International Journal of Smart Computing and Artificial Intelligence
International Institute of Applied Informatics
2019, Vol. 3, No. 2, 20 – 39

Autoencoder-Based Multi-Step Information Augmentation for Improving Multi-Layered Neural Networks

Ryotaro Kamimura [*], Haruhiko Takeuchi [†]

Abstract

The present paper aims to propose a new type of learning method for information augmentation by increasing the number of inputs or input dimensionality with multiple steps for improving supervised learning. One of the major problems of neural networks is that multi-layered neural networks, as a property of multi-layers as an in-formation channel, principally tend to lose any information content, for example, input patterns or error gradients. For overcoming the loss of information, unsupervised pretraining was proposed, giving initial weights for the supervised learning. However, the unsupervised pretraining to train multi-layered neural networks turned out to be not so effective as had been expected, because connection weights obtained by the unsupervised learning tend to lose their original characteristics immediately in supervised training. To keep original information by unsupervised learning, we try here to increase information in input patterns as much as possible to overcome the vanishing information problem. In particular, for acquiring detailed information more appropriately, we gradually increase detailed information through multiple steps. We applied the method to the real eye-tracking data set, where the number of inputs was strictly restricted and the majority of inputs were highly correlated. When the present method of information augmentation was applied, it was confirmed that generalization performance could be improved. Then, we could interpret the importance of input variables more easily by treating all connection weights collectively. In addition, this interpretation of collective weights conformed to that of the findings by the conventional eye-tracking experiments.

Keywords: information augmentation, excessive information, autoencoder, interpretation, generalization

1 Introduction

The present paper aims to show that input information plays a crucial role in improving the performance of neural networks. In neural networks, the input neurons of an input layer have not necessarily received due attention. For example, the input node is only considered to play a role in receiving inputs from the output side without any modifications. This situation is clearly described by an expression for the input node, saying that an input node is “clamped” to an input. Thus, in conventional neural network

[*] Kumamoto Drone Technology, Kumamoto, Japan and Development Foundation and Tokai University, Kanagawa, Japan
[†] National Institute of Advanced Industrial Science and Technology (AIST), Ibaraki, Japan

research, the input layer has so far played a very passive role. However, the input layer should play the most important role in learning, because it is natural that the performance of neural networks is dependent on the input information, and we can say that any learning rules in neural networks cannot exist without input information.

Considering the importance of input information, one of the troublesome situations in neural networks research is that the researchers have tried to reduce input information as much as possible. For example, any types of sparse regularization methods in a broader sense aim to reduce the information content [1], [2], [3], [4], [5] by reducing the strength of connection units and connection weights. Those methods have been based on an idea that input information is abundant, and all we have to do is to reduce unnecessary information content. However, the input information is not necessarily abundant but is very limited, because the input information is given through a limited number of input nodes or neurons, compared with the very complex objects to be analyzed. More seriously, the regularization methods cannot reduce or eliminate unnecessary information, but we have a high possibility that unnecessary information will remain after regularization, because the unnecessary information is naturally abundant in the input information.

In addition, the input information has a natural tendency to decrease, as shown by the channel of information theory [6], [7], [8]. When a multi-layered neural network is considered an information channel, any information content should decrease through this channel or multi-layers. Thus, we can say that the fundamental property of a multi-layered neural network lies in the fact that information in inputs and error gradients naturally and rapidly decreases through going over many layers. The vanishing information of error gradients has been extensively discussed in neural networks, producing new computational methods such as unsupervised pre-training [9], [10] and new and very simplified a ctivation functions [11]. In particular, pre-training by unsupervised learning [9], [10] was reported to be effective in weakening the vanishing information problem or gradient descent, inherent to multi-layered neural networks. However, contrary to the brilliant success of convolutional neural networks with application to image processing [12], neural networks with unsupervised pre-training have not always been successful. In addition, the simplified activation functions [11] have not been necessarily effective for all situations, compared with the conventional sigmoidal or tansig functions from our experiments.

Now, let us consider the pre-training for overcoming the vanishing information problem in multi-layered neural networks. As mentioned, the pre-training triggered a new research trend on multi-layered networks, in which layer-wise unsupervised pre-training could be useful in coping with the vanishing information, in particular, information on the error gradients. However, contrary to our expectation, the pre-training has not been used in fundamental neural network research [12]. We can explain this fact by two reasons. First, information obtained by the pre-training is not useful in training supervised and main learning. This is because connection weights, obtained by the pre-training, are forced to change drastically in the fine tuning, losing the main characteristics of weights in the pre-training. Connection weights are of no use in training multi-layered neural networks or in reducing errors between targets and outputs. However, in many applications, the unsupervised autoencoders have still been used, expecting that the unsupervised pre-training can extract important features to be used in the supervised learning [13], [14], [15], [16]. Thus, we need to consider another reason why little attention has been paid to the pre-training recently.

One of the possible reasons is related to the present paper, namely, the vanishing or loss of input information. As mentioned, a multi-layered network is considered an information channel, having a natural property of losing input information. Any input information tends to decrease gradually through multiple layers. The well-known unsupervised pre-training cannot escape from this property. The unsupervised learning at a layer in the layer-wise unsupervised pre-training proceeds by using the outputs obtained from learning in the previous or precedent layer as inputs. As mentioned, any information tends to decrease: the outputs as inputs have less information than those from the previous or precedent layer. Gradually, information on inputs as well as gradients decreases as a natural property of multiple layers as an information channel.

To overcome this problem, and by paying much attention to the input information, we try to increase input information as much as possible. However, it is not so easy to increase the number of input patterns, and in addition, it is very difficult to increase the number of input variables in actual experiments. In our experiments on the eye-tracking data, discussed below, only five input variables were used to describe the data set, and in addition, four of five input variables were highly correlated with each other, leading to the problem of multi-collinearity. Though neural networks are said to be strong at dealing with this kind of problem of multi-collinearity, a smaller number of input variables causes the main difficulty for learning.

To overcome this problem, we propose here a new type of learning method to increase information content in input patterns, and then the unnecessary information can be reduced in the subsequent multi-layered neural networks. Because we do not know which information is necessary for the subsequent learning, we need to increase any information as much as possible. However, it is not so easy to increase the information, because the data set for the experiment described in this paper is supposed to be fixed. Thus, we try here to increase the number of input variables by using the autoencoders. These autoencoders can produce the overcomplete representation in which the number of dimensions of outputs is much larger than the input dimension [17], [18]. Though this research on the overcomplete representation has focused on the production of sparse representations, we use it here only for increasing the number of input variable in terms of pseudo-inputs. These pseudo-input variables can be produced by taking into account the input variables and their correlations, which can be used to produce quite similar inputs but that are slightly different from the original ones. Because much detailed information on input patterns is extracted and represented, the subsequent supervised learning can choose appropriate information from among a great number of options or candidates for supervised learning. For information reduction, we need not use special techniques in learning, because the multi-layered neural networks have the characteristic of losing information on input patterns, by its going through many different layers and neurons from the information-theoretic points of view [6], [8], [19]. Thus, it is very important to create detailed information from input patterns.

