Teacher-Directed Information Maximization: Supervised Information-Theoretic Competitive Learning with Gaussian Activation Functions

Ryotaro Kamimura,
Information Science Laboratory, Tokai University,
1117 Kitakaname Hiratsuka Kanagawa 259-1292, Japan

Abstract

In this paper, we propose a new method for information-theoretic competitive learning that maximizes information about input patterns as well as target patterns. The method is called *teacher-directed information maximization*, because target information directs networks to produce appropriate outputs. Target information is given in the input layer, and errors need not be back-propagated, as with conventional supervised learning methods. In the new method, we use information-theoretic competitive learning with Gaussian activation functions to simulate competition, because information maximization processes are accelerated by changing the width of the functions. Teacher information is added by distorting the distance between input patterns and connection weights. We applied our method to a road classification problem. In the problem, we could show that training errors could be significantly decreased and better generalization performance could be obtained.

1 Introduction

In this paper, we propose a new approach to supervise competitive learning. In the new method, teacher information is included in the input layer, and it directs networks to produce appropriate outputs. Because errors between targets and outputs need not be back-propagated, this is a very efficient learning method. The method can contribute to neural computing in three ways: (1) this is a new type of flexible information-theoretic competitive learning; (2) we use Gaussian functions to compute outputs from competitive units; (3) weighted distances between input patterns and connection weights are used.

First, this is a new type of competitive learning in which information is maximized to simulate competitive processes. Conventional competitive learning has been used as one of the main learning algorithms in neural networks [1], [2]. In competitive learning, a winner is obtained by the winner-take-all algorithm or more biologically plausible lateral inhibition. However, several serious problems have been reported in conventional competitive learning. For example, some neurons become dead or under-utilized. To overcome this problem, there have been many attempts to eliminate dead neurons [4], [5], [6], [7], [8], [9], [10], to cite a few. The problem still remains serious in conventional competitive learning. In our new method [11], [12], [13], [14], this problem can clearly be solved, because competitive processes are realized by maximizing information between input patterns and competitive units. In maximizing mutual information, the entropy of competitive units is increased as much as possible. When the entropy is maximized, all competitive units are equally used on average, and no dead neurons can be produced. In addition, when information is completely maximized, this method becomes close to conventional competitive learning with the winner-take-all algorithm. On the other hand, when information is smaller, many competitive units are activated, and soft competitive processes are realized. Thus, our method includes conventional competitive learning and is a very flexible competitive learning method.

Second, we use Gaussian activation functions to realize competitive processes. In the previous methods, we used the sigmoidal activation function [15], [16], [17] [11], [12], [13]. As information is increased, strongly negative connections are generated; information can easily be increased. However, too strongly negative connections may blur teacher information, and this causes difficulty in decreasing errors between targets and outputs. To remedy this shortcoming, we use Gaussian functions in this paper, because we can increase information content by adjusting the width of Gaussian functions. When the width is smaller, competitive units tend to respond to a limited number of input patterns, which should be realized by maximizing mutual information between input patterns and connection weights. Thus, information maximization can be facilitated by adjusting the Gaussian width.

Third, we try to extend our information-theoretic competitive learning to supervise learning. Because unsupervised competitive learning cannot deal with complex problems, we need to incorporate teacher information in competitive learning. For example, LVQ by Kohonen [3] is one of the most successful techniques for including teacher information in competitive learning. LVQ measures distance between input patterns and competitive units, and if an input pattern is not included in the corre-

sponding class, distance is increased. Another model is a hybrid model in which competitive learning is directly connected with supervised learning. The method is called *counter-propagation* [18], [19]. In the method, learning is significantly accelerated, compared with conventional BP. Rumelhart and Zipser [2] used correlated teachers to supervised competitive learning in which overwhelming large correlated teachers are needed. In the new approach, information on input patterns as well as on targets is maximized. To realize this situation, we add targets to the ordinary input units. Then, distance between input patterns and connection weights are adjusted by taking input account target information. For example, even if distance between input patterns and corresponding connection weights is small, distance is forced to be increased when target information tells a network to do so. In the new approach, we need not back-propagate errors between targets and outputs, and this is a very efficient computational method.

