginning by assigning one for specific conditional probability, many connection weights are forced to be zero, while only several connection weights remained strong. This supposed maximum information had the effect of accelerating the process of mutual information. The present results confirmed that only several steps are necessary for attaining the maximum information state.

Figure 15 more explicitly shows connection weights between the input layer and the first hidden layer, depicted by red squares in Figure 14. As can be seen in the figure, the first node in the input layer was connected with the third neuron in the first hidden layer. The second input node was connected with the fifth hidden neuron, and the third input node was connected with the eighth hidden neuron. Finally, the fourth input node was connected with one hidden neuron, namely, the sixth hidden neuron. Thus, the majority of input nodes were connected with different hidden neurons, while the remaining ones were not connected with any hidden neurons. The results show that several inputs and hidden neurons were not directly connected. Thus, the present method had the effect of eliminating unnecessary inputs as well as hidden neurons.

3.3.4 Interpreting Collective Weights

The clearest collective weights obtained by mutual information maximization could be easily interpreted. In addition, when collective weights to two output neurons were combined, they became similar to the coefficients by regression analysis. Figure 16 shows collective weights for the first and second output neuron by mutual information (a), non-mutual information (b), and regression coefficients by the regression analysis (c). As shown in Figure 16(a), for the first output neuron, the fifth input node had the largest weights, while the first and the third weights had larger absolute values for the second output neuron in Figure 16(b). On the other hand, collective weights by non-mutual information in Figure 16(b) had different large weights for the first and second output neurons.

The logistic analysis produced coefficients that seemed to be different from those produced by mutual information. However, we could see quite similar characteristics between them. For example, the largest positive coefficient was input variable No.6, which was also the largest positive collective weight for the first output neuron in Figure 16(a). On the other hand, the largest and the second largest absolute value for the negative weights were input variables No.1 and No.3. They had also the largest and the second largest values for the second output neuron. Thus, the largest coefficients of the logistic regression analysis were decomposed into the collective weights to two output neurons by the present method. These results show that two output neurons tended to extract mutually exclusive features. The above results can be made clearer by noting that the correlation coefficient between connection weights to two output neurons was greatly reduced, from -0.9804 to 0.033, by the mutual information maximization method.

Let us interpret the final results more concretely. The first output neuron refers to the probability that employees will leave the company. Variable No.5 represents the time spent at the company. Thus, employees tend to leave the company after they have spent more time there. The second output, in Figure 16(a2), represents the probability with which the employees will not leave the company. The largest absolute weight is connected with variable No.3, representing the number of projects. Thus, as the number of projects increases, the employees tend to be more reluctant to leave the company. The second largest weight was connected with variable No.1, representing satisfaction level. Thus, as the satisfaction level increases, the employees tend to be naturally reluctant to leave the company. The results show that, to prevent the employees from leaving the company, two factors?the satisfaction level and the number of projects?must be seriously taken