The present method is described by three main features: the complete separation of unsupervised learning from the corresponding supervised learning, excessive information acquisition, and multi-stage information augmentation. First, it is necessary to separate unsupervised learning from supervised learning completely. As mentioned, supervised and unsupervised learning have different objectives in learning, and those objectives may be contradictory to each other. Thus, connection weights obtained by unsupervised learning tend to lose their characteristics immediately in the main supervised learning. To keep original information obtained by unsupervised learning, the original information should be separately treated. One of the main characteristics of the present method is that connection weights by unsupervised learning are not given as initial weights, but the outputs from the unsupervised learning are considered the inputs to the supervised learning. This means that we transfer information by unsupervised learning to supervised learning in terms of outputs. These inputs or pseudo-inputs can give many candidates or options to be chosen by neural networks with supervised learning, and the possibility to obtain appropriate weights consequently rises. Similar attempts have been popular in machine learning for data augmentation using conventional methods and newly developed generative models [20], [21],[22], [23], [24]. However, one of the most important differences is that our method is based not on data augmentation but on dimensionality augmentation. In addition, it can be said that the data augmentation is a kind of variable reduction [25]. Thus, input information cannot be augmented by the small number of the dimensionality of input variables, even with plenty of data sets. In addition, the number of inputs is supposed to increase step by step or gradually, because we are concerned with the abrupt change in inputs, which may cause drastic changes in the final inputs, with negative effects on learning. Thus, we think that, for information augmentation, we should increase it very carefully and step by step. This means that the detailed information is created by increasing the number of neurons step by step and as carefully as possible. Contrary to the abrupt augmentation of information on inputs, the present method tries to increase information gradually, expecting that more detailed information on input patterns can be created in the process of gradual and careful information augmentation.

The paper is organized as follows. In Section 2, first, we stress how input information plays an important role in learning. Then, multi-step information augmentation is described, where the autoencoders are used to increase the number of neurons step by step to obtain excessive information. Finally, we explain how to interpret the final results by collecting all weights, namely, collective weights. In Section 3, the experimental results on the snack food selection, or eye-tracking, data set, where two-step information augmentation was used, are analyzed. In the experimental results, we point out that generalization performance was improved when the number of neurons in the augmentation component increased. In the final experimental results, two types of collective weights were obtained. Detailed examination showed that the weights with the best generalization performance conformed to the explanation presented by the eye-tracking theory. In addition, we compared the results with those obtained by the conventional method. The results showed that any conventional methods could not produce explicit results comparable to those obtained by the present method.

2 Theory and Computational Methods

2.1 Excessive Information Augmentation

Here, we explain intuitively how the present method tries to infer or estimate the important information. Neural networks have been applied to estimate an object by information through the input layer, as shown in Figure 1(b). For simplicity, we ignore the information given to the output layer in the supervised learning. The object to be analyzed is supposed to be composed of many factors, but there are only four important factors, shown in black, and moderate important factors, shown in gray, and the others are unimportant factors. In the actual experimental situation, we do not know how many factors or variables will be needed, and the inputs are usually supposed to be smaller. In the case of a large number of input variables, some information reduction methods, such as the principal component analysis, are applied for the variable reduction. The variable reduction has played a very important role in neural networks as well as other machine learning methods [26]. However, we cannot definitely say that the information through input variables represents well the target object. Thus, even if the information selection or reduction is applied, it is not guaranteed that the compressed input variables will represent the main factors of the target object.

In particular, we deal here with a case where the number of input variables is much smaller, because there is some difficulty in defining the input variables. Figure 1 shows this situation clearly, where only two input variables are prepared, and it is quite difficult to extract important factors because of too few input variables. For this situation, the present method tries to increase the number of inputs as much as possible by taking into account inputs as well as the relations between them. As shown in Figure 1(c), the information on inputs and their relations can produce a number of different factors, some of which may be important or contain, though partially, information on the real important factors. Then, the information selection and reduction method, using multi-layered networks, is applied. In this case, it is possible to find the important factors, because we have a high possibility of finding some hints or partial information regarding the important ones.

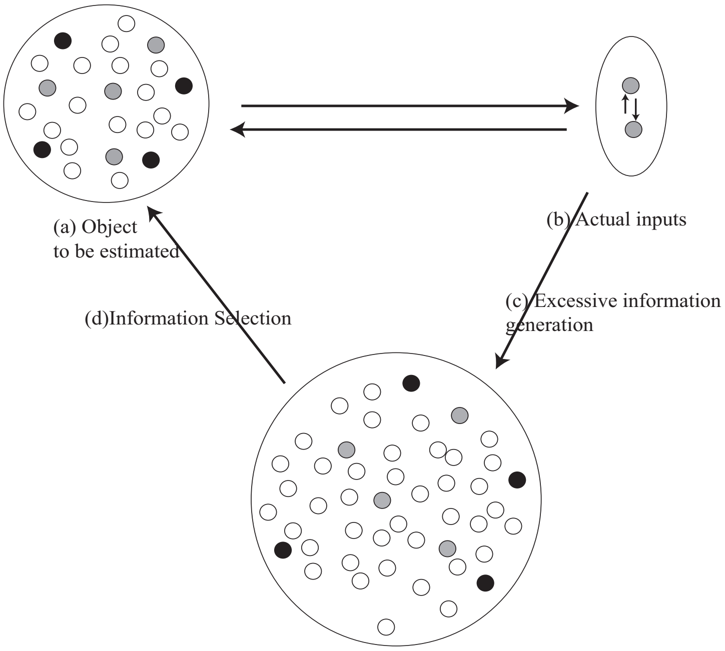
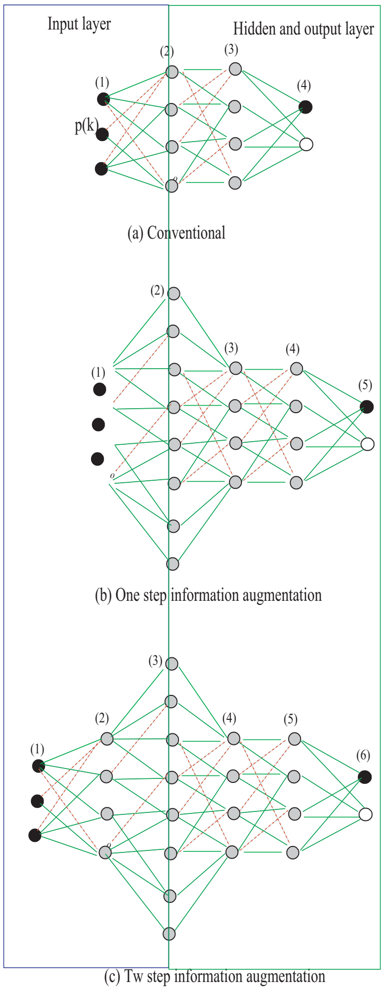


Figure 1: Information augmentation to estimate an object for learning.

An actual implementation is shown in Figure 2. Figure 2(a) shows a conventional method in which only three inputs are prepared. In Figure 2(b), the number of inputs is increased from three to eight by the autoencoder, and the inputs are used in the subsequent supervised learning. Finally, we use two-step augmentation, shown in Figure 2(c), in which the number of inputs increases from three to eight through four inputs in the second layer for a gradual and smooth transition.

