2003 International Symposium on Advanced Intelligent Systems
September 25-28, 2003, Jeju, Korea

Additional Learning Framework for Multipurpose Image Recognition

Michiaki Itani
Keio University
3-14-1 Hiyoshi, Kohoku-ku,
Yokohama 223-8522, Japan
Email: itani@soft.ics.keio.ac.jp

Hitoshi Iyatomi
Keio University
3-14-1 Hiyoshi, Kohoku-ku,
Yokohama 223-8522, Japan
Email: iyatomi@soft.ics.keio.ac.jp

Masafumi Hagiwara
Keio University
3-14-1 Hiyoshi, Kohoku-ku,
Yokohama 223-8522, Japan
Email: hagiwara@soft.ics.keio.ac.jp

***Abstract*— We propose a new framework that aims at multipurpose image recognition, a difficult task for the conventional rule-based systems. This framework is formed based on the idea of computer-based learning algorithm. In this research, we introduce the new functions of an additional learning and a knowledge reconstruction on the Fuzzy Inference Neural Network (FINN) [1] to enable the system to accommodate new objects and enhance the accuracy as necessary. We examine the capability of the proposed framework using two examples. The first one is the capital letter recognition task from UCI machine learning repository to estimate the effectiveness of the framework itself. Even though the whole training data was not given in advance, the proposed framework operated with a small loss of accuracy by introducing functions of the additional learning and the knowledge reconstruction. The other is the scenery image recognition. We confirmed that the proposed framework could recognize images with high accuracy and accommodate new objects recursively.**

I. Introduction

Computer vision is a very important research theme to achieve high-level information processing on computers. It has extensive range of applications such as robotics, human interface, image searching, etc. We can classify computer vision studies into two major types.

The first one is conventional rule-based systems [2] [3] and the other is learning systems. The rule-based systems might perform tasks correctly and fastly by means of stored knowledge for target objects and situations. However, these systems require heuristic knowledge in advance and the performance is heavily depends on relevance of the knowledge and its expression. Namely, they need special model construction and have difficulty how to acquire appropriate knowledge and how to express it properly. On the other hand, learning systems are able to solve such problems. The learning systems can acquire these rules or knowledge due to learning ability of neural networks so that they operate tasks without predefined knowledge. Well-trained learning systems [4] [5] show superior system performance and can compete with rule-based systems in accuracy. However, these systems often use a back-propagation (BP) neural network, the most typical hierarchic neural network, so that the role of hidden units are not clear and it's difficult to know the behavior of the system.

Researches on fuzzy neural networks, which combine both merits of the learning ability of neural network and the knowledge description ability of fuzzy systems have been made since early 1990s. Generally speaking, they can create fuzzy if-then rules automatically from training data and obtained rules are expressed as their network weights. Therefore, it's easy to analyze the behavior of the system. We focus on a fuzzy inference neural network (FINN) [1], which is one of simple and efficient fuzzy neural network and we use it as a basic component.

Most of the studies on image recognition, either rule-based systems nor learning systems, has fixed number of outputs. They should be determined in advance. For example, researches [5] [6] have 10 pre-defined output types and they are not considered for extension. Therefore, when we intend to construct multipurpose recognition system, it is necessary to make the systems to expand the capability of handling rules and types of the output. However, most of the conventional image recognition researches employ the additional learning as the supplement of necessary rules to improve accuracy. So there is almost no study to support additional output types for system availability. The proposed framework corresponds to these problems with the additional learning of output types.

In this paper, first, we explain the structure of FINN on next section, and then describe our framework. In section 4, we confirm capability of proposing framework using two examples. We conclude our proposal in Section 5.

II. Fuzzy Inference Neural Network

A FINN can divide input-output data space and provide appropriate rules automatically. Fig.1 shows the structure of FINN. It consists of two layers. One is the input-output(I/O) layer and another is the rule-layer. Each node in the rule-layer represents one fuzzy rule. Weights from the input-part to the rule-layer and those from the rule-layer to the output-part are fully connected and they can store fuzzy if-then rules.

