[7] D. Koson, C. Kitchen, M. Stokdolosky, and D. Stokdolosky, "Psychological testing by computer: effect of response bias," *Educational Psychological Measurement*, vol. 30, pp. 803–810, 1970.
[8] G. J. Lautenschlager and V. L. Flaherty, "Computer administration of questions: more desirable or more social desirability," *J. Applied Psychology*, vol. 75, no. 3, pp. 310–314, 1990.
[9] R. B. Littermann, "Forecasting with Bayesian Vector autoregressions–Five years of experience," *J. Business and Economic Statistics* vol. 4, pp. 25–38, 1986.
[10] M. Lobrovich, "Computer interviews net quality employees," *Computer Decisions*, vol. 14, no. 3, p. 72, 1982.
[11] D. L. Mitchell, *A Multivariate Analysis of the Effects of Gender and Computer vs. Paper/Pencil Modes of Administration on Survey Results*, Unpublished DBA Dissertation, College of Administration and Business, Louisiana Tech University, Ruston, LA, 71270, 1993.
[12] J. E. Rafael, "Computer-assisted telephone interview benefits far outweigh the problems," *Marketing News*, vol. 18, no. 23, p. 16, 1984.
[13] V. Rezmovic, "The Effects of computerized experimentation on response variance," *Behavior Research Methods and Instrumentation*, vol. 9, pp. 144–147, 1977.
[14] P. Rosenfeld, L. M. Doherty, S. M. Vincino, J. Kantor, and J. Greaves, "Attitude assessment in organizations: Testing three microcomputer-based survey systems," *J. General Psychology*, vol. 116, no. 2, pp. 145–154, 1989.
[15] Sawtooth Software, 1007 Church Street, Suite 402, Evanston, IL, 60201.
[16] J. M. Shanks, "Computer assisted surveys: Recent progress and future developments," in James I. Cash and Jay F. Nunamaker, Jr., (ed.), *The Information Systems Research Challenge: Survey Research Methods*, 3. Boston, MA: Harvard Business School, 1991.
[17] M. Silver, "Decisional guidance for computer-based decision support," *MIS Quarterly*, vol. 15, no. 1, pp. 105–122, 1991.
[18] W. V. Slack, "A History of computerized medical interviews," *M.D. Computing*, vol. 1, no. 5, pp. 52–59, 68, 1984.

Combining Multiple Neural Networks by Fuzzy Integral for Robust Classification

Sung-Bae Cho and Jin H. Kim

***Abstract*—Recently, in the area of artificial neural networks, the concept of combining multiple networks has been proposed as a new direction for the development of highly reliable neural network systems. In this paper we propose a method for multinetwork combination based on the fuzzy integral. This technique nonlinearly combines objective evidence, in the form of a fuzzy membership function, with subjective evaluation of the worth of the individual neural networks with respect to the decision. The experimental results with the recognition problem of on-line handwriting characters confirm the superiority of the presented method to the other voting techniques.**

I. Introduction

In the conventional approach for applying neural networks to real-world problems, one starts with a training database, chooses a network by making an educated guess, and then uses a learning algorithm to load as many of the training examples as possible onto the network. In principle one can always find a neural network that can solve a given problem, provided that there is no restriction on the size of the network and an infinite amount of data is available. In practice, however, one has to deal with a limited amount of resources, and has to heavily rely on the generalization abilities of the network.

Manuscript received April 16, 1993; revised December 22, 1993 and March 14, 1994. This work was supported in part by a grant from the Korean Science and Engineering Foundation (KOSEF) and by the IEEE Computer Society.

The authors are with the Center for Artificial Intelligence Research and Computer Science Department, Korea Advanced Institute of Science and Technology, 373-1 Koosung-dong, Yoosung-ku, Taejeon 305-701, Republic of Korea.

S.-B. Cho is currently with ATR Human Information Processing Research Laboratories, 2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02, Japan. His permanent address is the Department of Computer Science, Yonsei University, 134 Shinchon-dong, Sudaemoon-ku, Seoul 120-749, Republic of Korea.

IEEE Log Number 9405708.

