A multi-stage counterpropagation neural network applied to pattern analysis

著者	Nakayama Kenji
journal or publication title	Proceeding of ICANN'94, Sorrento, Italy
page range	811-814
year	1994-05-01
URL	http://hdl.handle.net/2297/18390

A MULTI- STAGE COUNTERPROPAGATION NEURAL NETWORK APPLIED TO PATTERN ANALYSIS.

Y.TOMIKAWA*, K. NAKAYAMA**
*Research & Development Dept.,Yoshida Kogyo K.K. (YKK)
200, Yoshida, Kurobe, 938 Japan
**Dept. of Electrical and Computer Eng.,Faculty of Tech., Kanazawa Univ.
2-40-20, Kodatsuno, Kanazawa, 920 Japan

1 INTRODUCTION

Recently, the productive lines in factories are automated and the number of workers are reduced. These situations require the automation for checking products and detecting machine troubles. These automations however, are difficult. Already many kinds of devices for this purpose have been developed and used in our factory. However, many of them are designed to detect the characteristics proper to specified objects, so that they can not applied to other objects. One of the major subjects in our factory is development of a device, which can find many types of error shapes in zippers. This is also a main subject of this paper. On the other hand, neural network have possibility of learning ,which may be applied for the above purpose.

We first applied BP(back propagation) model to finding the error shapes of zippers. It could recognize the error shapes with about 90% recognition ratio, However the results of judgment was easily affected by position of error shapes. Furthermore, the convergence was slow, and so many neurons were needed[5]. In the BP model, the judgment is thought to be affected strongly by the whole shapes of objects. However in zippers error shapes frequently appear at the partial part and the position of them change in each sample. Then we have developed the recognition model which focuses especially on extracting the partial error shapes. This model will be described using one dimensional data. However, it can deal with the spatial shape data by transforming them to curvature data along the outline.

2 NETWORK STRUCTURE

The proposed model consists of mainly three layers, input layer, competitive layer and output layer such as the counter propagation model[1]. The model is shown in Fig.1. This model has two stages to deal with the whole shapes and the partial shapes of input data. To Stage.1, input data are directly presented and the whole of the input data connect to the sub-neurons in the competitive layer so that the whole shapes are recognized. The data gained by removing the components of the winner neuron in Stage.1 from the input data are inputted to Stage.2 and the connections of sub-neurons are partial to them so that the partial shapes are recognized.

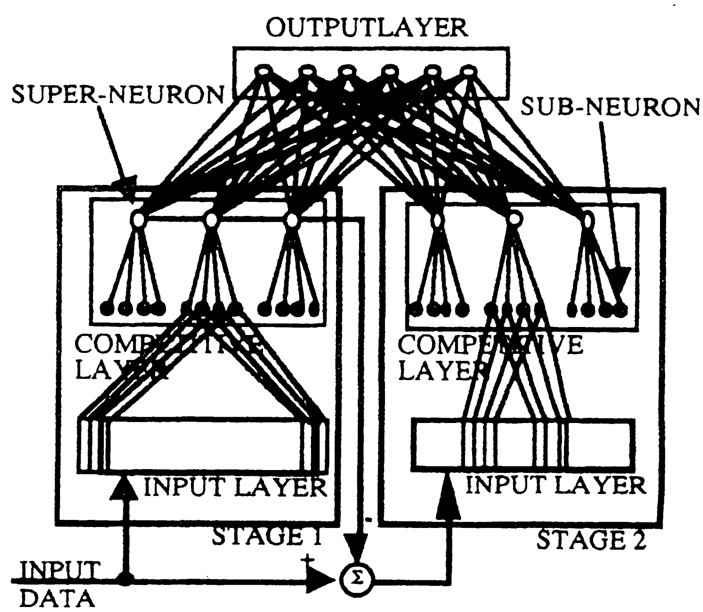


Fig.1 2 stages structure model to recognize a partial feature

The sub-neurons connected to a same super-neuron have the same components but the sifted connections to the input layer. The activation value o each super-neuron takes the maximum value of those of the sub-neurons connectec to it. The idea of these sub-neuron and super-neuron is derived from the ideas of the overlapping receptive field of T-C Problem[2] and of S-neuron and C-neuron in Neocognitron[3]. The components of the super-neuron are self-organized by the competitive learning, which are described in the next section. In this paper, we use 'components' instead of 'weights' for the template pattern formed in super-neuron in order to avoid confusing with the accumulated weight which is described later.

The neurons in the output layer are connected with all super-neurons. Their weights are adjusted by the delta rule, that is the supervised learning. The output neuron is assigned to one of whole shapes and partial shapes. The weights are learned to react to each independently .