2 Teacher-Directed Information Maximization

Information is defined as decrease in uncertainty from an initial state to a state after receiving input patterns [20], [16]. This uncertainty decrease, or the information, is defined by

$$I(j \mid s) = -\sum_{\forall j} p(j) \log p(j) + \sum_{\forall s} \sum_{\forall j} p(s) p(j \mid s) \log p(j \mid s), \quad (1)$$

where $p(j)$, $p(s)$ and $p(j|s)$ denote the probability of firing of the jth unit, the sth input pattern and the conditional probability of firing of the jth unit, given the sth input pattern, respectively.

Then, we attempt to apply the information discussed above to neural networks. For simplicity, we suppose that the number of the competitive units corresponding to the output units is two and the number of the competitive units in an intermediate layer is an even number. In addition, we suppose for simplicity that input patterns can be divided evenly into two classes. These suppositions are only for simplifying the following presentation. As shown in Figure 1, a network is composed of an input, the first competitive and the second competitive layer. In each layer, information is maximized. However, some connections are fixed and do not change throughout learning. Figure 1(a) shows a network with only fixed connections. Connections from teachers to the first competitive units are fixed, and connections between two competitive layers are also fixed. Thus, connections to be updated are those from training units to the first competitive layer, as shown in Figure 1(b). In a testing phase (Figure 1(c)), no connection weights between correlated teachers and the first competitive layer exist. Thus, networks must infer the final states without teacher information.

Let us present update rules to maximize information content. As shown in Figure 1, a network is composed of L input units and M competitive units. We denote the value of the kth input, given the sth input pattern by x_k^s. For simplicity, the number of competitive units is even, the number of output unit is two and targets are binary. This means that a unit corresponding to a target class is turned on, while all the other units are off. We use weighted distance between input patterns and connection weights to incorporate information on targets. Weighted distance of the jth competitive unit, given the sth input pattern, is defined by

$$d_j^s = \phi_j^s \sum_{k=1}^{L} (x_k^s - w_{jk})^2, \quad (2)$$

where w_{jk} denote connections from the kth input unit to the jth competitive unit, and ϕ_j^s is defined by

$$\phi_j^s = \sum_{k=1}^{N} c_{jk} t_k^s, \quad (3)$$

where t_k^s are targets (supposed to have one or zero), and connection weights c_{jk} are constants. In this paper, the weights c_{j1} for correlated teachers are set to ϵ for $1 \leq j \leq M/2$ and $2-\epsilon$ for $M/2+1 \leq j \leq M$, respectively, where M is the number of competitive units and $0 < \epsilon < 1$. The weights c_{j2} have inverse values: $2-\epsilon$ for $1 \leq j \leq M/2$ and ϵ for $M/2+1 \leq j \leq M$. These connections are not updated and fixed throughout learning. The jth competitive unit receives a net input from input units, and an output from the jth competitive unit can be computed by

$$v_j^s = \exp\left(-\frac{d_j^s}{2\sigma^2}\right). \quad (4)$$

In modeling competition among units, we use normalized outputs as conditional probabilities:

$$p(j \mid s) = \frac{v_j^s}{\sum_{m=1}^{M} v_m^s}, \quad (5)$$

where M is the number of competitive units. Because input patterns are supposed to be given uniformly to networks, the probability of the jth competitive unit is computed by

$$p(j) = \frac{1}{S} \sum_{s=1}^{S} p(j \mid s), \quad (6)$$

where S is the number of input patterns. Thus, information is computed by

$$I(j \mid s) = -\sum_{j=1}^{M} p(j) \log p(j) + \frac{1}{S} \sum_{s=1}^{S} \sum_{j=1}^{M} p(j \mid s) \log p(j \mid s). \quad (7)$$