To cope with this difficulty, a variety of modular neural networks have been proposed [1]. Several researchers have attempted to use multiple networks with an appropriate collective decision strategy. The idea of representing a decision from multiple sources is not new. Several methods for combining evidence produced by multiple information sources have been applied in statistics, management sciences, and pattern recognition [2], [3]. Mori and Yokosawa [4] developed a large-scale neural network for recognizing Kanji characters by dividing the original task into several subtasks, with networks for each subtasks, and then integrating these subnetworks in a large network. Jacobs *et al.* [5] also proposed a modular version of a multilayer supervised network. It is a tightly coupled system composed of many separate networks, each of which learns to handle a subset of the complete set of training cases.

Multiple networks are desirable because selection of the weights is an optimization problem with many local minima. While the conventional approach utilizes the one with the best performance, this technique improves an estimate of a given statistic by combining multiple estimates generated by different networks. If we have networks with different accuracy, however, the estimation would be improved by giving the combiner the ability to bias the outputs based on *a priori* knowledge about the reliability of the networks.

In this paper, we present a multiple neural network architecture combined by an evidence fusion technique, based on the notion of the fuzzy integral. In the fuzzy integral both objective evidence supplied by various sources and the expected worth of subsets of these sources are considered in the fusion process; It combines objective evidence for a hypothesis with the system's expectation of the importance of that evidence to the hypothesis. This approach may provide a possibility for incorporating any *a priori* knowledge regarding the underlying problem to improve the ability of the networks to generalize.

The rest of this paper is organized as follows. Section II reviews the back propagation neural network as a classifier, and shows how it is related with the Bayes classifier. In Section III, we introduce multiple neural networks and two typical methods of combining them. The proposed method based on the fuzzy integral is demonstrated in Section IV. Explained in Section V are results with the recognition of on-line handwriting characters. Finally, Section VI discusses the summary of the paper and the future research.

II. Neural Network as Bayes Classifier

A neural network can be considered as a mapping device between an input set and an output set. Mathematically, a neural network represents a function F that maps I into O; $F: I \rightarrow O$, or $y = f(x)$ where $y \in O$ and $x \in I$. Since the classification problem is a mapping from the feature space to some set of output classes,we can formalize the neural network, especially two-layer feed forward neural network trained with the generalized delta rule, as a classifier.

Fig. 1 shows a two-layer neural network classifier with T neurons in the input layer, H neurons in the hidden layer, and c neurons in the output layer. Here, T is the number of features, c is the number

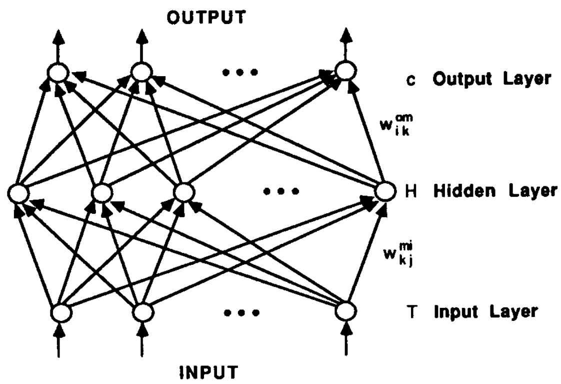

Fig. 1. A two-layered neural network architecture. In this network, T is the number of features, c is the number of classes, and H is an appropriately selected number.

of classes, and H is an appropriately selected number[1]. The network is fully connected between adjacent layers. The operation of this network can be thought of as a nonlinear decision-making process; Given an unknown input $X = (x_1, x_2, \ldots, x_T)$ and the class set $\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}$, each output neuron estimates the possibility y_i of belonging to this class by

$$y_i = f\left\{\sum_{k=1}^{H} w_{ik}^{om} f\left\{\sum_{j=1}^{T} w_{kj}^{mi} x_j\right\}\right\}, \tag{1}$$

where w_{kj}^{mi} is a weight between the jth input neuron and the kth hidden neuron, w_{ik}^{om} is a weight from the kth hidden neuron to the ith class output, and f is a sigmoid function such as $f(x) = 1/(1+e^{-x})$. The neuron having the maximum value is selected as the corresponding class.