3 LEARNING METHOD IN COMPETITIVE LAYERS

For the calculation of the activation of neurons, the inner-product of their inputs and weights is usually used. However, when input data include a partial feature, the inner-product cannot ignore the partial data so that it may not make the neuron active as though most of the weights of it coincide with the input data. Besides, because the classification of usual neural network is strongly depend on the initial state of neurons, it needs a lot of neurons in the medium layer to deal with an unpredictable feature. Then in the proposed model, we introduce an accumulated weight $wa_i(t)$ which indicates the amount of learning and improve the calculation of the output o_{ij} of each sub-neuron as follows;

$$o_{ij}=s(wa_i(t)(\sum_k g((x_{jk}-m_{ik}(t))/\sigma)/N-\eta)),\ 0<\eta<1. \tag{1}$$

where t is an index for updating and s() means a sigmoid function and g() means a Gaussian function($g(x)=\exp(-x^2)$). The suffix i,j and k correspond to the i-th super-neuron, the j-th sub-neuron belong to the i-th super-neuron and the k-th datum of input data or components, respectively. x_{jk} is the normalized input datum to the j-th sub-neuron. $m_{ik}(t)$ is the memorized component in the i-th super-neuron. σ is variance which controls the permitted degree of coincidence with x_{jk} and m_{ik}. N is the number of input data. η is the constant.

The output O_i of the i-th super-neuron is the maximum value of the sub-neurons belong to it. It is calculated by

$$O_i=\max_j(o_{ij}). \tag{2}$$

In a competitive layer, the components of the super-neuron are updated by the competitive learning. If the i'-th super-neuron is the winner in the competitive layer and the j'-th sub-neuron takes the maximum value, the update rule of the components of the super-neuron is written by

$$m_{i'k}(t+1)=(wa_{i'}(t)m_{i'k}(t)+dw_{i'}x_{j'k})/(wa_{i'}(t)+dw_{i'}), \tag{3}$$
$$wa_{i'}(t+1)=wa_{i'}(t)+dw_{i'}. \tag{4}$$

The initial state of $wa_i(t)$ is set to 0. These equations mean that the components $m_{i'k}$ become the average value of $x_{j'k}$ which have been updated $m_{i'k}$. dw_i is the increment of $wa_i(t)$, determined using the learning constant κ as follows;

$$dw_i=\kappa(1-wa_i(t)), \tag{5}$$

From this equation, $dw_i=0$ for $wa_i(t)=1$. Thus, $wa_i(t)$ will be saturated at 1.

4 BEHAVIOR OF MULTI-STAGES NETWORK

The whole shapes and the partial shapes in input data are learned in Stage.1 and Stage 2. When the first input data are presented to the network, they are transferred to Stage 1 at first. After that, the activation values of the super-neurons in Stage.1 are calculated by Eq.s.(1) and (2). At initial state, since the accumulated weights of all the super-neurons are 0, the output of them become all 0.5 and the winner can not determined. Then an arbitrary set of a super-neuron and a sub-neuron is selected and updated according to Eq.s.(3) and (4). In this case, Stage 2 is not used and the next data are presented to the network.

When the next input data are presented, the data are transferred to Stage.1 as same as the previous data. If the data are similar to the previous data, the output of the super-neuron which has been updated previously become larger than 0.5 and those of the other super-neurons remain 0.5, so that the updated super-neuron becomes the winner and it is updated. Whereas, if the data are different from the previous data, the output of the updated super-neuron becomes less than 0.5, so that the winner is selected from the other super-neurons and updated.

While the activation values of the super-neurons in Stage 1 are small, learning progresses only in Stage.1. When an unknown pattern is presented, the fresh super-neuron is assigned to the data. Besides, when a familiar pattern is presented, the trained super-neuron is adjusted to the average of patterns which have made it the winner. Since the average operation in the updating rule makes random components decrease, the partial components of a super-neuron disappear and only the whole components remain. As this result, in Stage.1, the super-neurons become to detect whole shapes.

If the output of the winner neuron becomes more than a certain threshold, the components of the winner are removed from the input data and the rest components are sent to Stage.2. Since the components of the winner in Stage.1 are of an whole shape, the rest components are of an partial shape. In Stage.2, the same learning as Stage 1 is done and the super-neurons learn partial shapes.

The super-neuron in both stages can learn a pattern independently of its initial state, so that the network does not need extra neurons to response unpredictable input data. Therefore, the proposed network can learn various patterns with less time and less neurons.

