Proceedings of the 2007 IEEE Symposium on
Foundations of Computational Intelligence (FOCI 2007)

Information-Theoretic Variable Selection in Neural Networks

Ryotaro Kamimura, †Fumihiko Yoshida, Yamashita Toshie and ‡Ryozo Kitajima
Information Science Laboratory, Information Technology Center
ryo@cc.u-tokai.ac.jp
†Department of Media Studies
bun@f07.itscom.net
‡Graduate Program in Computer Sciences
6adrm@keyaki.cc.u-tokai.ac.jp
Tokai University, 1117 Kitakaname Hiratsuka Kanagawa 259-1292, Japan

Abstract

In this paper, we propose a new type of information-theoretic approach to variable selection. Many approaches have been proposed in estimating the importance of input variables. The majority of these approaches have focused upon output errors. We here introduce an approach concerning internal representations. First, we delete an input unit with corresponding connection weights. Then, by examining some change in hidden unit activation with and without a input variable, we can extract an important variable. We apply this method to an artificial data in which the number of hidden units is redundantly increased so as to clearly show improved performance and the stability of our method. Then, we apply the method to the cabinet approval ratings in which better interpretation of input variables can be given.

1 Introduction

In this paper, we propose a new type of information-theoretic method to extract important input variables. Feed-forward networks with back-propagation algorithm have been extensively used in real applications. However, one of the major problems is that it is difficult to interpret final internal representation due to much distributedness of the representations. To interpret easily internal representations, it is needed to develop methods to extract important features or units.

Some methods have been proposed to reduce the dimensionality or complexity of neural networks [Reed, 1993], [Mao et al., 1995], [Castellano and Fanelli, 1999]. In the majority of the methods to detect important variables or units, output errors play important roles. At this point, we think that we need to pay due attention to internal representations [Rumelhart and et al., 1986], because one of the most important features in multi-layered networks is to create internal representations, as shown in Figure 1. Thus, we here present a method to focus upon internal representations, and more concretely hidden units' activations. By examining the change in activations, we try to determine important variables. We can naturally imagine that hidden units store information content in input-output relations. Then, if it is possible to measure information content for each input variable, the information content can be used to detect the important variables.

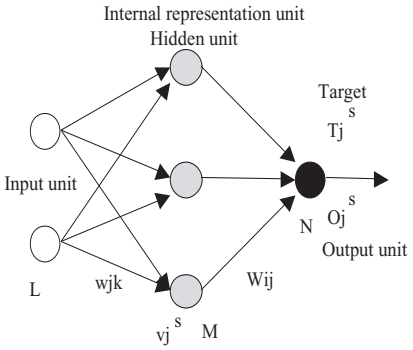

Figure 1. A network with internal representation units or hidden units.

We have so far developed an information-theoretic method to replace standard competitive learning [Kamimura et al., 2001], [Kamimura, 2003]. This method has been developed for competitive learning, but I think that the main concept, that is, information content in competitive units, can be used in the standard feed-forward network architecture. Thus, borrowing the concept of information content in competitive learning, we define information content in hidden units. This information measures how much information can be stored in hidden units with respect to given input patterns. For measuring the importance of an input unit, we delete the input unit in a network, and then compute information content. If the information content is greatly changed, the input unit surely plays an important role in learning. This information change or information loss is used in this paper to extract important variables.

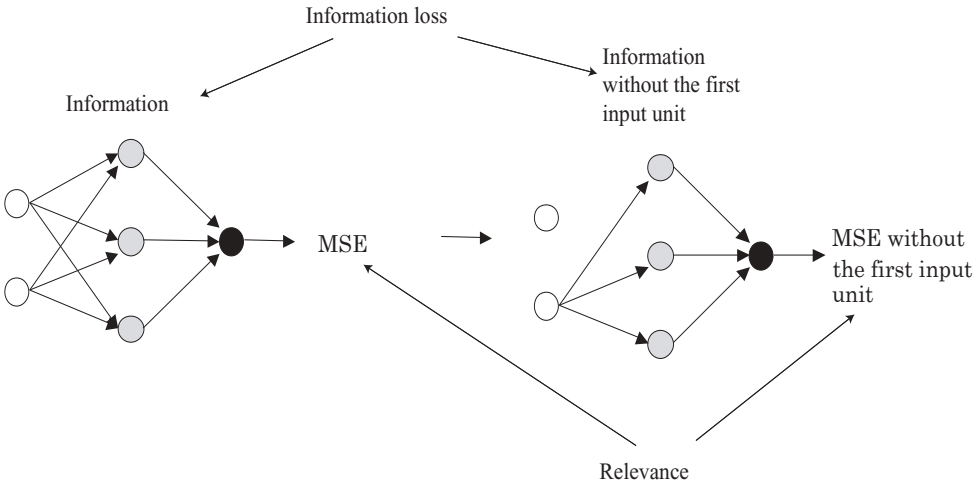

Figure 2. Difference between relevance and information loss.