2.2 Information and Dimensionality Augmentation by Autoencoders

In the dimensionality augmentation or information augmentation, we use the autoencoder as shown in Figure 3(a1). The output ${}^{s}v_j^{(2)}$ from the second layer ($j = 1, 2, ..., n_2$) for the sth



(a) Conventional

(b) One step information augmentation

(c) Tw step information augmentation

Figure 2: A network architecture without information augmentation (a) and with one-step (b) and two-step information augmentation (c).

input pattern ($s = 1,2,...,q$) is computed by

$$ {}^{s}v_j^{(2)} = \mathrm{sigmoid}\left(\sum_{k=1}^{n_1} w_{jk}^{(2)}\, {}^{s}x_k \right), \tag{1} $$

where x_k^s represents the kth element of the sth input pattern and $w_{jk}^{(2)}$ represents connection weights from the kth input ($k = 1,2,...,n_1$) to the jth hidden neuron of the second layer denoted by (2). Note that the transfer functions, such as the sigmoid function, are adopted by using the default functions in the Matlab neural network package, because we put much importance on the reproduction of the experimental results discussed below. The final output from the autoencoder can be obtained by taking the pure-linear function

$$ {}^{s}v_k^{(1)} = \sum_{j=1}^{n_2} w_{kj}^{(1)}\, {}^{s}v_j^{(2)}. \tag{2} $$

The error between outputs and inputs is

$$ E = \sum_{s=1}^{q} \sum_{k=1}^{n_1} \left({}^{s}x_k - {}^{s}v_k^{(1)} \right)^2. \tag{3} $$

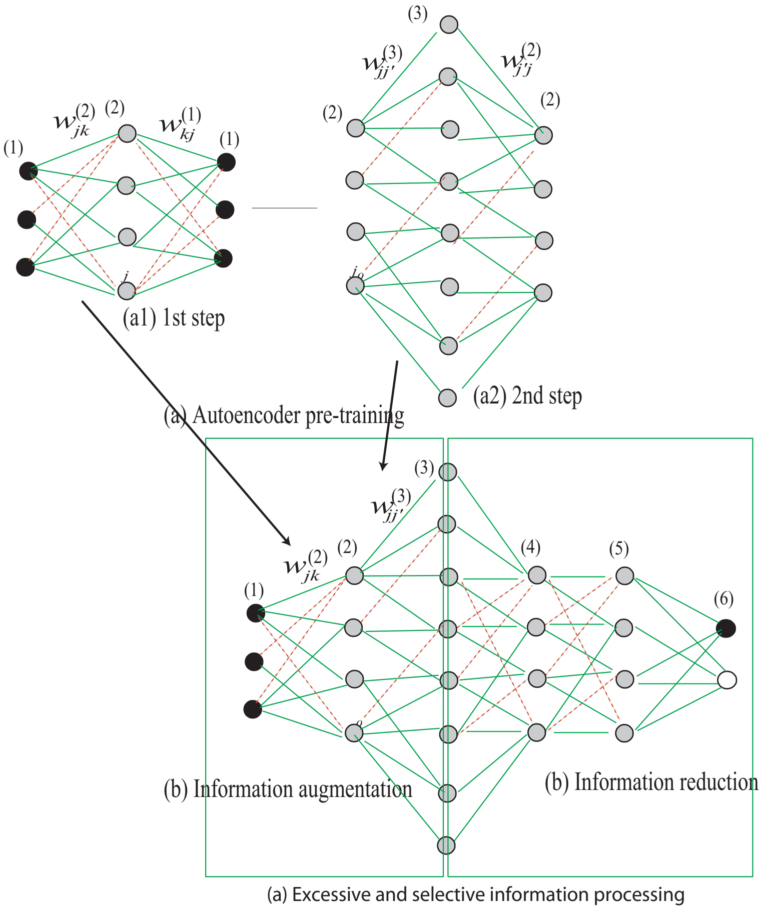


Figure 3: A network architecture with two-step information augmentation with stacked autoencoders.

In the second step, the same type of autoencoder is used, as shown in Figure 3(a2). Let $w^{(3)}_{j'j}$ denote connection weights from the second layer to the third layer; then, the output from the third layer in the autoencoder is computed by

$$v^{(3)}_{j'} = \text{sigmoid}\left(\sum_{j=1}^{n_2} w^{(3)}_{j'j}\,{}^{s}v^{(2)}_{j}\right). \tag{4}$$

The number of neurons in the third layer is obtained by multiplying the number of neurons in the second layer by a constant θ, which should be larger than or equal to zero. Thus, the number of neurons in the third layer is obtained by

$$n_3 = \theta n_2. \tag{5}$$

When the constant is zero, one layer, namely the second layer, is used for information augmentation. Then, the output is computed by

$${}^{s}v^{(2)}_{j} = \sum_{j'=1}^{n_3} w^{(2)}_{jj'} v^{s}_{j'}. \tag{6}$$

The error between the second outputs and the first outputs is obtained by

$$E = \sum_{s=1}^{q} \sum_{j=1}^{n_2} \left({}^{s}v_j^{(2)} - {}^{s}o_j^{(2)} \right)^2 . \tag{7}$$

Because the autoencoders try to augment information content on input patterns as much as possible, the regularization methods, such as weight decay, should not be included.

Finally, we should mention briefly some important information, information reduction, included in the detailed information obtained by the present method. As mentioned, multi-layered neural networks have the natural property of losing information content, meaning that, as the number of hidden layers increases, more information decreases consequently. Thus, information reduction or selection among detailed information can be simply realized by using multi-layers. By appropriately increasing the number of multi-layers, we can decrease information content. However, it is better or simpler to use more explicit regularization terms to decrease information. Here, we use the L2 weight decay for this regularization, and we try to treat the error minimization and the L2 weight decay evenly without subtle adjustment. We should stress that an increase in the number of multi-layers may be enough for losing information content.

2.3 Collective Interpretation

Multi-layered neural networks have faced difficulties in being understood from outside, because too many connection weights in many layers are entangled with each other. For this interpretation problem, we have introduced the concept of collective weights [27], [28], where complex connection weights of multi-layered neural networks are reduced to much simpler ones by treating all connection weights collectively. Then, it is possible to interpret relations between inputs and outputs by examining simplified neural networks to ones without hidden layers, as shown in Figure 4(b).

Now, collective weights between inputs and outputs are computed by summing and multiplying all weights in the input, hidden, and output layers. First, connection weights from the sixth layer to the fifth layer $w_{ij'}^{(6)}$ and connection weights from the fourth layer to the fifth layer $w_{jj'}^{(5)}$ are compressed into new weights $w_{ij}^{(6*4)}$

$$w_{ij}^{(6*5)} = \sum_{j'=1}^{n_5} w_{ij'}^{(6)} w_{j'j}^{(5)}, \tag{8}$$

where the superscript $(6 * 5)$ means that connection weights to the fifth and sixth layers are combined with each other. The same procedures are repeatedly applied, and we have the final collective weights