5 SIMULATION

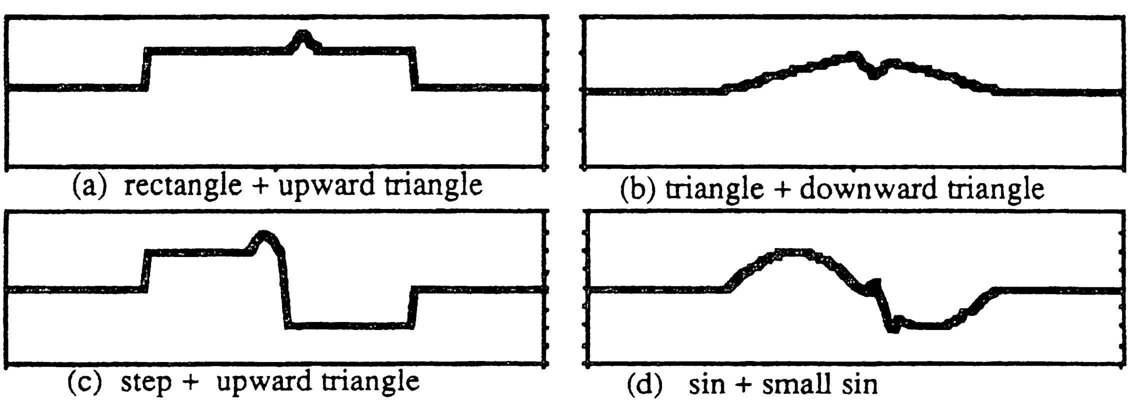


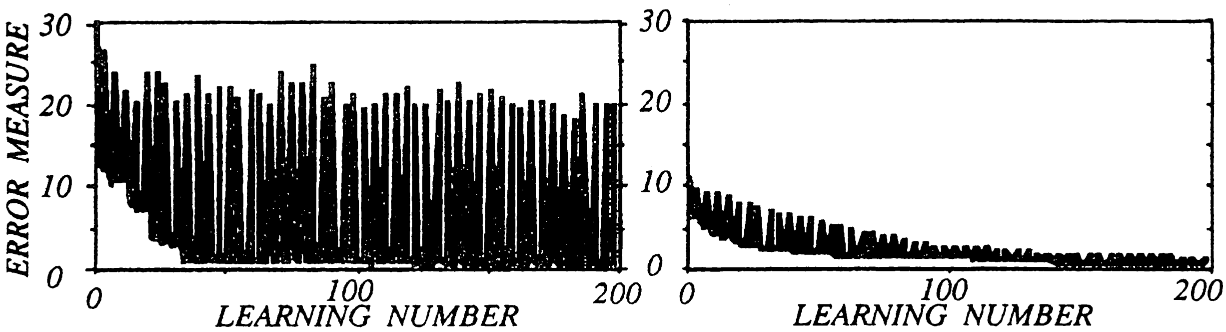
Fig.2 Examples of input data

The proposed model was tested by computer simulation using some sets of input data. The input data are composed of one of 4 kinds of whole shapes and one of 3 kinds of partial shapes. The examples of the input data are shown in Fig.2. We used rectangle, triangle, sin curve and step as whole shapes, and small upward triangle, small downward triangle, and small sin curve as partial shapes. The number of input data is 128 for whole shapes and 32 for partial shapes. The 12 types of data for one learning were made by combining the shapes. Position of a partial shapes were determined at random for each learning.

The learning are done by presenting input data and target data to the network. Each component of the target pattern is assigned to the output neuron, and if a specific shape exist in input data, it set to 1, otherwise it set to 0.

Learning processes were iterated until 200 times. But 3 of 4 times learning partial shapes were not included in the set of input data. This was for reason that the components in the super-neuron of Stage 1 were forced to converge to the whole shape more rapidly. If the super-neuron in Stage.1 has error components, they affect the rest components, so that the super-neuron in Stage.2 can not learn the partial shape.

Figure 3 shows the comparison of the proposed model with BP model for the error in the learning. The error increase when input data include a partial shape, so that the shapes of lines become like comb. In these graphs, the tops of pins indicate the error for the input data composed of a whole shape and a partial shape, while the bottoms of pins indicate the error for only a global feature. In BP model, the error in including only a whole shape could become to be near 0. However, the value in including both shapes could not converge to 0. Though we increase the learning number to 10000, it could not converge. Whereas in the proposed model it could become enough small in 200 times learning. To account for this result, two reasons can be thought. One of them is that BP model can not find the small difference of the local part in input data. Another is that BP model can not adapt itself to the transition of the position of a partial shape which is changed at random.



(a) BP model (b) new model

Fig.3 Comparison of error with BP model and new model

6 CONCLUSION

The pattern recognition model which can recognize the whole shape and the partial shape separately from mixed data has been proposed. Its efficiency has been demonstrated through the simulation. We found from the result that the proposed model can learn the partial shapes in input data faster than BP model. This model can decrease learning time and increase recognition rates for the error shape detection of products.

7 REFERENCE

[1] Hecht-Nielsen,R.,"Applications of counterpropagation networks",Proc.of the Int. Conf. on Neural Networks,II, 19-32,IEEE Press,New York,June 1987

[2] Rumelhart,D.E.,and McClelland,J.L.,Parallel Distributed Processing:Explorations in the Microstructure of Cognition,I&II,MIT Press,Cambridge MA,1986.

[3] K.Fukushima,Neural Networks and Information Processing(in Japanese),Asakura Publishing Co.,Ltd.,1989.

[4] Judith Dayhoff,Neural Network Architectures,A Van Nostrand Reinhold,1990.

[5]Y. Tomikawa and K. Nakayama,"Shape analysis for zippers with neural network"(in Japanese),Technical Report of IEICE.,NC92-144(1993-03)