2 Computational Methods

In this paper, we will introduce a new relevance measure of input units. To introduce it, we must compute the output from the jth hidden unit defined by

$$v_j^s = f(\sum_{k=1}^{L} w_{jk} x_k^s), \tag{1}$$

where w_{jk} denote connection weights from the kth input unit to the jth hidden unit and x_k^s represent the kth element for the sth input pattern. We use the sigmoid activation defined by $f(x) = 1/(1 + \exp(-x))$. The output from the output unit is computed by

$$O_i^s = \sum_{i=1}^{M} W_{ij} v_j^s, \tag{2}$$

where W_{ij} denote connection weights from the j hidden unit to the ith output unit. We must minimize the squared error defined by

$$E = \sum_{s=1}^{S} \sum_{i=1}^{S} (T_i^s - O_i^s)^2, \tag{3}$$

where T_i^s represent a target for the ith output unit for the sth input pattern.

Now, let us turn to the hidden unit activation. By normalizing this activation, we have conditional probabilities of the jth unit firing, given the sth input pattern,

$$p(j \mid s) = \frac{v_j^s}{\sum_j v_j^s}. \tag{4}$$

The probability of the jth unit firing is defined by

$$p(j) = \frac{1}{S} \sum_{s=1}^{S} p(j \mid s), \tag{5}$$

where S is the number of input patterns. By using these probabilities, information content in hidden units is defined by

$$I = -\sum_j p(j) \log p(j) + \frac{1}{S} \sum_s \sum_{j=1}^{M} p(j \mid s) \log p(j \mid s). \tag{6}$$

We now define information when a neuron is damaged by some reasons. In this case, the output without the mth unit is defined by

$$v_{jm}^s = f(\sum_{k \neq m} w_{jk} x_k^s), \tag{7}$$

where summation is over all input units except the mth unit. The normalized output is computed by

$$p^m(j \mid s) = \frac{v_{jm}^s}{\sum_{l=1}^{M} v_{lm}^s}. \tag{8}$$

Now, let us define mutual information without the mth input unit by

$$I^m = -\sum_{j=1}^{M} p^m(j) \log p^m(j) + \sum_{s=1}^{S} \sum_{j=1}^{M} p(s) p^m(j \mid s) \log p^m(j \mid s), \tag{9}$$

where p^m and $p^m(j \mid s)$ denote a probability and a conditional probability, given the sth input pattern. Information loss is defined by difference between original mutual information with full units and connections and mutual information without a unit. Thus, we have information loss for the mth input unit

$$IL^m = I - I^m. \tag{10}$$

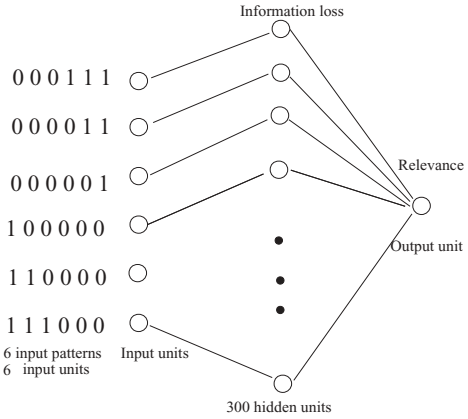

Figure 3. A network architecture for the symmetric problem.

3 Symmetric Problem

Figure 3 shows a network architecture for a symmetry data. In a network, we have six input units corresponding to the symmetry data. The output unit is one corresponding to the first part and the latter part of the data. The number of hidden units is increased to 300 units, because we can see clearer difference for a large number of hidden units.