In the meantime, the outputs of neural networks are not just likelihoods or binary logical values near zero or one. Instead, they are estimates of Bayesian *a posteriori* probabilities [6]. With a squared-error cost function, the network parameters are chosen to minimize the following:

$$E\left[\sum_{i=1}^{c} (y_i(X) - d_i)^2\right] \tag{2}$$

where $E[\cdot]$ is the expectation operator, $\{y_i(X) \mid i = 1, \ldots, c\}$ the outputs of the network, and $\{d_i \mid i = 1, \ldots, c\}$ the desired outputs for all output neurons. Performing several treatments in this formula allows it to cast in a form commonly used in statistics that provides much insight as to the minimizing values for $y_i(X)$:

$$E\left[\sum_{i=1}^{c} (y_i(X) - E[d_i \mid X])^2\right] + E\left[\sum_{i=1}^{c} \mathrm{var}[d_i \mid X]\right] \tag{3}$$

$E[d_i \mid X]$ is the conditional expectations of d_i, and $\mathrm{var}[d_i \mid X]$ is the conditional variance of d_i.

Since the second term in (3) is independent of the network outputs, minimization of the squared-error cost function is achieved by choosing network parameters to minimize the first expectation term.This term is simply the mean-squared error between the network outputs $y_i(X)$ and the conditional expectation of the desired outputs. For a 1 of M problem, d_i equals 1 if the input X belongs to class ω_i and 0 otherwise. Thus, the conditional expectations are the following:

$$\begin{aligned} E[d_i \mid X] &= \sum_{j=1}^{c} d_i P(\omega_j \mid X) \\ &= P(\omega_i \mid X) \end{aligned} \tag{4}$$

which are the Bayesian probabilities. Therefore, neural networks trained to minimize a mean squared-error cost function for a 1 of M problem yield network outputs that estimate the Bayesian *posteriori* probabilities.

[1] There has been a long debate as to how to determine H as appropriate for any given problem. This has motivated the development of several constructive training techniques, such as Fahlman's *Cascade Correlation*.

III. Multiple Neural Networks Classifier

The network mentioned in the previous section trains on a set of example patterns and discovers relationships that distinguish the patterns. A network of a finite size, however, does not often load a particular mapping completely or it generalizes poorly. Increasing the size and number of hidden layers most often does not lead to any improvements [7]. Furthermore, in complex problems such as character recognition, both the number of available features and the number of classes are large. The features are neither statistically independent nor unimodally distributed. Therefore, if we can make the networks consider only a specific part of the complete mapping and combine them, the hybrid estimator can perform better in the mean squared error sense than any single network.

The basic idea of multiple network scheme is to develop n independently trained neural networks with relevant features, and to classify a given input pattern by obtaining a classification from each copy of the network and then utilizing combination methods to decide the collective classification [8], [9] (see Fig. 2). There are a lot of previous works that report the usefulness of voting procedures in classification area. The methods based on voting techniques consider the result of each network as an expert judgement. A variety of voting procedures can be adopted from group decision making theory such as unanimity, majority, plurality, Borda count, and so on. In particular, we present two of them, majority voting and Borda count.

The majority voting rule chooses the classification made by more than half the networks. When there is no agreement among more than half the networks, the result is considered an error. To appreciate network performance, assume that all neural networks arrive at the correct classification with a certain likelihood $1-p$ and that they make independent errors. The chances of seeing exactly k errors among n copies of the network is then

$$\binom{n}{k} p^k (1-p)^{n-k} \tag{5}$$

which gives the following likelihood of the majority rule being in error

$$\sum_{k>n/2}^{n} \binom{n}{k} p^k (1-p)^{n-k}. \tag{6}$$

It can be shown by induction when n is odd (or separately when n is even) that provided $p < 1/2$, (6) is monotonically decreasing in n. In other words, if each network can get the correct answer more than half the time, and if network responses are independent, then the more networks used, the less the likelihood of an error by a majority decision rule. In the limit of infinite n, the coordinated error rate goes to zero.

For any particular class c, the Borda count is the sum of the number of classes ranked below c by each network; Let $B_j(c)$ be the number of classes ranked below the class c by the jth network. Then, the Borda count for class c is $B(c) = \sum_{j=1}^{n} B_j(c)$. The final decision is given by selecting the class label whose Borda count is the largest.