Membership functions as premise part are expressed in the weights. The weights from the input-part to the rule-layer indicate if-parts of fuzzy if-then rules and those from the rule-layer to the output-part indicate then-parts. Suppose that the number of neurons in the input-part, which is equal to the dimension of the input data, is N_1, the number of rules N_2, and the number of neurons in the output-part, which is equal to the dimension of the output data, is N_3.

The subscripts i, j, and k refer to the nodes in the input-part, those in the rule layer, and those in the output-part respectively. Fig.1 (b) shows an example of a membership function. The bell shaped membership function represents the if-part of fuzzy rule, which is placed between the ith input node and the jth node in the rule-layer. The membership function is expressed as:

$$\mu_{ij} = \exp\left(-\frac{(u_i - w_{ij})^2}{\sigma_{ij}^2}\right) \quad (1)$$

$$i = (1, 2, \cdots, N_1)\ j = (1, 2, \cdots, N_2)\ .$$

The shapes of membership functions are adjusted automatically in the learning phase by LMS learning Method. In the rule-layer, the degree of the jth rule ρ_j is calculated.

$$\rho_j = \prod_{i}^{N_1} \mu_{ij}^{\frac{1}{N_{adj}}} . \quad (2)$$

Here, N_{adj} is around $N_1/4$ which is the bias factor related to the number of input dimension. Then, the inference result of the kth node in the output-part, $\hat{y_k}$, is calculated by the following equation:

$$\hat{y_k} = \frac{\sum_j^{N_2}(w_{jk}\rho_j)}{\sum_j^{N_2}\rho_j} \quad (3)$$

$$k = (1, 2, \cdots, N_3)\ .$$

The logical form of the fuzzy inference if-then rules is given such as:
If u_1 is $\tilde{w}_{1j}$, and $\cdots$,u_i is $\tilde{w}_{ij}$, $\cdots$, u_{N_1} is $\tilde{w}_{N_1 j}$
then $\hat{y_k}$ is w_{jk}
where $\tilde{w}_{ij}$ means the value near w_{ij}. It should be noted here that it depends on the value of σ_{ij}.

FINN has two learning phases. The first one is the Self-organizing learning phase and another is the LMS learning phase.

First, at the Self-organizing phase, the center values of membership functions which correspond to the if-part and the estimated values which correspond to the then-part are determined by Kohonen's algorithm [7] temporarily. Second, Least Mean Square (LMS) learning phase (supervised learning phase) is executed to reduce the total mean-square error of the network to finely adjust the weights and the shapes of membership functions.

III. Additional Learning Framework

Next, we explain about proposed framework. Our framework consists of three phases: the basic learning phase, the additional learning phase and the knowledge reconstruction phase. Fig.2 shows the overview of our framework. In the basic learning phase, the system carries out training according to algorithm of the conventional FINN. In the next phase, the system performs interactive learning with the user. More concretely, the system creates additional rules and types of output during this phase for the accuracy and availability.