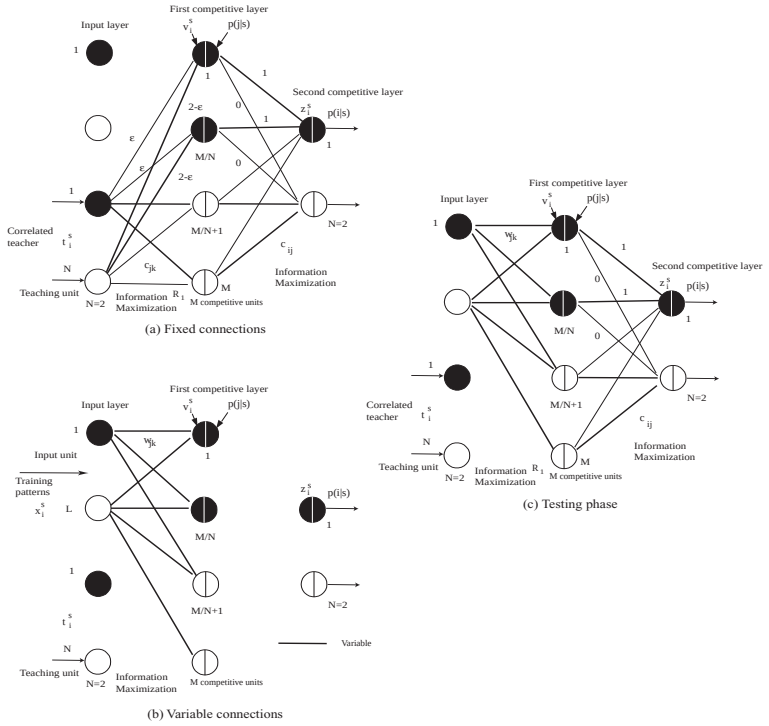


Figure 1. Multi-layered network architecture for teacher-directed learning. Figures (a) and (b) show a network with fixed connections and variable connections in a training phase, respectively. Figure (c) shows a network in a testing phase.

Differentiating information I with respect to input-competitive connections w_{jk}, we have easily the final update rules. By using this update rule, mutual information is increased as much as possible. In a process of information maximization, units compete with each other, and finally a unit wins the competition. By maximizing information, we can simulate competitive learning.

In the second competitive layer, probabilities are computed in the same way. The only difference to be noted is that input units x_k^s are replaced by normalized competitive unit activities $p(j \mid s)$. Thus, the ith competitive unit in the second competitive layer receives a net input from the first competitive layer, and an output from the ith competitive unit can be computed by

$$z_i^s = \exp\left(-\frac{\sum_{j=1}^{M}(p(j \mid s) - w_{ij})^2}{2\sigma^2}\right), \qquad (8)$$

where c_{ij} denotes a connection from the jth competitive unit in the first competitive layer to the ith competitive unit in the second competitive layer. Weights c_{ij} in the second competitive layer are some constants. For example, when the number of competitive units is two, c_{1j} are set to 1 for $1 \leq j \leq M/2$ and 0 for $M/2+1 \leq j \leq M$, respectively. Weights c_{2j} have inverse values. Conditional probabilities are computed by normalized activities:

$$p(i|s) = \frac{z_i^s}{\sum_n^N z_n^s}. \qquad (9)$$

Let us explain how teacher information directs a network to produce appropriate outputs. As shown in Figure 1(a), we imagine a case where the first competitive unit of the correlated teacher units and the output units are on. By the equation $\phi_j^s = \sum_{k=1}^{N} c_{jk} t_k^s$, ϕ_j^s are set to ϵ. Thus, connection weights from the unit to the first two competitive units in the first competitive layer are set to ϵ, which is smaller than one by definition. Thus, distances between the connection weights and these competitive units are forced to be smaller than actual distances obtained by the equation $\sum_{k=1}^{L}(x_k^s - w_{jk})^2$. This means that even if actual distance is large weighted distance becomes small, and this forces two competitive units to turn on. On the other hand, connection weights c_{jk} for the third and the fourth competitive units are set to $2-\epsilon$. This means that even if actual distance is smaller weighted distance is larger, and the competitive units tend to be off. In a testing phase (Figure 1(c)), correlated teachers are all dropped off, and a network must infer final outputs only with input patterns.