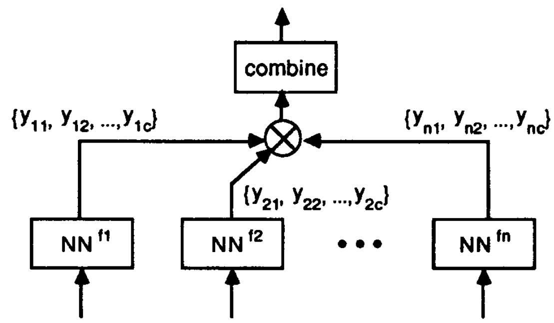

Fig. 2. The multiple network architecture with consensus scheme.

IV. Network Integration With Fuzzy Integral

The fuzzy integral introduced by Sugeno [10] and the associated fuzzy measures [11], [12] provide a useful way for aggregating information. In the following we introduce some definitions and properties about the fuzzy integral.

Definition 1: Let X be a finite set of elements. A set function $g : 2^X \rightarrow [0,1]$ with

1) $g(\emptyset) = 0$
2) $g(X) = 1$
3) $g(A) \leq g(B)$ *if* $A \subset B$

is called a fuzzy measure. Note that g is not necessarily additive. This property of monotonicity is substituted for the additivity property of the measure.

From the definition of a fuzzy measure g, Sugeno introduced the so-called g_λ-fuzzy measures satisfying the following additional property: For all $A, B \subset X$ and $A \cap B = \emptyset$,

$$g(A \cup B) = g(A) + g(B) + \lambda g(A)g(B), \text{ for some } \lambda > -1.$$

It affords that the measure of the union of two disjoint subsets can be directly computed from the component measures.

Using the notion of fuzzy measures, Sugeno developed the concept of the fuzzy integral, which is a nonlinear functional that is defined with respect to a fuzzy measure, especially g_λ-fuzzy measure [10], [11], [13].

Definition 2: Let X be a finite set, and $h : X \rightarrow [0,1]$ be a fuzzy subset of X. The fuzzy integral over X of the function h with respect to a fuzzy measure g is defined by

$$\begin{aligned} h(x) \circ g(\cdot) &= \max_{E \subseteq X} \left[\min \left(\min_{x \in E} h(x), g(E) \right) \right] \\ &= \max_{\alpha \in [0,1]} \left[\min \left(\alpha, g(h_\alpha) \right) \right] \end{aligned} \tag{7}$$

where h_α is the α level set of h,

$$h_\alpha = \{x \mid h(x) \geq \alpha\}. \tag{8}$$

The following properties of the fuzzy integral can be easily proved [13].

1) If $h(x) = c$, for all $x \in X$, $0 \leq c \leq 1$, then
$$h(x) \circ g(\cdot) = c.$$
2) If $h_1(x) \leq h_2(x)$ for all $x \in X$, then
$$h_1(x) \circ g(\cdot) \leq h_2(x) \circ g(\cdot).$$
3) If $\{A_i \mid i = 1, \ldots, n\}$ is a partition of the set X, then
$$h(x) \circ g(\cdot) \geq \max_{i=1}^{n} e_i.$$

where e_i is the fuzzy integral of h with respect to g over A_i. For further details on the properties of the fuzzy integral and associated fuzzy measures for aggregating information, see the recent publication made by Yager [12].

The calculation of the fuzzy integral is as follows: Let $Y = \{y_1, y_2, \ldots, y_n\}$ be a finite set and let $h : Y \rightarrow [0,1]$ be a function. Suppose $h(y_1) \geq h(y_2) \geq \cdots \geq h(y_n)$, (if not, Y is rearranged so that this relation holds). Then a fuzzy integral, e, with respect to a fuzzy measure g over Y can be computed by

$$e = \max_{i=1}^{n}[\min(h(y_i), g(A_i))] \tag{9}$$

where $A_i = \{y_1, y_2, \ldots, y_i\}$.