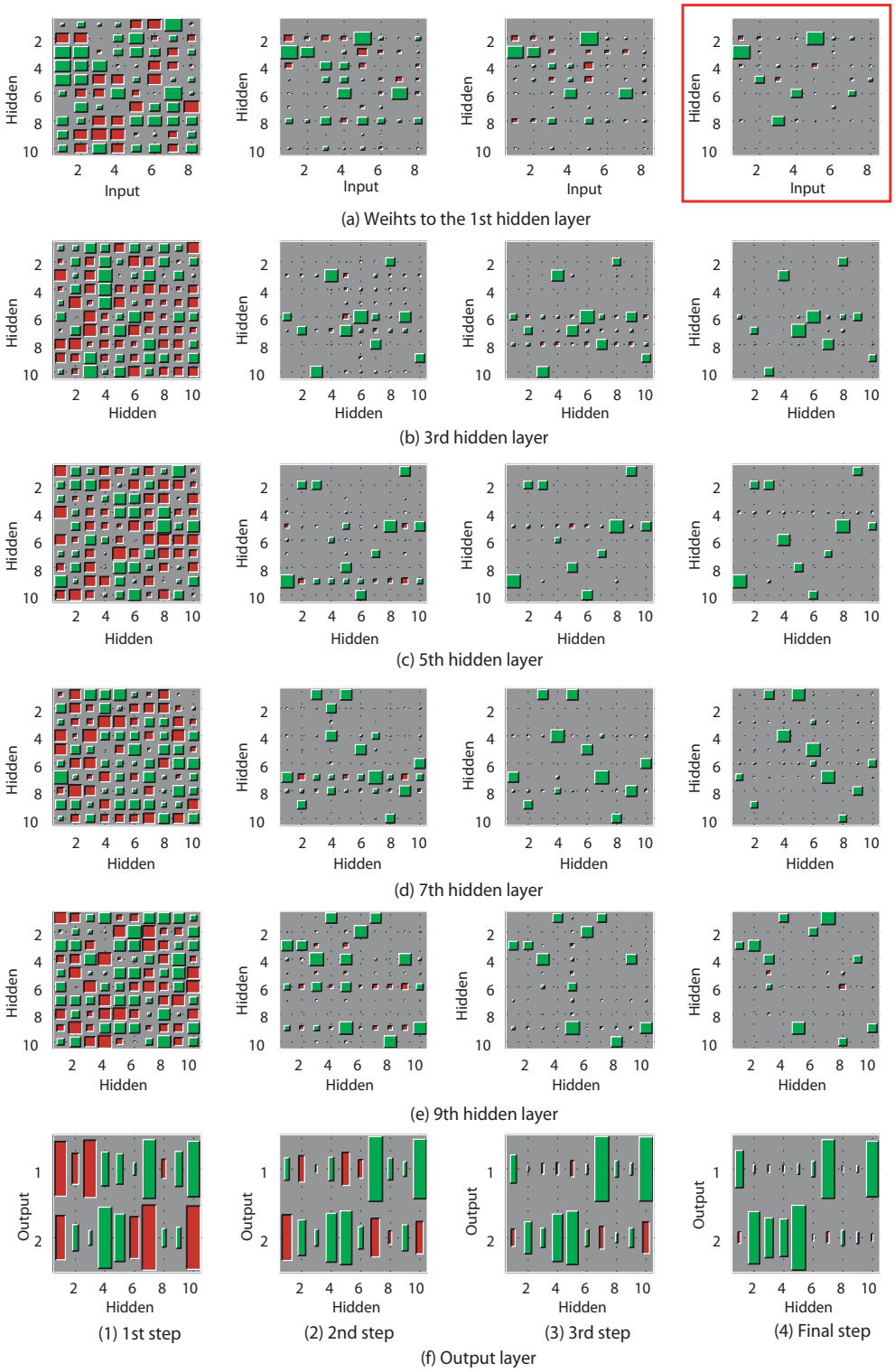

Figure 14. Connection weights from the first (a) through to the output (f) layer for the first step (1), the second (2), the third (3), and the 20th step (4) for the human resources data set.

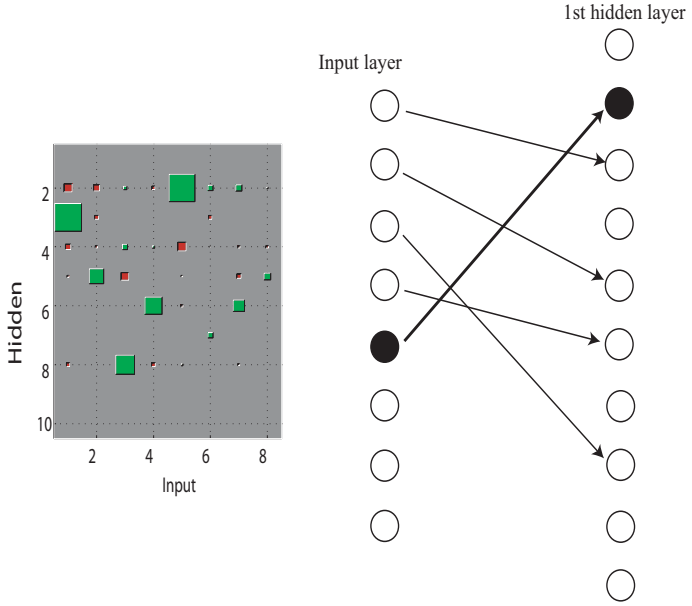

Figure 15. Connection weights to the input layer to the first hidden layer, corresponding to Figure 14(a4) for the human resources data set.

into account. Compared with the other results by the non-mutual information method in Figure 16(b) and by the regression analysis in Figure 16(c), the mutual information method could produce the most explicit interpretations, which seemed to be intuitively reasonable.