$$w_{ik}^{(6*1)} = \sum_{j=1}^{n_2} w_{ij}^{(6*2)} w_{jk}^{(2)}. \tag{9}$$

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

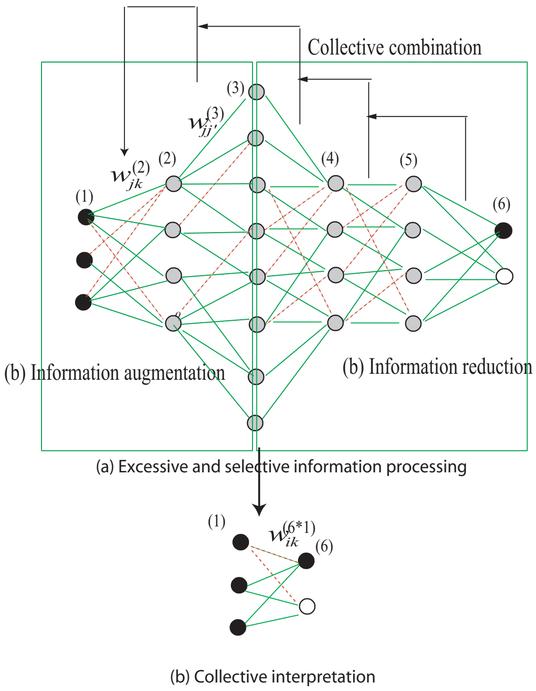

Collective combination
(3)
$w_{jj'}^{(3)}$
$w_{jk}^{(2)}$
(2)
(1)
(4)
(5)
(6)
(b) Information augmentation
(b) Information reduction
(a) Excessive and selective information processing
(1)
$w_{ik}^{(6*1)}$
(6)
(b) Collective interpretation

Figure 4: Reduction from the conventional weights (a) to the collective weights (b).

3 Results and Discussion

3.1 Snack Food Selection

3.1.1 Experimental Outline

The experiment aimed to predict which snack foods, displayed on a monitor, were chosen by the subjects via their eye-tracking records. We prepared six digital images as stimuli, each of which contained four snack food pictures. Subjects were instructed to browse a stimulus and to choose the one snack food that they wanted to buy the most, and the same task sequence was repeated for the other five stimuli. Fixation data were calculated from the eye-tracking records, relating to snack food pictures as AOI (Area of Interest). Five eye-tracking major indices were then calculated for each snack food. The eye-tracking data sets consisted of the first variable as the time for the first fixation, the second variable as a total fixation duration, the third variable as a fixation count, the fourth variable as a total visit duration, and the fifth variable as a visit count, with the subjects' decision of "chosen" or "not chosen" label.

Eye-tracking data for 22 subjects with 528 instances were used to predict snack food selection. The data set was modified by using the over-sampling method "SMOTE" to reduce the imbalance between targets in the data set [29]. Only half of all instances was for training, while the remaining half was divided equally into the validation and testing data set. We used the network architecture shown in Figure 5(b). The number of input nodes was five, and the number of neurons in the second layer increased from 5 to 100. Then, the number of neurons in the third layer was multiplied by the constant θ ranging from 1.5 to 5. Thus, the number of neurons in the third layers increased from 5θ to 100θ . Finally, collective weights were computed by multiplying and summing all connection weights, as

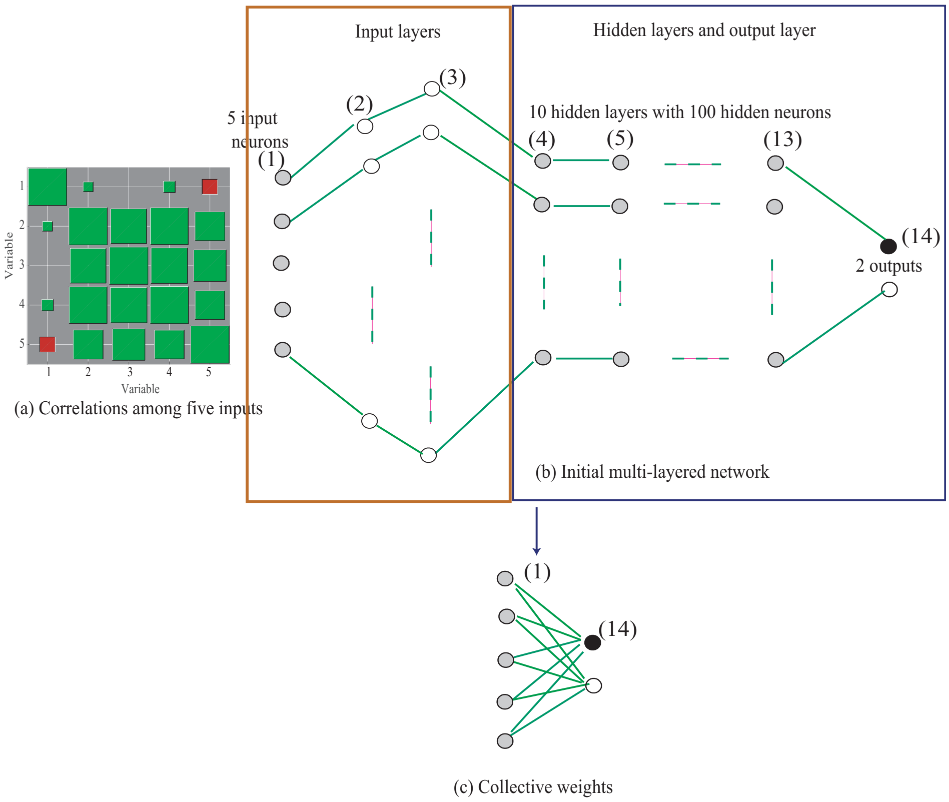

Input layers
Hidden layers and output layer
5 input neurons
(1)
(2)
(3)
10 hidden layers with 100 hidden neurons
(4)
(5)
(13)
(14)
2 outputs
Variable
1
2
3
4
5
Variable
(a) Correlations among five inputs
(b) Initial multi-layered network
(1)
(14)
(c) Collective weights

Figure 5: Correlation coefficients among five input variables (a), two-step information augmentation (b), and collective weights (c) for the snack food selection data set. The size of the squares represents the magnitude of correlations; green and red ones show positive and negative correlations, respectively.

shown in Figure 5(c).

3.1.2 Objectives of Experiments

The experiments have two basic objectives: the effectiveness of information or dimensionality augmentation and the validity of final results with respect to the eye-tracking theory. First, the effectiveness of dimensionality augmentation as information augmentation should be demonstrated. For this purpose, we used an extreme case in which almost all input variables were closely connected. Figure 5(a) shows correlation coefficients among five input variables used in the experiments, where green squares represent positive correlation coefficients, and their size shows the magnitude of the coefficients. As can be seen in the figure, four of the five input variables show very high positive correlations in large green squares, close to the maximum of one. This means that the four input variables should play almost the same role in this experiment. Thus, neural networks must actually infer the final outputs by two inputs, namely, the first and the other combined one. The conventional methods such as regression analysis cannot naturally deal with this situation, because of strong multicollinearity. Thus, if it is possible to improve the performance in terms of generalization even in this bad situation, eventually, we succeed in demonstrating the good performance of the present method. Second, the effectiveness of the present method can be further confirmed, if it is possible to explain the final results by the conventional theory of eye tracking. Then, we try to show that even if the input variables are highly correlated, the present method can detect important variables, differentiated from the correlated ones. In addition, it is shown that the importance of the obtained variables conforms to that of the eye-tracking theory.

3.1.3 Generalization Errors

Figure 6 shows generalization errors when the number of hidden neurons n_2 in the second layer in the unsupervised information augmentation component increased from 5 to 100. Correspondingly, the number of neurons in the third layer was determined by $\theta\ n_2$, where the constant increased from 1.5 to 5 by 0.5 increments.