Note that when g is a g_λ-fuzzy measure, the values of $g(A_i)$ can be determined recursively as

$$g(A_1) = g(\{y_1\}) = g^1 \tag{10}$$
$$g(A_i) = g^i + g(A_{i-1}) + \lambda g^i g(A_{i-1}), \quad \text{for } 1 < i \leq n. \tag{11}$$

λ is given by solving the equation

$$\lambda + 1 = \prod_{i=1}^{n}(1 + \lambda g^i) \tag{12}$$

where $\lambda \in (-1, +\infty)$, and $\lambda \neq 0$. This can be easily calculated by solving an $(n-1)$st degree polynomial and finding the unique root greater than -1.

Thus the calculation of the fuzzy integral with respect to a g_λ-fuzzy measure would only require the knowledge of the density function, where the ith density, g^i, is interpreted as the degree of importance of the source y_i towards the final evaluation. The value obtained from comparing the evidence and the importance in terms of the min operator is interpreted as the grade of agreement between real possibilities, $h(y)$, and the expectations, g. Hence fuzzy integration is interpreted as searching for the maximal grade of agreement between the objective evidence and the expectation.

Let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}$ be a set of classes of interest. Note that each ω_i may, in fact, be a set of classes by itself. Let $Y = \{y_1, y_2, \ldots, y_n\}$ be a set of neural networks, and A be the object under consideration for recognition. Let $h_k : Y \rightarrow [0,1]$ be the partial evaluation of the object A for class ω_k, that is, $h_k(y_i)$ is an indication of how certain we are in the classification of object A to be in class ω_k using the network y_i where 1 indicates absolute certainty that the object A is really in class ω_k and 0 implies absolute certainty that the object A is not in ω_k.

Now corresponding to each y_i the degree of importance, g^i, of how important y_i is in the recognition of the class ω_k must be given. These densities can be subjectively assigned by an expert, or can be generated from training data. The g^i's define the fuzzy density mapping. Hence λ can be calculated using (12) and the g_λ-fuzzy measure, g can be constructed. Now, using (9) to (12), the fuzzy integral can be calculated. Thus the following algorithm for network integration is given.

Algorithm: Network fusion by fuzzy integral
    calculate λ; /* using importance of each network */
    for each class ω_k **do**
        for each neural network y_i **do**
            calculate $h_k(y_i)$;
            determine $g_k(\{y_i\})$;
        end_for
        compute the fuzzy integral for the class;
    end_for
    determine the final class;

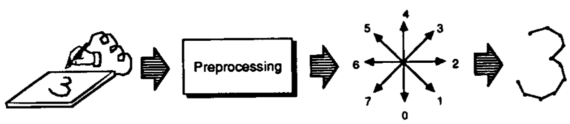

Fig. 3. Schematic diagram of the process for data collection.

In the final step, the class ω_k with the largest integral value can be chosen as the output class.

V. Experimental Results

In the experiment, handwriting characters were inputted to the computer (SUN workstation) by an LCD tablet of Photron FIOS-6440 which samples 80 dots per second. The tasks were to classify Arabic numerals, uppercase letters, and lowercase letters which were collected from 13 writers. The writers were told to draw the numerals and letters into prepared square boxes in order to facilitate segmentation.

An input character consists of a set of strokes, each of which begins with a pen-down movement and ends with pen-up movements. Several preprocessing algorithms were applied to successive data points in a stroke to reduce quantization noises and fluctuations of the writer's pen motion. The processes used are as follows: the wild point reduction, the dot reduction, the hook analysis, the three point smoothing, peak preserving filtering, and N point normalization. A sequence of preprocessed data points is approximated by a sequence of 8-directional straight-line segments [14]. The procedure for collecting handwriting data is schematically presented in Fig. 3.

For training the neural network, 40 examples per each class were used, while for recognition an additional 500 examples were used as test inputs. To evaluate the performance of the proposed network scheme, we implemented three different networks, each of which is a two-layered neural network having different number of input neurons and 20 hidden neurons. NN_1, NN_2 and NN_3 have 10, 15, and 20 input neurons, respectively. In this fashion each network makes the decision through its own resolution; NN_1 using sparsely sampled input produces the result by means of coarser view of input image, while NN_3 uses finer view. Thus, NN_1 has large possibility to overcome the variation or noise of input image though it utilizes rather blurred input.