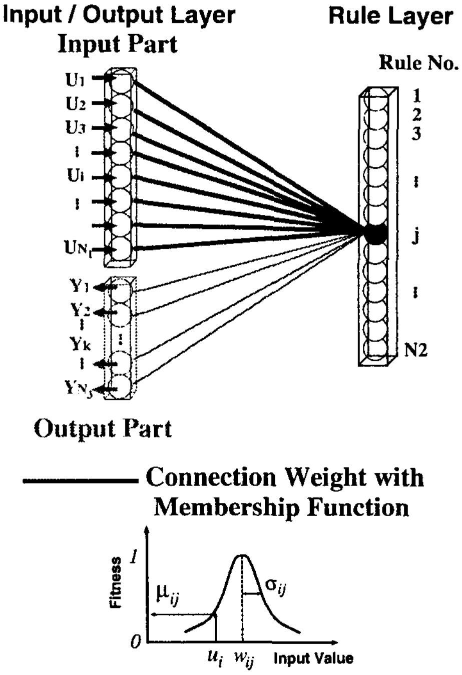


Fig. 1. (a) Structure of FINN and (b) its membership function

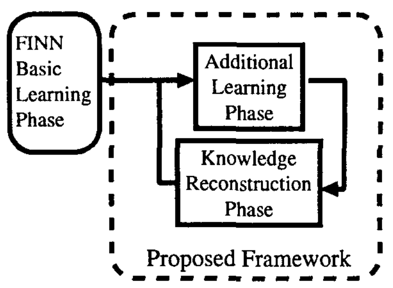


Fig. 2. Overview of Proposed Framework

In the last phase, the system reconstructs similar rules to prevent the explosive increase of the number of the rules. The additional learning of our framework contributes to improve not only the system accuracy by creating additional rules but also the system flexibility by creating new types of the output. Additional learning and knowledge reconstruction work as a paired mechanism. In this research, we perform additional learning as simple memorizing step so as not to omit the data. After that, in the last phase, knowledge reconstruction is carried out more appropriately.

A. Additional Learning

To realize the additional online learning, the proposed framework continues to adjust the rules through the system operations. Furthermore, our framework can correspond to the need of new types of the output. Additional learning of the rule is carried out when output value is less than the *threshold* value or when it is recognized wrongly. If confidence of output is low, the cause is considered that the network has no similar knowledge to the input. So it is possible that the network works more correctly by memorizing the new input vector as a new rule.

When $\hat{y_k}$ is less than the predetermined threshold, the new input is used to make new weight vector:

$$W_{N_2+1} = InputVector. \quad (4)$$

Then to strengthen the output part, the following operation is carried out:

$$Y_{N_2+1,k} \leftarrow C_1 \times Y_{N_2+1,k} \qquad (k = 1, \cdots, N_3) \qquad (C_1 > 1). \quad (5)$$

The number of the rule layer neurons is incremented:

$$N_2 \leftarrow N_2 + 1. \quad (6)$$

In order to form membership function, the width is determined as follows:

$$\sigma_{N_2,i} = 2\left(W_{N_2,i} - W_{E,i}\right) \qquad (i = 1, \cdots, N_1). \quad (7)$$

where:

$$E = arg\left(min_M |W_{N_1+1,j} - W_{M,j}|\right). \quad (8)$$

Eq. 5 means how the center value of new rule is determined. Eq. 8 shows the way to determine the width of the membership function. On the SOM learning phase of basic FINN learning, the width of membership function is fixed constant value. This is because FINN adjusts the width on the LMS learning phase.

Even though error recognition occurred, it means that the influence of wrong rule is rather strong. Eq. 3 shows that FINN decides the output value by weighted average. Consequently, if we add the input as the new rule simply, it is not certain of improvement. To prevent such a phenomena, our framework strengthen the connection of new rule especially and correct output by constant C_1. The flow of output type addition is as follows. First, new neuron is added to the I/O layer. It expresses the new type of output. Next, a new rule is added to the rule layer. It is carried out same as additional rule learning. Because the FINN does not have any knowledge on new output type.

B. Knowledge Reconstruction

We focused on creating new rules in the additional learning phase. By performing additional learning, some useless rules emerge. They cause problems: increase of computational cost, error recognition, and lower confidence. FINN needs to build its rule-base as weight vector. It is not desirable to keep those large number of rules, so our framework reconstructs the rules by combining the similar rules. It is a challenging area of study [8] to express the distance of knowledge. Since proposing study expresses the knowledge as fuzzy rules, it is considerable to express the distance as the distance between the fuzzy rules. Note that we need not the distance of $fuzzyvalue$ but the distance of $fuzzyrules$. Taking the integral as a distance of the two fuzzy rule is one of the best way to measure. The bell function of the FINN cannot be integrated with the low calculation costs. So, our framework compresses the number of the rules by reconstructing similar rules according to the following distance index:

$$Distance\left(W^1, W^2\right) =$$

$$\sum_{i=0}^{N_1} |W_i^1 - W_i^2| \left(1 - \sigma_i^1\right)\left(1 - \sigma_i^2\right) + \sum_{i=N_1}^{N_1+N_3} |W_i^1 - W_i^2| \quad (9)$$