As can be seen in Figure 6, generalization errors in terms of average (left), minimum (middle), and maximum (right) decreased gradually when the number of the second layer increased from 5 to 100 for the first hidden layer, and correspondingly the number of neurons of the third layer increased with the constant θ ranging from 1.5 (top) to 5.0 (bottom). The results suggest that the number of neurons in the second and third layer in the autoencoders increased, and correspondingly the generalization errors decreased gradually for all cases.

In particular, the minimum values, depicted in the middle of Figure 6, showed a tendency for minimum error values to decrease gradually, and their variation also became smaller, in particular, when the constant increased from 1.5 to 4.5. When the constant increased to 5 (at the bottom), the minimum values seemed to increase again. However, we could see that the lowest error of minimum values (0.0446) was obtained by the present method when the constant was 5.

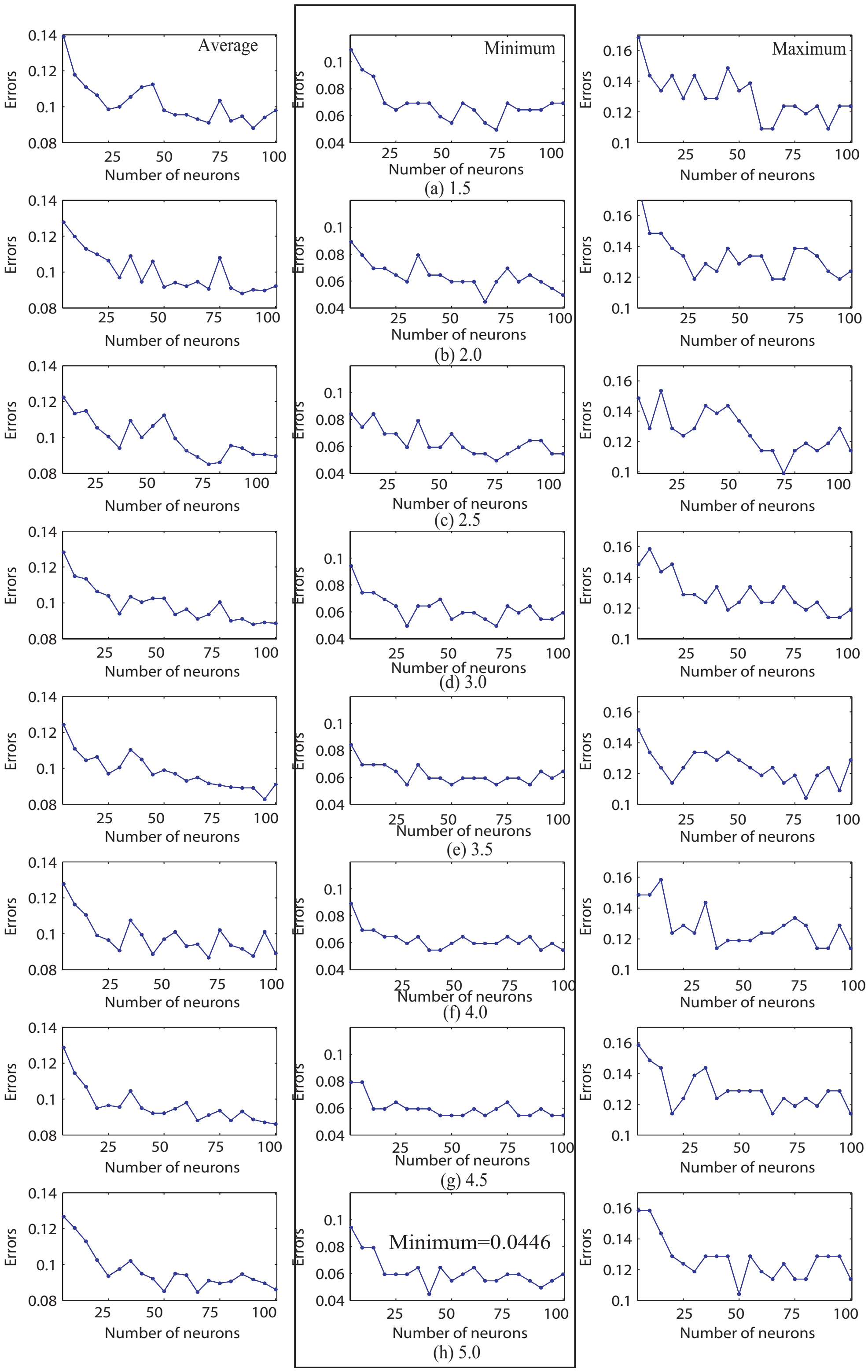
These results on generalization show that generalization performance can be improved by increasing the dimensionality of the second layer as well as the third layer. In particular, the minimum errors suggest that much better performance on generalization can be obtained when we can appropriately adjust the number of neurons in the second and third layer.

3.1.4 Generalization Comparison

Table 1 shows the summary of generalization errors by six methods. In the present method, the number of the third layer in the autoencoder was obtained by $\theta\ n_2$.

In the first place, we could see that the lower values in terms of all measures were produced by the present method with one and two hidden layers in the autoencoders. In particular, the best errors (0.0827 and 0.1040 in terms of average and maximum) were obtained by the present method with 95 neurons, and the constant θ for the third layer was 3.5, namely, 333 neurons. Thus, even for this kind of small-sized real data set, a large number of neurons was needed to improve generalization performance. The lowest minimum error of 0.0495 was obtained by the present method with two layers and with θ =2.5. The lowest maximum value of 0.1040 was obtained by the present method with two steps with 95 neurons (θ =3.5), 90 neurons (θ =1.5). With one layer in the information augmentation component, a slightly larger average error (0.0886) was obtained, but the best maximum error (0.1040) and the smallest standard deviation (0.0113) were obtained by the method with 65 neurons. In addition, for results with different hidden neurons and constants, the final errors ranged between 0.0827 (3.5) and 0.0881 (1.5, 2 and 3), which were all much smaller than those obtained by any other conventional methods. The results show explicitly that the dimensionality augmentation was related to improved generalization.

Let us examine the results by the other methods. Four-layered neural networks with



(a) 1.5

(b) 2.0

(c) 2.5

(d) 3.0

(e) 3.5

(f) 4.0

(g) 4.5

(h) 5.0

Figure 6: Generalization errors in terms of average (left), minimum (middle), and maximum (right) when the constant θ for the third layer increased from 1.5 (top) to 5 (bottom) for the snack food data set.

Table 1: Summary of experimental results by six methods on generalization performance for the eye-tracking data set. The values in the table were for the minimum average errors.