3.3.5 Generalization Comparison

Multi-layered neural networks produced better generalization performance than the logistic analysis. In particular, mutual information maximization produced the best generalization performance. Table 2 shows generalization errors for ten different hidden layers. As can be seen in the figure, the lowest errors of 0.0483, 0.0377, and 0.0549 were all obtained by mutual information maximization. One exception was the standard deviation of 0.0034, which was obtained by the non-mutual information method. Without mutual information maximization, generalization errors were larger than those by mutual information maximization for almost all hidden layers. On the other hand, the logistic analysis produced the worst errors of 0.2270, 0.2149, and 0.2509 in terms of average, minimum, and maximum values.

One of the main differences between mutual information and non-mutual information is the number of steps needed to reach the smallest validation errors. By non-mutual information, the number of steps ranged between 3.1 and 8.2 on average, while by mutual information maximization, the number of steps increased from 5.9 with one hidden layer to 13.5 with five hidden layers. As was already mentioned, with five layers, the minimum generalization error was obtained. Thus, the number of steps is directly related to improved generalization performance. We can say that, by mutual information maximization, the over-training is appropriately controlled.

4 Conclusion

The present paper aimed to propose a new type of information-theoretic method in which mutual information is supposed to be maximized before learning, or at least at the beginning of learning. We have so far developed a new information-theoretic method for simplifying information maximization [21, 22, 23, 24, 25]. Basically, information maximization is realized by decreasing the number of important neurons and connection weights. Thus, by appropriately defining and maximizing information content, only one neuron or connection weight becomes the strongest, while all the others are inactive. This method has so far produced neural networks with a small number of connection weights

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

Methods	Layers	Steps	Average	Std dev	Min	Max
Logisitc			0.2270	0.0112	0.2149	0.2509
Without	1	4.3	0.0566	0.0072	0.0451	0.0691
	2	5.6	0.0535	**0.0034**	0.0474	0.0583
	3	3.9	0.0553	0.0063	0.0451	0.0680
	4	4.6	0.0546	0.0071	0.0434	0.0646
	5	5.1	0.0553	0.0067	0.0457	0.0669
	6	4.2	0.0548	0.0046	0.0491	0.0600
	7	5.1	0.0565	0.0046	0.0474	0.0617
	8	3.1	0.0550	0.0055	0.0457	0.0640
	9	3.9	0.0555	0.0052	0.0429	0.0629
	10	8.2	0.0621	0.0090	0.0503	0.0771
With	1	5.9	0.0567	0.0057	0.0486	0.0663
	2	9.4	0.0494	0.0051	0.0411	0.0589
	3	11	0.0497	0.0061	0.0440	0.0646
	4	12	0.0491	0.0055	0.0383	0.0566
	5	13.5	**0.0483**	0.0054	0.0389	**0.0549**
	6	13.4	0.0495	0.0066	0.0394	0.0566
	7	12.2	0.0486	0.0049	0.0411	0.0583
	8	13	0.0511	0.0057	0.0429	0.0583
	9	14	0.0534	0.0091	**0.0377**	0.0657
	10	9.5	0.0619	0.0165	0.0383	0.0897