To reduce the complexity and to improve the stability of learning, we introduce the weight decay method. We have MSE for the output errors and also the mean squared weight, that is, MSW for the weights. Thus, we must minimize the equation: $\gamma MSE+(1-\gamma)MSW$. Figure 4 shows results by three methods. Figure 4(a) shows the results by standard BP without the weight decay. In both information loss and relevance, we cannot see symmetric patterns. In addition, for different runs with different initial conditions, we have completely different patterns. Figure 4(b) shows information loss and relevance by introducing the weight decay ($\gamma = 0.5$). Though the values of information loss are small, two measures show similar patterns in which the rightmost and left most inputs represent more important features. In Figure 4(c), the information loss show clearer symmetric patterns than the relevance. This result shows that the information loss can be used to detect important features when we introduce the weight decay.

4 Approval Rating Estimation

4.1 Data Preparation by Natural Language Processing

In this study, cabinet approval ratings were estimated using a data set generated by a natural language processing system from 2371 newspaper editorials of four major newspapers – Asahi Shimbun, Mainichi Shimbun, Nihon Keizai Shimbun, and Yomiuri Shimbun. These editorials cover the period through April 27, 2001, the day of inauguration of the Koizumi cabinet, through September 26, 2004, and include at least one sentence that refer to the Koizumi cabinet. From these editorials 8585 sentences that referred to the Koizumi cabinet were extracted and subsequently analyzed by the language processing system. Using the processing system with modality recognition functions and its word count functions on a good-bad scale, each sentence was assessed in terms of forty variables. Of these variables, two of them assess the number of positive words and negative words appearing within the last two phrases of each sentence. Here, "positive word" means a Japanese word which with no doubt most Japanese speakers regard as a word with "good" connotation. "Negative word" means, of course, the one most Japanese speakers regard as a word with "bad" connotation. The remaining thirty-eight variables assess the modality pattern of each sentence, with each variable corresponding to one of thirty-eight modality patterns.

4.2 Network Architecture

Figure 5 shows a network architecture to infer approval ratings. The number of input units is 37 units, corresponding to 37 variables extracted from 40 variables above discussed. We chose 37 variables of 40 variables, because there were no records in some variables. The number of hidden units is experimentally chosen so as to give the best performance. The number of output unit is one for the scores of approval ratings. The number of input patterns is 46. The first 38 periods are considered to be a training data set, and the remaining eight periods are set to a testing data set. The 38 periods were chosen to compare our results with those by conventional regression analyses whose computation is severely restricted. In the experiments, we used a Matlab neural network package with all default values except the method of the Polak-Ribiere conjugate gradient considering learning stability.

4.3 Generalization Performance

To examine generalization performance, we used the testing data to stop training. As errors between targets and outputs are increased for testing data, learning is forced to stop even though training errors are significantly decreased. Thus, learning was forced to stop due to over-training. This situation does not really happen in actual situations. However, the method can be used to estimate a kind of potentiality of neural networks in terms of generalization performance. Experimental results showed that the coefficient was increased to over 0.7, and for an initial condition, it was well over 0.8. This shows a potentiality of networks for the inference of approval ratings.