Methods	θ	2nd layer	Hidden	Avg	Std dev	Min	Max
Logistic				0.1455	0.0219	0.1089	0.1733
Bagging				0.1010	0.0190	0.0788	0.1330
BP		95	4	0.0985	0.0169	0.0792	0.1337
Autoencoder		45	4	0.1267	0.0225	0.0990	0.1634
One layer	0	65	4	0.0886	**0.0113**	0.0693	**0.1040**
Two layers	1.5	90	7	0.0881	0.0138	0.0644	**0.1040**
	2	85	5	0.0881	0.0211	0.0644	0.1188
	2.5	70	4	0.0851	0.0216	**0.0495**	0.1188
	3	90	2	0.0881	0.0119	0.0743	0.1139
	3.5	95	1	**0.0827**	0.0136	0.0594	**0.1040**
	4	70	3	0.0866	0.0223	0.0594	0.1188
	4.5	100	10	0.0861	0.0223	0.0545	0.1287
	5	65	8	0.0847	0.0150	0.0644	0.1139

BP produced the best error of 0.0985 with 95 hidden neurons except the present methods. The multi-layered neural networks with the autoencoder pre-training could not improve generalization performance, and their error (0.1267) was larger than 0.0985 obtained by the simple multi-layered neural networks without pre-training. This means that the pre-training was not effective but even harmful to improved generalization. The third worst error (0.1010) was obtained by the bagging ensemble method [30], [31], and the worst error (0.1455) was obtained by the logistic regression analysis. These results show that the two-step information augmentation method could really improve generalization performance. Therefore, dimensionality augmentation is effective for this kind of data set with a few input variables.

3.1.5 Collective Interpretation

We then try to interpret all connection weights collectively. The experimental results showed that two types of collective weights were generated by the present method. Thus, we examine here two typical examples of final collective weights obtained by the present method.

First, we present the collective weights when the best average generalization in Table 1 was obtained, and the constant θ was 3.5 with 95 neurons in the second layer. Figure 7 shows collective weights when the learning steps increased from one to 100, each of which was composed of ten learning epochs. As can be seen in Figure 7, the first input variable had the smallest absolute values for the first output (left) and the second output (right), while all the others had positive weights for the first output neuron and negative weights for the second output (right). Then, we could see that variable No.2 had the highest strength and variable No.4 had the second highest strength, and both of the variables were related to time measures. On the contrary, variables No.3 and No.5, both of which related to count measures, showed weak strength. Thus, the present method with the best generalization performance considered the input variable No.2 the most important and variable No.4 the second most important.

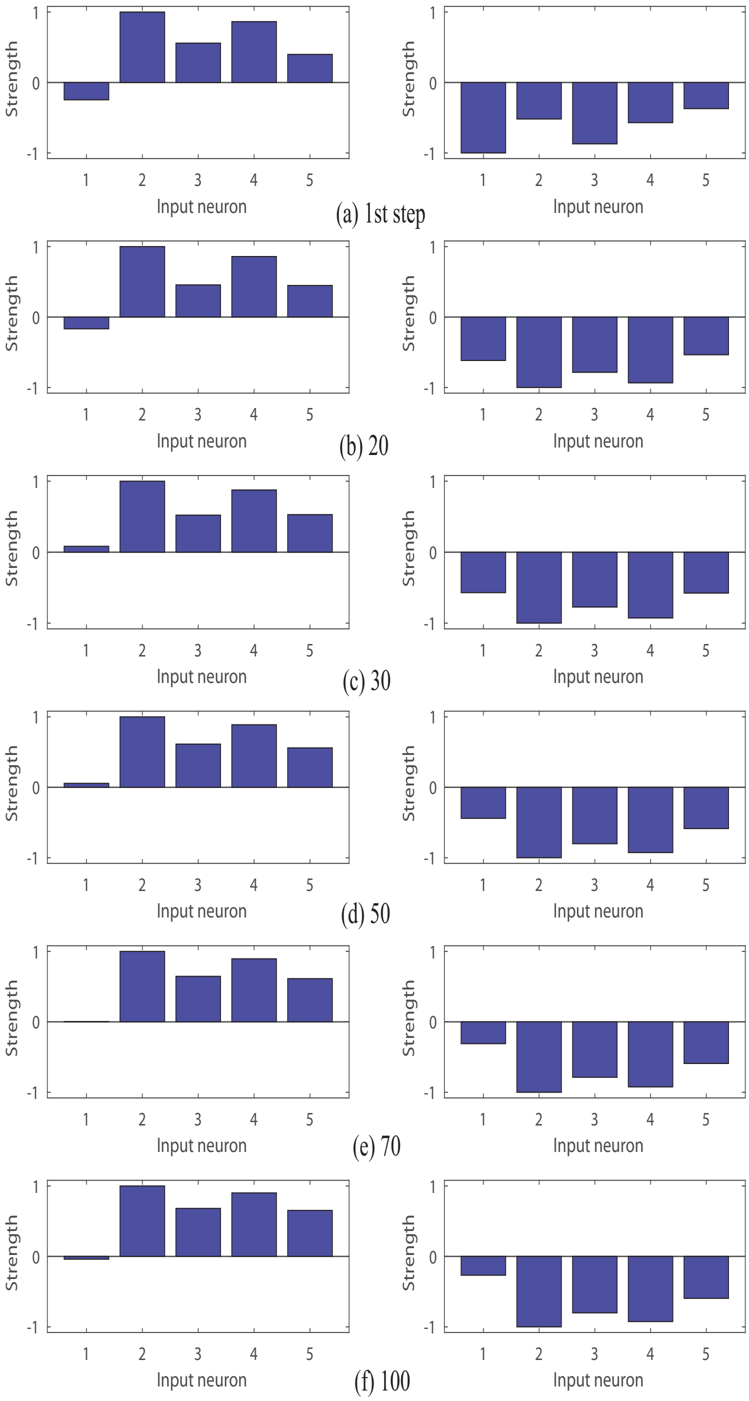
The second example was obtained when the constant was 3.0, with the worst average and minimum generalization errors among those by the present method in Figure 8. As shown in the figure, input No.3 had the highest strength for all learning steps. However, variable No.2, which showed the highest strength by the network with the best average generalization errors, had strength slightly lower than that by the second input variable. Considering these results, the best performance was due to the finding of the importance of variable No.2.

The first input variable denotes a time to the first fixation. This metric measures how long it takes before each subject fixates on the corresponding snack food picture for the first time. Although an attractive item often tends to be viewed in the earlier stage, it was shown that this metric had little relation to the consumer choice [32],[33]. The present method detected this characteristic properly, because the first input variable showed the lowest strength by the present method. The strength of the other variables differed slightly in accordance with the learning steps as they proceeded. In the eye-tracking research, the total fixation duration [32] represented by input variable No.2 and the total fixation counts of variable No.3 were considered the most important ones. As shown in the experimental results in Figure 7 and 8, input variables No.2 and No.3 showed the highest strength and importance. In addition, the present method with the best generalization performance considered input variable No.2 as the most important. We can say that, according to the results obtained by the present method, input variable No.2 was the most important and variable No.3 was the second most important. Thus, the present results show the possibility that the dimensionality or information augmentation method could extract the main important relations between inputs and outputs.

3.1.6 Interpretation Comparison

Figure 9(a) shows collective weights by the present method, with the best generalization performance. As discussed in the previous section, variable No.2 showed the highest importance, and variable No.4 showed the second highest importance. Figure 9(b) shows correlation coefficients between inputs and targets of the actual data sets. Except for the first input, all the other four inputs' correlation coefficients were almost the same, meaning that the four inputs contributed almost equally to the targets. We have shown that the present method considered variable No.2 the most important and variable No.4 the second most important. Thus, the present method succeeded in extracting important information, which was not described by the linear correlation coefficients.