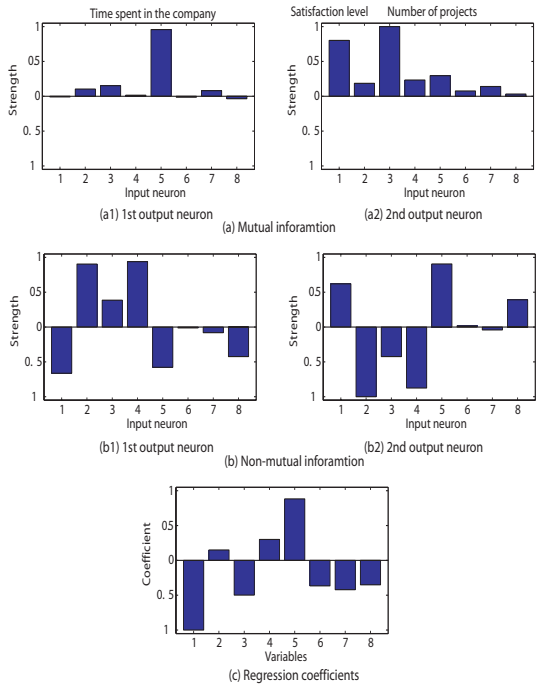

Figure 16. Collective weights into the first (1) and the second (2) output neuron by the mutual (a) and non-mutual (b) information method and regression coefficients (c) by the logistic regression analysis for the human resources data set. The correlation coefficients between connection weights to two output neurons were 0.033 and -0.9804 by the mutual information and non-mutual information methods, respectively.

or neurons. However, it has been observed that some redundant connection weights or neurons are necessary for improved generalization and interpretation. When the number of neurons and connection weights increases, some neurons or connection weights become entangled with other ones, preventing us from interpreting the final connection weights. To disentangle these components, we need to maximize mutual information. When mutual information between neurons is maximized, each neuron responds to very specific neurons, and at the same time, all neurons are used equally on average. However, it is well known that the precise computation of mutual information maximization is very expensive. Thus, we proposed here a simplified method in which mutual information was supposed to be already maximized.

The method was applied to the crab data set, the wholesale data set, and human resources data set. The experimental results show that mutual information could be increased by the present method, leading to improved generalization performance. Finally, we could interpret relations between inputs and outputs. The results confirm that the present method could extract the most important variables. In addition, generalization performance could be much better than that by the logistic regression analysis. Further research should examine the ways in which features extracted by the present method are different from those extracted by other methods.

References

[1] R. Kamimura, Mutual information maximization for improving and interpreting multi-layered neural network, in Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI) (SSCI 2017), 2017.

[2] R. Linsker, Self-organization in a perceptual network, Computer, vol. 21, no. 3, pp. 105–117, 1988.

[3] R. Linsker, How to generate ordered maps by maximizing the mutual information between input and output signals, Neural computation, vol. 1, no. 3, pp. 402–411, 1989.

[4] R. Linsker, Local synaptic learning rules suffice to maximize mutual information in a linear network, Neural Computation, vol. 4, no. 5, pp. 691–702, 1992.

[5] R. Linsker, Improved local learning rule for information maximization and related applications, Neural networks, vol. 18, no. 3, pp. 261–265, 2005.

[6] R. Battiti, Using mutual information for selecting features in supervised neural net learning, IEEE Transactions on neural networks, vol. 5, no. 4, pp. 537–550, 1994.

[7] S. Becker, Mutual information maximization: models of cortical self-organization, Network: Computation in Neural Systems, vol. 7, pp. 7–31, 1996.

[8] G. Deco, W. Finnoff, and H. Zimmermann, Unsupervised mutual information criterion for elimination of overtraining in supervised multilayer networks, Neural Computation, vol. 7, no. 1, pp. 86–107, 1995.

[9] G. Deco and D. Obradovic, An information-theoretic approach to neural computing. Springer Science & Business Media, 2012.

[10] J. C. Principe, D. Xu, and J. Fisher, Information theoretic learning, Unsupervised adaptive filtering, vol. 1, pp. 265–319, 2000.

[11] J. C. Principe, Information theoretic learning: Renyi's entropy and kernel perspectives, Springer Science & Business Media, 2010.

[12] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, Normalized mutual information feature selection, IEEE Transactions on Neural Networks, vol. 20, no. 2, pp. 189–201, 2009.

[13] P. Comon, Independent component analysis, Higher-Order Statistics, pp. 29–38, 1992.