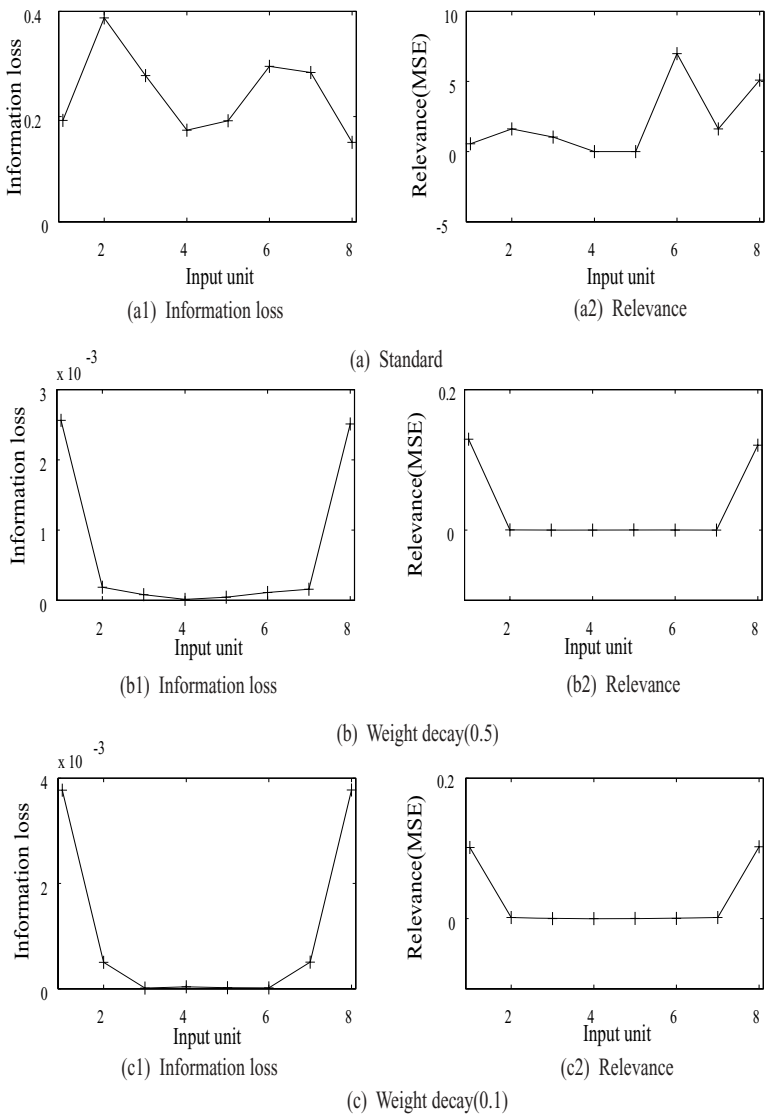

Figure 4. Information loss (left) and relevance (right) for the symmetry problem by three methods: standard BP (a), BP with the weight decay(0.5)(b) and BP with the weight decay (0.1)(c).

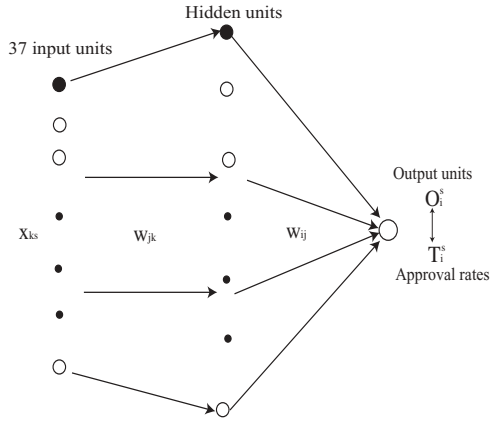

Figure 5. A network architecture for the approval-rating problem.

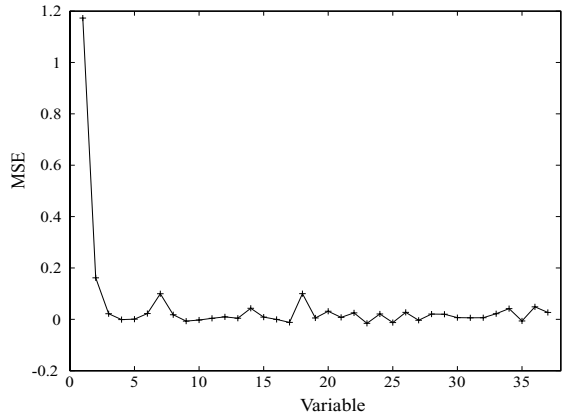

Figure 6. MSE as a function of the number of variables..

4.4 Variable Selection

Figure 6 shows the MSE without a variable. As shown in the figure, the first variable shows a large increase in MSE, meaning that the first variable, that is, the previous rating should play the most important role in decreasing MSE.

Parallel to this approach using MSE, we have used an information-theoretic measure detecting the importance of variables. Figure 7 shows information loss as a function of variables. As can be seen in the figure, the first unit shows a large loss in information. Compared with the previous results, some minor information losses are more visible. This is because information is not directly related to the MSE in the output layer.

Figure 8(a) shows actual and estimated approval ratings by the first input unit only. As can be seen in the figure, the networks well estimates approval ratings, and the correlation coefficient is 0.515 in generalization. On the other hand, Figure 8(b) shows actual and estimated ratings by the sixteenth variable (the lowest information loss). The

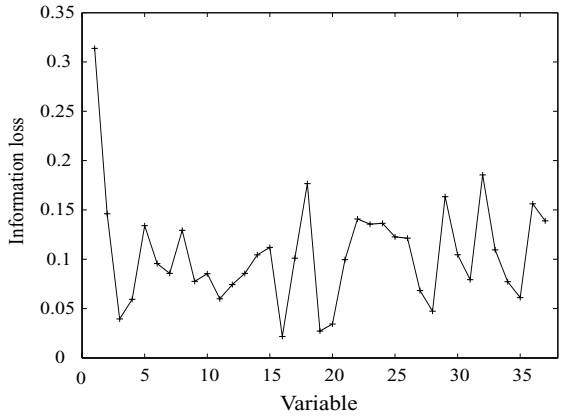

Figure 7. Information loss as a function of the number of variables.

network produces an almost flat pattern in ratings, meaning that it is impossible to estimate the ratings. These results show that the information loss well represents the importance of given variables.