The bagging method produced the similar characteristic that the first input variable had less importance in Figure 9(c), where variables No.4 and No.5 showed higher importance. However, the importance of the first variable was considerably larger, compared with the importance found by the present method. Then, the strength of predictor importance seems to increase gradually, which was quite similar to the correlation coefficients between inputs and targets in Figure 9(b). The bagging ensemble method could not produce generalization performance equivalent to that by the present method. This is because the bagging method succeeded in extracting only the relations measured by the linear correlation coefficients, while the present method could extract or detect the relations that could not be discovered by the linear correlation coefficients.



(a) 1st step

(b) 20

(c) 30

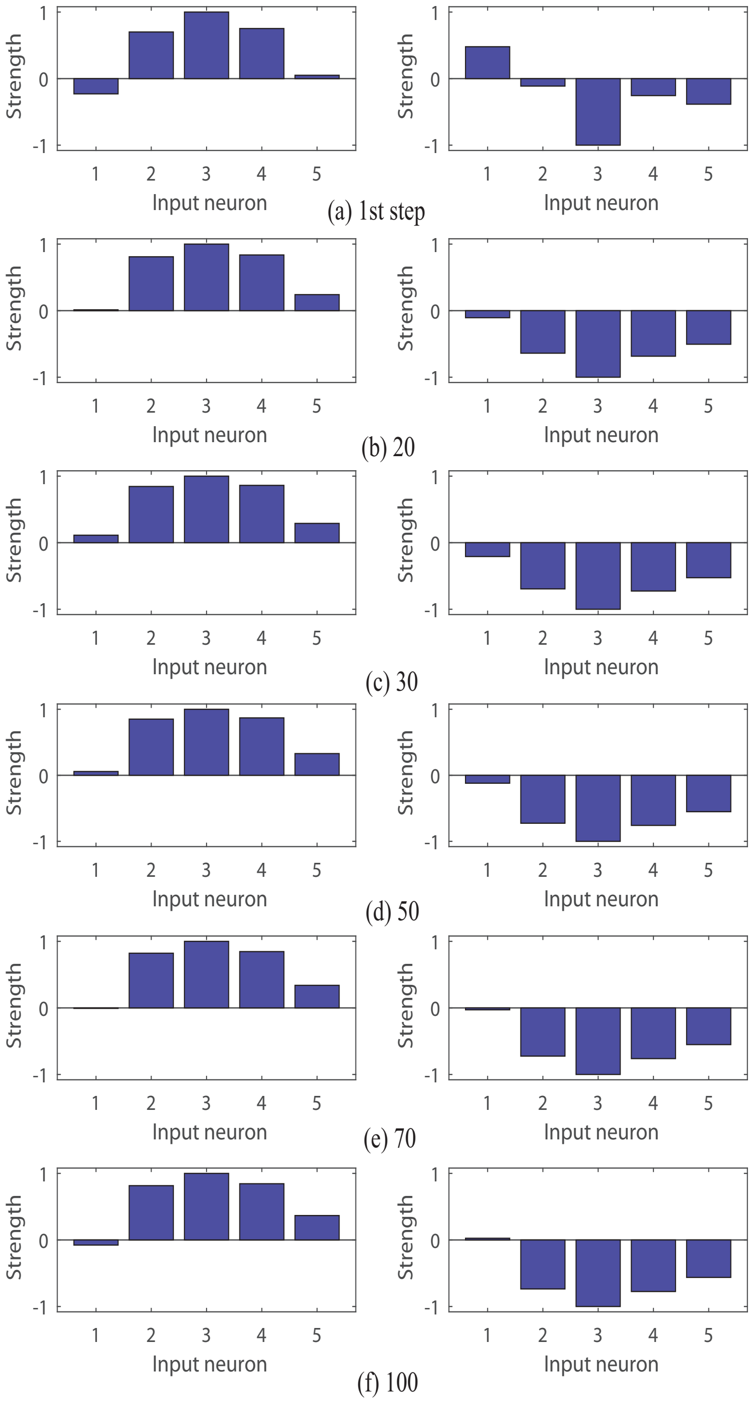
(d) 50

(e) 70

(f) 100

Figure 7: Collective weights with 95 neurons in the second layer with θ = 3.5 with the best generalization performance into the first and the second output neuron from the first learning step (top) to 100th tenth learning step (bottom) for the snack food data set.

The logistic regression produced a completely different result, in which only the fourth input kept the stronger value, while all the other inputs became almost zero in Figure 9(d). As has been well known, the regression analysis has the serious problem of multicollinearity. In the present snack food data set, all the four variables were quite similar to each other, with high correlations among them. This means that the regression analysis could



(a) 1st step

(b) 20

(c) 30

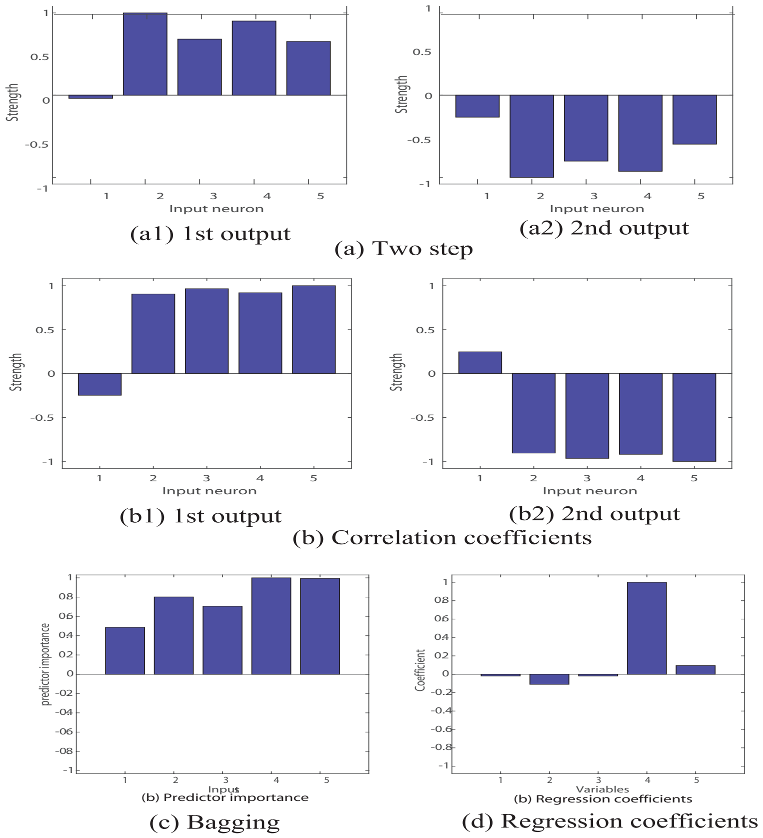
(d) 50

(e) 70

(f) 100

Figure 8: Collective weights with 90 neurons in the second layer and with $\theta = 3.0$ with the worst generalization errors into the first and the second output neuron from the first learning step (top) to the 100th learning step (bottom) for the snack food data set.

not deal with this situation because of the multicollinearity. Considering this fact on the correlation coefficients, the collective weights described in Figure 7 represent more exactly the functions of input variables.



(a1) 1st output (a2) 2nd output

(a) Two step

(b1) 1st output (b2) 2nd output

(b) Correlation coefficients

(c) Bagging (d) Regression coefficients

Figure 9: Strength of collective weights into the first and second output by the present method with the best generalization performance (a), correlation coefficients between inputs and targets (b), prediction performance by the bagging method (c), and regression coefficients by the logistic regression analysis (d) for the snack food data set.