Finally, we should note a computational method. We have stated that connection weights from the correlated teachers to competitive units are always fixed. However, we have observed that if fixed connection weights are changed according to information content obtained in the course of learning, more stable and more accelerated learning can be obtained. We set

$$\epsilon = \left(\frac{I}{\log M}\right)^q, \qquad (10)$$

where $\log M$ is maximum information, and q is the parameter with $0 < q < 1$. This equation means that when information content is larger ϵ becomes one. As the parameter q becomes smaller, it becomes one. The parameter q should be usually very low, because when the parameter q is lower, teacher information is not distorted as much[1].

3 Road Classification

We present here experimental results on a road classification problem. In this problem, networks must infer whether a driver drives on a local road or an urban road. This problem aims to make driving as safe as possible. In the experiment, we prepared 45 road photographs taken from the drivers' viewpoint. Figure 2 shows two examples of photos from the total of 45 photos. Of the 45 photos, 22 photos are classified as local roads that are relatively narrow, as an example shown in Figure 2(a) illustrates. On the other hand, the remaining 23 photos are those of relatively wide urban roads, as a sample shown in Figure 2(b) illustrates. In the experiments, we set the parameter q to 0.1, and learning was considered to be finished when relative information increase was less than 0.001 for three consecutive epochs. We reduced the size of these photos to 900 (30×30) pixels to facilitate learning. Thus, we must have 900 input units. The number of competitive units was set to four to give the best performance. We used five-fold cross validation to evaluate generalization performance.

As shown in Figure 3(a), information tends to be increased as the Gaussian width is increased up to about two. Then, information is rapidly decreased as the Gaussian width is increased from that point. Figure 3(b) shows training errors in a solid line and generalization errors in a dotted line as a function of the Gaussian width σ. Training errors are decreased as the width is increased up to about two. Then, from that point, training errors are gradually increased. For generalization errors, the same tendency can be seen. However, generalization errors move behind training errors, and generalization errors fluctuate greatly. The generalization error reaches the lowest level of 0.2 when σ is 3 and 3.1, and generalization errors increase rapidly from the point. Finally, we compared generalization errors obtained by our method with those obtained by three conventional methods: PNN (probabilistic networks), BP (back-propagation) and LVQ (learning vector quantization No.1)[2], as shown in Figure 3(c). The numbers of competitive units or hidden units of TDI, BP and LVQ were set to six, four and four, respectively. These numbers was determined to give the best performance in terms of generalization errors. Finally, we compared generalization errors obtained by our method with those obtained by the three conventional methods. The numbers of competitive units or hidden units of TDI, BP and LVQ were set to four, two and two, respectively, which were determined to give the best performance. As shown in the figure, the error rate obtained by PNN is 0.444, and the rate obtained by BP is 0.36. The rate obtained by LVQ is 0.33. TDIM gives the lowest error rate, of 0.22. Experimental results show that the teacher-directed information maximization can give the best performance in terms of generalization performance.

[1] In the following experiments, the parameter q was set to 0.1 for the first approximation.

[2] We used the Matlab neural networks package for PNN, BP and LVQ for easy comparison.

4 Conclusion

In this paper, we have proposed a new type of supervised learning based upon information-theoretic competitive learning. Teacher information is realized by using weighted distance between input patterns and connection weights. By using teacher information in the input layer, distance between input patterns and connection weights is changed. For example, when distance is large, and teacher information suggests smaller distance, the distance is made smaller by correlated teachers. In competition, we use Gaussian functions of the distance. When weighted distance is smaller, competitive units tend to strongly fire. In this way, we can incorporate teacher information in the input layer, and we do not have to back-propagate error information between targets and actual outputs. Thus, this is a very efficient supervised learning method. We have applied the method to a road classification problem. In the problem, we have shown that our method shows the lowest level of generalization errors.