We use this weighted Manhattan distance for the distance index of rules. Here σ is the width of the membership function, which takes $0 < \sigma < 1$ value. Accordingly the eq. 9 takes more than 0 value.

IV. Experimental Results

We examine the performance of the proposed framework using two examples. The first one is the capital letter recognition task from UCI machine learning repository. The second one is image recognition task.

A. Letter Recognition

This example is 16-inputs-26-outputs classification task. We used 16000 of 20000 letter data for learning, and used the others for estimation. We trained our system with letters from 'A' to 'J' in the basic learning phase and used remaining 'K' to 'Z' for additional training in the following phase. From this experiment, our framework achieved 81 % classification accuracy. On the other hand, the latest research using this dataset achieved 80 %, and the basic FINN showed 90 % accuracy, respectively, when all of the training data were given in advance. Even though the whole training data had not been given in advance, the proposed framework operated with small loss of accuracy by introducing functions of the additional learning and the knowledge reconstruction. We used the former 8000 letter data for the basic learning phase, and the latter 8000 for the additional learning phase. On the additional learning phase, we performed knowledge reconstruction phase for each 100 new rules.

Fig.3 shows the error rate of the additional learning for the rest 4000 test data. Fig.4 shows the change of the number of the rules during the additional learning. In the early additional learning phase, there exists an increase of the errors. This is due to the addition of the new output and the growth of the number of the rules caused by the new output type addition. There is also influence of merging proper rules into one rule. The oscillatory of the Fig.4 shows that the new rules contain numerous useless rules, while the reduction of the errors shows that our framework is successfully reconstructing rules.

B. Image Recognition

Another task is the scenery image recognition. The given image was segmented into several regions and 26 characteristics were calculated from each region. In this experiment, these characteristics were composed of colors, position, shapes and texture attributes. The objective of this task was to give proper

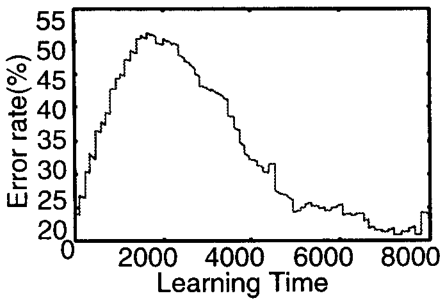


Fig. 3. Error rate on the additional learning phase

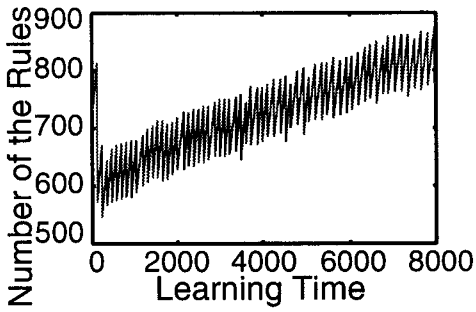


Fig. 4. Number of the rules

recognition labels such as wood, sky, water and the like to each region. Fig.5 shows the example of labels. We confirmed that the proposed framework could recognize images with high accuracy and accommodate new objects recursively. Examples of the recognition are shown in Fig. 6.