The selection of the features is largely *adhoc* and no attempt was made to find an optimal coding scheme although this is an important issue in character recognition problems. Our objective here is to evaluate and compare the presented methods through an example which has a certain complexity and practical significance.

The EBP algorithm [15] was used for the training and the iterative estimation process was stopped when an average squared error of 0.9 over the training set was obtained, or when the number of iteration reaches 1000, which was adopted mainly for preventing networks from over training. The parameter values used for training were: learning rate is 0.4 and momentum parameter is 0.6. An input vector is classified as belonging to the output class associated with the highest output activation.

First, the behavior of the fuzzy integral of a function h with respect to a g_λ-fuzzy measure, g is examined. Table I shows these results. Here, each case shows a set of fuzzy densities corresponding to the three networks and the recognition rates of numerals, uppercase letters, and lowercase letters using the fuzzy integral on the three networks. Using (12), the Sugeno measure g must have a parameter λ satisfying $0.006\lambda^2 + 0.11\lambda - 0.4 = 0$. The unique root greater than -1 for this equation is $\lambda = 3.109$.

As expected, the recognition results in the table depend on the g values. When the g values change, the new fuzzy integral value will change depending on how these changes are balanced with respect to the source corresponding to the fuzzy integral value. We assigned the g values, the degree of importance of each network, based on how good these networks performed on training data. The real values of these densities with the corresponding λ's are shown in Table II. All λ values are very close to -1 because the sum of g values is greater than 1. In this case, the degree of importance may be interpreted as a plausibility value. Banon showed that $\lambda \leq 0$ if g is a plausibility measure [16].

TABLE I
The Recognition Rates of the Fuzzy Integral for Different Densities Functions (%)

Case	g^1	g^2	g^3	Numerals	Uppercases	Lowercases
1	0.1	0.2	0.3	78.4	71.6	65.4
2	0.1	0.3	0.2	79.0	73.4	66.8
3	0.2	0.1	0.3	78.8	72.2	64.8
4	0.2	0.3	0.1	79.4	73.8	69.2
5	0.3	0.1	0.2	79.8	74.0	66.2
6	0.3	0.2	0.1	80.2	75.2	70.4

TABLE II
Fuzzy Densities (Recognition Rates on the Training Data) and the Corresponding λ

Subject	g^1	g^2	g^3	λ
Numerals	0.8875	0.8750	0.8925	−0.9984
Uppercases	0.8568	0.8932	0.9334	−0.9989
Lowercases	0.7585	0.8859	0.8034	−0.9940

Table III reports the results of network fusion using the fuzzy integral on the three different networks for numerals. In this table the value in the parentheses represent the confidence of the evaluation result. As can be seen, cases 2 and 3 were misclassified by NN_3 and NN_2, respectively. However, in the final evaluations they were correctly classified. In cases 5 and 17, one network with strong evidence overwhelmed the other networks, producing correct classification. Furthermore, in case 15, the fuzzy integral made a correct decision despite that the partial decisions from the individual neural networks were completely inconsistent. The effect of misclassification by the other networks has given rise to small fuzzy integral values for the correct classification in this case.

Table IV shows the recognition rates of numerals, uppercase letters, and lowercase letters with respect to the three different networks and their combinations by utilizing the consensus methods. In this table it is seen that the recognition rates of the consensus methods outperformed those of the individual networks in the cases of numerals and uppercase letters, but did not in the case of lowercase letters.

To understand why the consensus scheme did not produce better result than of the individual networks in the lowercase letter, let us consider three networks with error rates e_1, e_2, and e_3. If we take the minimum error to be e_1, the condition that the consensus (e.g., majority rule) error rate be less than the best individual error rate is

$$
\begin{aligned}
e_1 > {} & e_1 e_2 e_3 + (1 - e_1) e_2 e_3 \\
& + e_1 (1 - e_2) e_3 + e_1 e_2 (1 - e_3). \qquad (13)
\end{aligned}
$$

While the other two cases satisfy the above condition, the lowercase letter case does not (e_1 is 0.302, but the righthand side becomes 0.322).