Finally, we should mention several problems with our method. First, we changed connection weights from correlated teachers to competitive units according to information content obtained in learning. However, more subtle adaptation of the connection weights may improve network performance. Second, Gaussian width plays a very important role in information maximization. Thus, more theoretical treatment of Gaussian width will be necessary. Third, we experimentally determined relations between information and generalization. However, more theoretical study on the relations will be helpful to improve generalization performance. Though some problems remain unsolved for this method, it is certain that the new method opens a new perspective in neural computing.

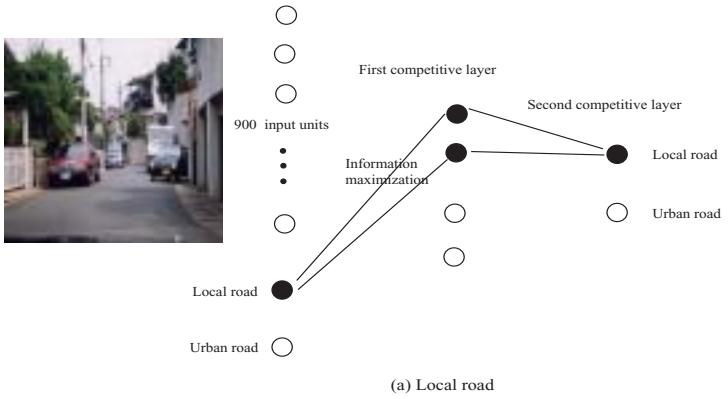
References

[1] S. Grossberg, "Competitive learning: From interactive activation to adaptive resonance," *Cognitive Science*, vol. 11, pp. 23–63, 1987.

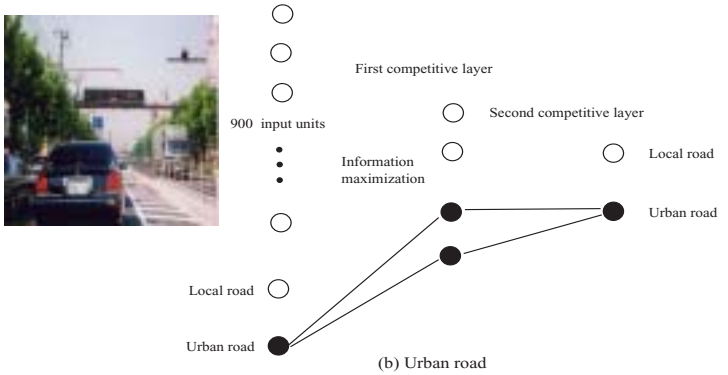
[2] D. E. Rumelhart and D. Zipser, "Feature discovery by competitive learning," in *Parallel Distributed Processing* (D. E. Rumelhart and G. E. H. et al., eds.), vol. 1, pp. 151–193, Cambridge: MIT Press, 1986.

[3] T. Kohonen, *Self-Organizing Maps*. Springer-Verlag, 1995.

[4] D. E. Rumelhart and J. L. McClelland, "On learning the past tenses of English verbs," in *Parallel Distributed Processing* (D. E. Rumelhart, G. E. Hinton,