4 Conclusion

The present paper aimed to show the importance of input information to be stored in neural networks. We stated that the input layer has not received due attention in neural network research in spite of the fact that, without input information, any learning rules cannot exist. Furthermore, input information tends to have the risk of being underestimated by the limited number of inputs or input dimensionality. For example, in the present paper, we dealt with a situation where the number of inputs, corresponding to the number of input variables, was considerably small, and the majority of input variables were highly correlated, and actually the number of inputs was even smaller. This means that, with these inputs, it became difficult to extract important information for learning because of the complexity of input patterns.

For coping with this difficult situation, we tried to increase the number of inputs or input variables by using the autoencoders. This corresponds to a situation where we try to acquire information from outside as much as possible. For living systems, it is natural that they must try to increase the information content of input patterns of complicated outer situations as much as possible to keep their existence. When detailed information or any kind of hints for the important information is obtained, it is easier to find appropriate information than without such information.

This method can be viewed as a way of information restoration for multi-layered neural networks. As has been well known, the vanishing information property, common to multi-layered neural networks, has prevented multi-layered neural networks from appropriately learning input patterns. For dealing with this vanishing information, and in particular, the error gradients, pre-training by the unsupervised learning was introduced [9], [10]. However, it has been known that the effect of pre-training has not been so effective as had been expected. This is because the objectives of unsupervised learning and supervised learning are different from each other, and weights by the unsupervised pre-training turned out to be not effective for training supervised learning. In addition, the input information tends to decrease when going through stacked autoencoders, and eventually, information by the pre-training becomes of no use for supervised learning.

The present method tried to use information obtained by the unsupervised learning in the form of inputs to multi-layered neural networks. Because multi-layered neural net-works have the property of vanishing information, we tried to increase information content obtained from input patterns as much as possible. For this purpose, the number of output neurons in the unsupervised learning was increased to extract detailed and redundant information content, because we do not know which features in input patterns are necessary to train multi-layered neural networks. Then, multi-layered neural networks tried to select necessary information from among many candidates in the form of redundant and excessive information in connection weights. This task was certainly easier, because many different options were already prepared.

The method was applied to real experimental data with only five input variables, which were highly correlated with each other. The highly correlated four input variables were combined into one variable, and then only two input variables were available. This means that too few input variables were available, making it difficult to learn relations between inputs and targets. The results show that the present method with a large number of neurons produced much better generalization performance. In particular, two steps of dimensionality augmentation could produce better generalization performance.

Finally, one of the major drawbacks of the present method is that the method is computationally expensive, because the number of input nodes must be increased as much as possible to extract redundant information content. Thus, we must compromise between the number of inputs and the corresponding costs for computation. In neurosciences, the compromise problem between the number of dimensions and the cost has already been discussed [17]. Thus, it may be quite interesting to examine the optimal ratio of the number of inputs to the corresponding cost.

For future directions, we should explore the possibility of much higher information augmentation. The present paper dealt only with one- and two-step dimensionality augmentation. However, it would be interesting to examine how many steps would be necessary or better for improving generalization. In addition, it is not enough only to increase the number of inputs for increasing information content, but it is better to increase information useful for detecting relations between inputs and targets. We think that information in input patterns should be increased, ordering the information for using it for the subsequent supervised learning. In the paper, the autoencoder seemed to be well suited for this information augmentation problem. One of our inferences on this point is that the autoencoders may have an ability to order complicated information in input patterns into more simplified forms, or more strongly to disentangle complicated information into simpler information to be easily deal with. Thus, for future direction, we need to examine more exactly what kind of information should be augmented in order to increase information effective for supervised learning. Though some problems should be solved for actual and practical data sets,the method can be applied at least to data sets with a few input variables.

References

[1] S. J. Hanson and L. Y. Pratt, "Comparing biases for minimal network construction with back-propagation," in *Advances in neural information processing systems*, pp. 177–185, 1989.

[2] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in neural information processing systems*, pp. 598–605, 1990.

[3] 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.

[4] 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.

[5] S. Srinivas and R. V. Babu, "Data-free parameter pruning for deep neural networks," *arXiv preprint arXiv:1507.06149*, 2015.

[6] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.

[7] C. E. Shannon, "Prediction and entropy of printed english," *Bell system technical journal*, vol. 30, no. 1, pp. 50–64, 1951.

[8] N. Abramson, "Information theory and coding," 1963. McGraw-Hill.

[9] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[10] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, *et al.*, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, pp. 153–160, 2007.

[11] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks.," in *Aistats*, vol. 15, p. 275, 2011.

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

[13] X. Zhang, H. Dou, T. Ju, J. Xu, and S. Zhang, "Fusing heterogeneous features from stacked sparse autoencoder for histopathological image analysis," *IEEE journal of biomedical and health informatics*, vol. 20, no. 5, pp. 1377–1383, 2016.

[14] J. Xu, L. Xiang, R. Hang, and J. Wu, "Stacked sparse autoencoder (ssae) based framework for nuclei patch classification on breast cancer histopathology," in *Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on*, pp. 999–1002, IEEE, 2014.

[15] C. Tao, H. Pan, Y. Li, and Z. Zou, "Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification," *IEEE Geoscience and remote sensing letters*, vol. 12, no. 12, pp. 2438–2442, 2015.

[16] J. Deng, Z. Zhang, E. Marchi, and B. Schuller, "Sparse autoencoder-based feature transfer learning for speech emotion recognition," in *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pp. 511–516, IEEE, 2013.

[17] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?," *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[18] H. Lee, C. Ekanadham, and A. Y. Ng, "Sparse deep belief net model for visual area v2," in *Advances in neural information processing systems*, pp. 873–880, 2008.

[19] R. Ash, *Information Theory*. John Wiley and Sons, 1965.

[20] J. Wang and L. Perez, "The effectiveness of data augmentation in image classification using deep learning," tech. rep., Technical report, 2017.

[21] A. Asperti and C. Mastronardo, "The effectiveness of data augmentation for detection of gastrointestinal diseases from endoscopical images," *arXiv preprint arXiv:1712.03689*, 2017.

[22] Y. Xu, R. Jia, L. Mou, G. Li, Y. Chen, Y. Lu, and Z. Jin, "Improved relation classification by deep recurrent neural networks with data augmentation," *arXiv preprint arXiv:1601.03651*, 2016.

[23] M. Marchesi, "Megapixel size image creation using generative adversarial networks," *arXiv preprint arXiv:1706.00082*, 2017.

[24] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[25] D. M. Allen, "The relationship between variable selection and data agumentation and a method for prediction," *Technometrics*, vol. 16, no. 1, pp. 125–127, 1974.

[26] R. May, G. Dandy, and H. Maier, "Review of input variable selection methods for artificial neural networks," *Artificial Neural Networks-Methodological Advances and Biomedical Applications, K. Suzuki, Ed. InTech*, pp. 19–44, 2011.

[27] 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.

[28] 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.

[29] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[30] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[31] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[32] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. Van de Weijer, *Eye tracking: A comprehensive guide to methods and measures*. OUP Oxford, 2011.

[33] L. N. van der Laan, I. T. Hooge, D. T. De Ridder, M. A. Viergever, and P. A. Smeets, "Do you like what you see? the role of first fixation and total fixation duration in consumer choice," *Food Quality and Preference*, vol. 39, pp. 46–55, 2015.