In order to prove our conjecture, we performed the same experiments after making some effort to improve the performance of each network. The result is given in Table V. As can be seen, the overall classification rates for the fuzzy integral became higher than those for other consensus methods as well as individual networks. Fig. 4 illustrates the error rates of the multiple network scheme as compared

TABLE III
RESULTS OF NETWORK FUSION USING THE FUZZY INTEGRAL ON THREE DIFFERENT NETWORKS FOR NUMERALS

Data index	Actual class	Partial decision NN_1	NN_2	NN_3	Fuzzy integral decision
1	5	5 (0.9859)	5 (0.8995)	5 (0.9941)	5 (0.9598)
2	6	6 (0.9968)	6 (0.9985)	5 (0.3301)	6 (0.6877)
3	8	8 (0.9999)	0 (0.0022)	8 (0.9996)	8 (0.6668)
4	2	2 (0.9922)	2 (0.9998)	2 (0.9920)	2 (0.9946)
5	7	8 (0.0162)	8 (0.0087)	7 (0.9615)	7 (0.3205)
6	9	8 (0.0001)	7 (0.1195)	7 (0.4780)	7 (0.1991)
7	7	6 (0.0137)	7 (0.9903)	7 (0.9988)	7 (0.6630)
8	7	2 (0.1342)	7 (0.9677)	7 (0.0023)	7 (0.3233)
9	1	1 (0.9999)	1 (0.9972)	1 (0.9993)	1 (0.9988)
10	0	6 (0.7116)	6 (0.4098)	8 (0.8098)	6 (0.3753)
11	7	1 (0.1794)	8 (0.0003)	1 (0.0080)	1 (0.0625)
12	4	4 (0.9998)	4 (0.9999)	9 (0.9964)	4 (0.6694)
13	9	9 (0.9965)	9 (0.9958)	8 (0.0740)	9 (0.6691)
14	3	3 (0.9987)	3 (0.9912)	3 (0.9999)	3 (0.9966)
15	7	8 (0.0365)	0 (0.0460)	7 (0.4831)	7 (0.1610)
16	5	5 (0.9311)	3 (0.1304)	3 (0.6245)	5 (0.3265)
17	2	8 (0.3470)	2 (0.9983)	8 (0.2092)	2 (0.3327)
18	8	8 (0.9899)	0 (0.9669)	8 (0.6815)	8 (0.8384)
19	0	1 (0.0353)	4 (0.0004)	4 (0.0004)	1 (0.0118)
20	9	9 (0.7519)	9 (0.3799)	9 (0.9540)	9 (0.6953)
21	0	8 (0.8032)	0 (0.9994)	0 (0.9993)	0 (0.6662)
22	4	4 (0.9997)	9 (0.9170)	4 (0.9871)	4 (0.7099)
23	9	9 (0.9989)	1 (0.9944)	9 (0.9902)	9 (0.6705)
24	4	4 (0.9998)	4 (0.9999)	4 (0.9995)	4 (0.9997)
25	8	8 (0.9998)	0 (0.9882)	8 (0.8353)	8 (0.6204)

TABLE IV
THE RESULT OF RECOGNITION RATES (1)

Subject	NN_1	NN_2	NN_3	Majority	Borda count	Fuzzy integral
Numerals	77.4	76.0	76.2	77.8	78.2	81.2
Uppercases	73.2	66.8	70.8	74.0	77.4	77.6
Lowercases	59.0	69.8	57.4	65.4	67.8	66.0

TABLE V
THE RESULT OF RECOGNITION RATES (2)

Subject	NN_1	NN_2	NN_3	Majority	Borda count	Fuzzy integral
Numerals	82.6	81.2	81.0	84.9	86.8	88.1
Uppercases	73.2	66.8	70.8	74.0	77.4	77.6
Lowercases	73.9	71.8	72.1	74.6	79.1	80.3

with each network. In this figure, consensus means the combining method of the three networks with the fuzzy integral. The results show that this methodology produced good classification results based on the objective information and the subjective expectation of the importance of the information.