4.5 Standard Regression Analysis

Finally, we used the standard regression analysis (stepwise regression) for the data. The correlation coefficient is 0.943 for the training data, and 0.678 for testing data. These results show that neural networks have much better performance than conventional regression analysis in terms of training and generalization performance. By the stepwise regression, we obtained just five variables, that is, No.1, No.2, No.6, No.20 and No.27. When we examine closely Figure 7, these variables show rather lower information loss, that is, not so important variables except the first variable. This means that important variables obtained by two methods are greatly different to each other.

5 Conclusion

In this paper, we have shown that important variables can be selected by information-theoretic measure called *information loss*. By applying it to the symmetry problem, we have shown that the information loss extract more clearly important features compared with the relevance measure. This tendency could be confirmed by the experimental results on the approval ratings. However, when we compare our results with those by the regression analysis, quite different variables were obtained. Thus, we should examine difference between two methods more explicitly for further study.

References

[Castellano and Fanelli, 1999] Castellano, G. and Fanelli, A. M. (1999). Variable selection using neural-network

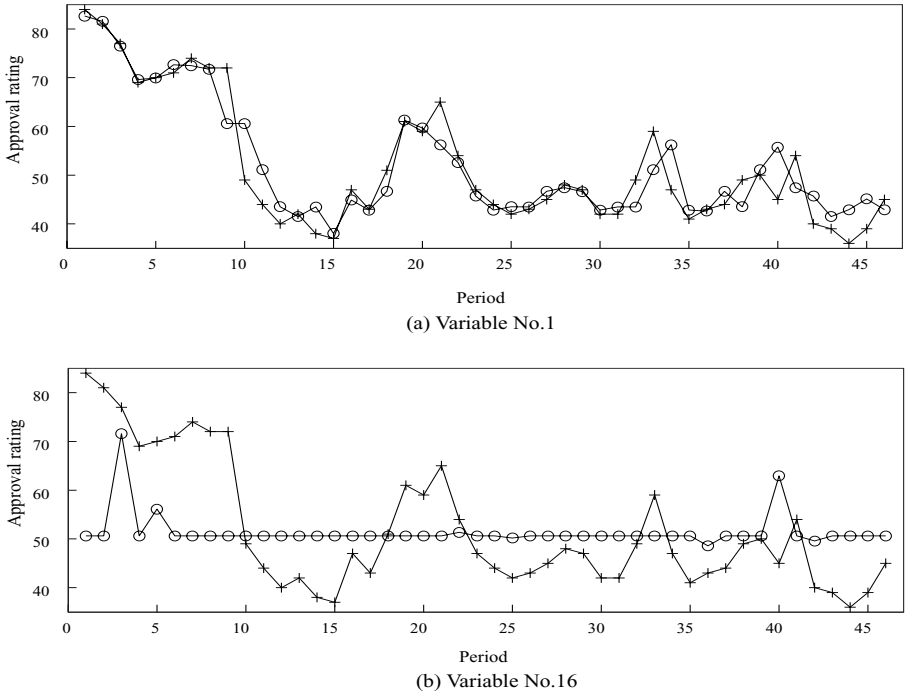

Figure 8. Actual and estimate approval ratings by one variable.

models. *Neurocomputing*, 31:1–13.

[Kamimura, 2003] Kamimura, R. (2003). Information-theoretic competitive learning with inverse euclidean distance. *Neural Processing Letters*, 18:163–184.

[Kamimura et al., 2001] Kamimura, R., Kamimura, T., and Uchida, O. (2001). Flexible feature discovery and structural information. *Connection Science*, 13(4):323–347.

[Mao et al., 1995] Mao, J., Mohiudden, K., and Jain, A. (1995). Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions on Neural Networks*, 6:296–317.

[Reed, 1993] Reed, R. (1993). Pruning algorithms-a survey. *IEEE Transactions on Neural Networks*, 5:740–747.

[Rumelhart and et al., 1986] Rumelhart, D. and et al., J. M. (1986). *Parallel Distributed Processing*, volume 1. MIT Press, MA.