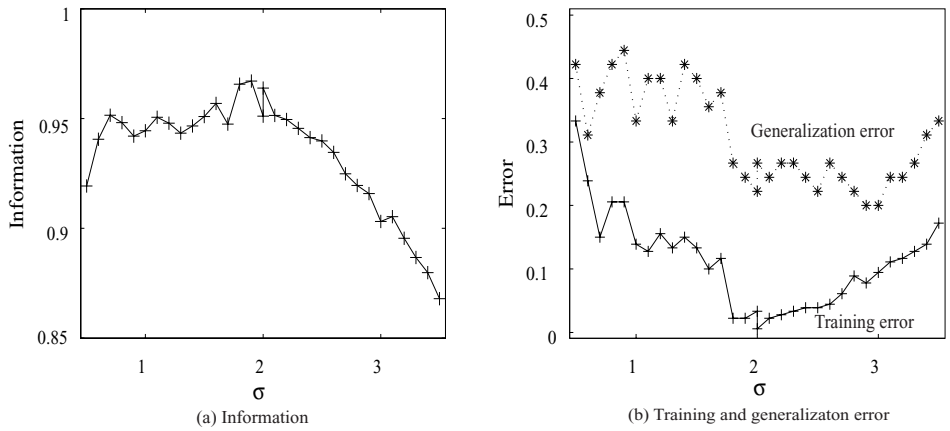


(a) Local road



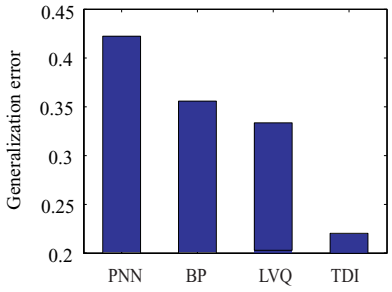
(b) Urban road

Figure 2. A network architecture for the road classification problem. Black and white circles represent fired neurons for the input and the second competitive layer.



(a) Information

(b) Training and generalizaton error



(c) Generalization comparison

Figure 3. Information (a), training and generalization errors (b) as a function of the number of epochs and generalization comparison (c).

and R. J. Williams, eds.), vol. 2, pp. 216–271, Cambrige: MIT Press, 1986.

[5] S. Grossberg, "Competitive learning: from interactive activation to adaptive resonance," *Cognitive Science*, vol. 11, pp. 23–63, 1987.

[6] D. DeSieno, "Adding a conscience to competitive learning," in *Proceedings of IEEE International Conference on Neural Networks*, (San Diego), pp. 117–124, IEEE, 1988.

[7] S. C. Ahalt, A. K. Krishnamurthy, P. Chen, and D. E. Melton, "Competitive learning algorithms for vector quantization," *Neural Networks*, vol. 3, pp. 277–290, 1990.

[8] L. Xu, "Rival penalized competitive learning for clustering analysis, RBF net, and curve detection," *IEEE Transaction on Neural Networks*, vol. 4, no. 4, pp. 636–649, 1993.

[9] A. Luk and S. Lien, "Properties of the generalized lotto-type competitive learning," in *Proceedings of International conference on neural information processing*, (San Mateo: CA), pp. 1180–1185, Morgan Kaufmann Publishers, 2000.

[10] M. M. V. Hulle, "The formation of topographic maps that maximize the average mutual information of the output responses to noiseless input signals," *Neural Computation*, vol. 9, no. 3, pp. 595–606, 1997.

[11] R. Kamimura, T. Kamimura, and T. R. Shultz, "Information theoretic competitive learning and linguistic rule acquistion," *Transactions of the Japanese Society for Artificial Intelligence*, vol. 16, no. 2, pp. 287–298, 2001.

[12] R. Kamimura, T. Kamimura, and O. Uchida, "Flexible feature discovery and structural information," *Connection Science*, vol. 13, no. 4, pp. 323–347, 2001.

[13] R. Kamimura, T. Kamimura, and H. Takeuchi, "Greedy information acquisition algorithm: A new information theoretic approach to dynamic information acquisition in neural networks," *Connection Science*, vol. 14, no. 2, pp. 137–162, 2002.

[14] R. Kamimura, "Progressive feature extraction by greedy network-growing algorithm," *Complex Systems*, vol. 14, no. 2, pp. 127–153, 2003.

[15] R. Kamimura and S. Nakanishi, "Improving generalization performance by information minimization," *IEICE Transactions on Information and Systems*, vol. E78-D, no. 2, pp. 163–173, 1995.

[16] R. Kamimura and S. Nakanishi, "Hidden information maximization for feature detection and rule discovery," *Network*, vol. 6, pp. 577–622, 1995.

[17] R. Kamimura, "Minimizing α-information for generalization and interpretation," *Algorithmica*, vol. 22, pp. 173–197, 1998.

[18] R. Hecht-Nielsen, "Counterpropagation networks," *Applied Optics*, vol. 26, pp. 4979–4984, 1987.

[19] R. Hecht-Nielsen, "Applications of counterpropagation networks," *Neural Networks*, vol. 1, no. 2, pp. 131–139, 1988.

[20] L. L. Gatlin, *Information Theory and Living Systems*. Columbia University Press, 1972.