VI. CONCLUDING REMARKS

In this paper, we have introduced a design method of the multilayer neural network, called the multiple network scheme, and proposed a consensus method based on the fuzzy integral. The most important advantage of this methodology is that not only are the classification results combined but that the relative importance of the different networks is also considered. Initial trials to use the method for classifying a large set of on-line handwriting characters were promising, but several works are remained for further research.

The relatively easy ones are to increase the recognition rate of each base neural network for practical usage and to try the same experiments with the increased number of networks. Furthermore, an interesting theoretical development could be a deeper discussion on the issue of aggregating the neural network outputs by some alternative aggregating mechanism, such as generalized means, OWA operators [12], and Dempster-Shafer method [2].

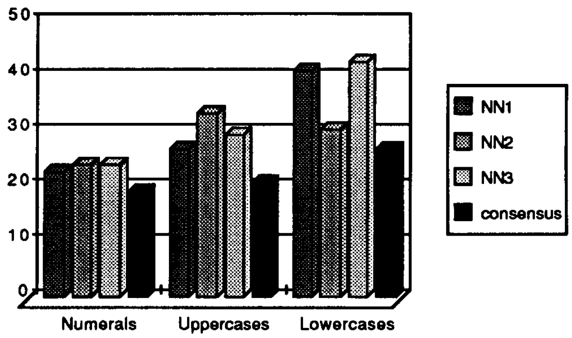

Fig. 4. A comparison of error rates of the proposed consensus method with the three separate networks with respect to the three tasks.

ACKNOWLEDGMENT

The first author wishes to thank Dr. Katsunori Shimohara, Head of Department 6 at ATR HIP research laboratories, for his interest in this work.

REFERENCES

[1] F. Fogelman Soulie, "Neural network architectures and algorithms: a perspective," in *Artificial Neural Networks*. Netherlands: Elsevier Science Publishers B.V., 1991, pp. 605–615.
[2] L. Xu, A. Krzyzak and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Trans. Syst. Man Cybern.*, vol. 22, pp. 688–704, 1992.
[3] J. A. Benediktsson and P. H. Swain, "Consensus theoretic classification methods," *IEEE Trans. Syst. Man Cybern.*, vol. 22, pp. 418–435, 1992.
[4] Y. Mori and K. Yokosawa, "Neural networks that learn to discriminate similar Kanji characters," *Advances in Neural Information Processing Systems I*. Morgan Kaufmann, 1989, pp. 332–347.
[5] R. A. Jacobs, M. I. Jordan, S. J. Nowlan and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, pp. 79–87, 1991.
[6] M. D. Richard and R. P. Lippmann, "Neural network classifiers estimate Bayesian *a posteriori* probabilities," *Neural Computation*, vol. 3, pp. 461–483, 1991.
[7] A. Waibel, "Connectionist glue: Modular design of neural speech systems," in *Proc. 1988 Connectionist Models Summer School*, pp. 417–425, 1988.
[8] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 993–1001, 1990.
[9] S. Shlien, "Multiple binary decision tree classifiers," *Pattern Recognition*, vol. 23, pp. 757–763, 1990.
[10] M. Sugeno, "Fuzzy measures and fuzzy integrals: A survey," *Fuzzy Automata and Decision Processes*. Amsterdam: North Holland, pp. 89–102, 1977.
[11] K. Leszeynski, P. Penczek and W. Grochulskki, "Sugeno's fuzzy measures and fuzzy clustering," *Fuzzy Sets and Systems*, vol. 15, pp. 147–158, 1985.
[12] R. R. Yager, "Element selection from a fuzzy subset using the fuzzy integral," *IEEE Trans. Syst. Man Cybern.*, vol. 23, pp. 467–477, 1993.
[13] H. Tahani and J. M. Keller, "Information fusion in computer vision using the fuzzy integral," *IEEE Trans. Syst. Man. Cybern.*, vol. 20, pp. 733–741, 1990.
[14] H. Freeman, "Computer processing in line drawing images," *Computing Survey*, vol. 6, pp. 57–98, Mar. 1974.
[15] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Exploration of the Microstructure of Cognition*, vol. 1; D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986.
[16] G. Banon, "Distinction between several subsets of fuzzy measures," *Fuzzy Sets & Systems*, vol. 5, pp. 291–305, 1981.