[14] A. J. Bell and T. J. Sejnowski, The independent components of natural scenes are edge filters, Vision research, vol. 37, no. 23, pp. 3327–3338, 1997.

[15] A. Hyvärinen and E. Oja, Independent component analysis: algorithms and applications, Neural networks, vol. 13, no. 4, pp. 411–430, 2000.

[16] P. Comon, Independent component analysis: a new concept, Signal Processing, vol. 36, pp. 287–314, 1994.

[17] A. Bell and T. J. Sejnowski, An information-maximization approach to blind separation and blind deconvolution, Neural Computation, vol. 7, no. 6, pp. 1129–1159, 1995.

[18] J. Karhunen, A. Hyvarinen, R. Vigário, J. Hurri, and E. Oja, Applications of neural blind separation to signal and image processing, in Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on, vol. 1, pp. 131–134, IEEE, 1997.

[19] H. B. Barlow, Unsupervised learning, Neural computation, vol. 1, no. 3, pp. 295–311, 1989.

[20] H. B. Barlow, T. P. Kaushal, and G. J. Mitchison, Finding minimum entropy codes, Neural Computation, vol. 1, no. 3, pp. 412–423, 1989.

[21] R. Kamimura, Simple and stable internal representation by potential mutual information maximization, in International Conference on Engineering Applications of Neural Networks, pp. 309–316, Springer, 2016.

[22] R. Kamimura, Self-organizing selective potentiality learning to detect important input neurons, in Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on, pp. 1619–1626, IEEE, 2015.

[23] R. Kamimura, Collective interpretation and potential joint information maximization, in Intelligent Information Processing VIII: 9th IFIP TC 12 International Conference, IIP 2016, Melbourne, VIC, Australia, November 18-21, 2016, Proceedings 9, pp. 12–21, 2016. Springer.

[24] R. Kamimura, Repeated potentiality assimilation: simplifying learning procedures by positive, independent and indirect operation for improving generalization and interpretation, in Neural Networks (IJCNN), 2016 International Joint Conference on, pp. 803–810, IEEE, 2016.

[25] R. Kamimura, Collective mutual information maximization to unify passive and positive approaches for improving interpretation and generalization, Neural Networks, vol. 90, pp. 56–71, 2017.

[26] R. Kamimura, Direct potentiality assimilation for improving multi-layered neural networks, in Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, pp. 19–23, 2017.

[27] R. Andrews, J. Diederich, and A. B. Tickle, Survey and critique of techniques for extracting rules from trained artificial neural networks, Knowledge-based systems, vol. 8, no. 6, pp. 373–389, 1995.

[28] J. M. Benítez, J. L. Castro, and I. Requena, Are artificial neural networks black boxes?, IEEE Transactions on Neural Networks, vol. 8, no. 5, pp. 1156–1164, 1997.

[29] M. Ishikawa, Rule extraction by successive regularization, Neural Networks, vol. 13, no. 10, pp. 1171–1183, 2000.

[30] T. Q. Huynh and J. A. Reggia, Guiding hidden layer representations for improved rule extraction from neural networks, IEEE Transactions on Neural Networks, vol. 22, no. 2, pp. 264–275, 2011.

[31] B. Mak and T. Munakata, Rule extraction from expert heuristics: a comparative study of rough sets with neural network and ID3, European journal of Operational Research, vol. 136, pp. 212–229, 2002.

[32] J. Yosinski, J. Clune, T. Fuchs, and H. Lipson, Understanding neural networks through deep visualization, in In ICML Workshop on Deep Learning, Citeseer, 2015.

[33] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, Visualizing higher-layer features of a deep network, University of Montreal, vol. 1341, 2009.

[34] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Networks, vol. 61, pp. 85–117, 2015.

[35] M. G. Cardoso, Logical discriminant models, in Quantitative Modelling In Marketing And Management, pp. 223–253, World Scientific, 2013.

Ryotaro Kamimura received his B.A. and M.A. from the University of Tsukuba, Japan, and Ph.D in electrical engineering from Tokai University, Japan. He is currently a lecturer at IT Education Center. His research interests focus on the information-theoretic approaches to neural networks.