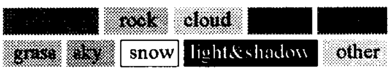


Fig. 5. Example of the labels

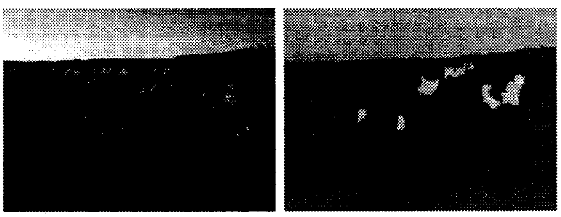


Fig. 6. Original image and the result

We examined the effectiveness of the output type addition by the next experimental task. First we constructed 26-input-9-output FINN for scenery image recognition. This network has one less output types, the output type 'snow' is missing. We trained this network with 70 scenery images, which contain about 1000 segments, without the images containing 'snow' area. Then we added the new output type, 'snow', and tested using the image show in Fig. 7 which contains wide snow area. As a result, this FINN were not able to recognize 'snow' at all as Fig. 8(a). After additional learning some images with snow area, our framework could recognize 'snow' area successfully.

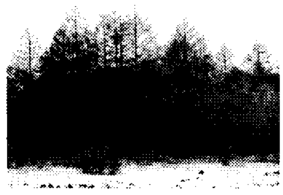
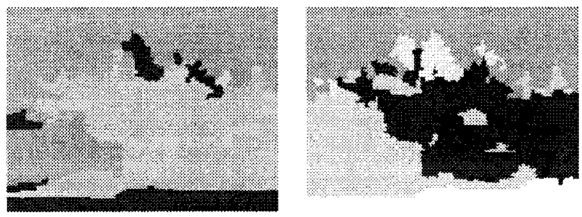


Fig. 7. Original Image



(a) Pre addition (b) Post addition

Fig. 8. Result

V. Conclusion

In this research, we proposed Additional Learning Machine Framework for Multipurpose Image Recognition. This framework can adapt the need for the new types of the output during system operations, that is not possible for the conventional studies. We have confirmed the effectiveness of this framework by two example tasks. The proposed framework can be applied to not only natural scenery images but also more general images that contain artificial objects.

Acknowledgement This work is supported in part by a Grant in Aid for the 21st century Center Of Excellence for Optical and Electronic Device Technology for Access Network from the Ministry of Education, Culture, Sport, Science, and Technology in Japan.

References

[1] Takatoshi Nishina, Masafumi Hagiwara: "Fuzzy Inference neural network", Neurocomputing 14, pp.223-239, 1997.

[2] M.Mirnehdi:"Feedback Control Strategies for Object Recognition", IEEE Trans. Image Processing. Vol.8, No.8, pp.1084-1101, 1999.

[3] M.S.Thomas, and A.F.Martin: "Context-based vision: Recognizing objects using information from both 2-D and 3-D imagery", IEEE Trans. Pattern Anal. & Syst., vol.13, No.10, pp.1050-1065, Oct.1991.

[4] Doulamis, A.D., Doulamis, N.D., Kollias, S.D.:"On-line retrainable neural networks: improving the performance of neural networks in image analysis problems", IEEE Trans. Neural Networks, Vol. 11, pp.137-155, 2000.

[5] M.Mukunoki, M.Minoru, and K.Ikeda:"A Retrieval Method of Outdoor Scenes using Object Sketch and an Automatic Index Generation Method", The Trans. of IEICE Vol.J79-D-II, No.6, pp.1025-1033, 1996-6 (In Japanese).

[6] Hitoshi Iyatomi, Masafumi Hagiwara: "Knowledge Extraction from Scenery Image and Recognition using Fuzzy Inference Neural Network", The Trans. of IEICE Vol.J82-D-II, No.4, pp.685-693, 1999-4. (In Japanese)

[7] Teuvo Kohonen: "The Self-Organizing Map", Proceeding of IEEE, Vol.78, no.9, pp.1464-1480, September, 1990.

[8] Yixin Che, James Z. Wnag: "A Region-Based Fuzzy Feature Matching Approach to Content-Based Image Retrieval", IEEE Trans. Pattern Anal. & Syst., vol.24, No.9, pp.420-425, Sep.2002.