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 multineuron 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.

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 subtask, 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 $O$, that is, $y = f(x)$ 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. 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

$$y_i = f\left(\sum_{k=1}^{H} w_{ik}^m f\left(\sum_{j=1}^{c} w_{kij}^m x_j\right)\right),$$

where $w_{ik}^m$ is a weight between the $j$th input neuron and the $k$th hidden neuron, $w_{kij}^m$ is a weight from the $k$th hidden neuron to the $i$th 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:

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

where $E[\cdot]$ is the expectation operator, $\{y_i(X) | i = 1, \ldots, c\}$ the outputs of the network, and $\{d_i | i = 1, \ldots, c\}$ 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 $y_i(X)$:

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

$E[d_i | X]$ is the conditional expectations of $d_i$, and $var[d_i | X]$ 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, $d_i = 1$ if the input $X$ belongs to class $\omega_i$.

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.

and 0 otherwise. Thus, the conditional expectations are the following:

$$E[d_i | X] = \sum_{j=1}^{c} d_j P(\omega_j | X)$$

$$= P(\omega_i | X)$$

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 posterior probabilities.

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

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

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

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

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 $B_i(c)$ be the number of classes ranked below the class $c$ by the $i$th network. Then, the Borda count for class $c$ is $B(c) = \sum_{i=1}^{n} B_i(c)$. The final decision is given by selecting the class label whose Borda count is the largest.
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 \( A \subseteq B \)

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 \( g_\lambda \)-fuzzy measures satisfying the following additional property: For all \( A, B \subseteq X \) and \( A \cap B = \emptyset \),

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

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 \( g_\lambda \)-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 a function \( h \) with respect to a fuzzy measure \( g \) is defined by

\[
h(x) \circ g(\cdot) = \max_{\alpha \in [0, 1]} \left[ \min_{E \in h} \left( \min_{x \in E} h(x), g(E) \right) \right] = \max_{\alpha \in [0, 1]} \left[ \min_{x \in h} \left( g(h_x) \right) \right]
\]

where \( h_x \) is the \( \alpha \) level set of \( h \),

\[
h_x = \{ x \mid h(x) \geq \alpha \}.
\]

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

\[
h(x) \circ g(\cdot) \geq \max_{i=1}^n c_i,
\]

where \( c_i \) 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, \( c \), with respect to a fuzzy measure \( g \) over \( Y \) can be computed by

\[
c = \max_{i=1}^n \left[ \min(h(y_i), g(A_i)) \right]
\]

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

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

\[
g(A_1) = g(\{ y_1 \}) = g_1^0
\]

\[
g(A_i) = g_1^i + g(A_{i-1}) + \lambda g'g(A_{i-1}), \quad \text{for } 1 \leq i \leq n.
\]

\( \lambda \) is given by solving the equation

\[
\lambda + 1 = \prod_{i=1}^n \left( 1 + \lambda g' \right)
\]

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 \( g_\lambda \)-fuzzy measure would only require the knowledge of the density function, where the \( i \)th density, \( g' \), 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, \( h(y_i) \), 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 \( \Omega = \{ \omega_1, \omega_2, \ldots, \omega_n \} \) be a set of classes of interest. Note that each \( \omega_k \) 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 \( \omega_k \), that is, \( h_k(y_i) \) is an indication of how certain we are in the classification of object \( A \) to be in class \( \omega_k \) using the network \( y_i \), where \( 1 \) indicates absolute certainty that the object \( A \) is really in class \( \omega_k \) and 0 implies absolute certainty that the object \( A \) is not in \( \omega_k \).

Now corresponding to each \( y_i \), the degree of importance, \( g' \), of how important \( y_i \) is in the recognition of the class \( \omega_k \) must be given. These densities can be subjectively assigned by an expert, or can be generated from training data. The \( g' \)'s define the fuzzy density mapping. Hence \( \lambda \) can be calculated using (12) and the \( g_\lambda \)-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

1. For each class \( \omega_k \) do
   1.1. For each neural network \( y_i \) do
      1.1.1. Calculate \( h_k(y_i) \);
      1.1.2. Determine \( g_k(\{ y_i \}) \);
   1.1. EndFor
   1.2. Compute the fuzzy integral for the class;
1. EndFor

2. Determine the final class;

3. EndAlgorithm
V. EXPERIMENTAL RESULTS

In the experiment, handwriting characters were inputted to the computer (SUN workstation) by an LCD tablet of Photon FOS-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 X 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.

Table I shows the recognition rates of the fuzzy integral for different densities functions (%) for the training data. The recognition rates of the fuzzy integral for the training data are shown in Table II. All λ values are very close to −1 because the sum of γ values is greater than 1. In this case, the degree of importance may be interpreted as a plausibility value. Banon showed that λ ≤ 0 if γ is a plausibility measure [16].

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 classified by NN3 and NN2, respectively. However, in the final evaluations they were correctly classified. In cases 5 and 7, 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 mismatch of 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 ε1, ε2, and ε3. If we take the minimum error to be ε1, the condition that the consensus (e.g., majority rule) error rate be less than the best individual error rate is

\begin{equation}
ε_1 > ε_1ε_2 + (1 - ε_1)ε_2 + ε_1(1 - ε_2)ε_3 + ε_1ε_2(1 - ε_3).
\end{equation}

While the other two cases satisfy the above condition, the lowercase letter case does not (ε1 = 0.302, but the right hand 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

<table>
<thead>
<tr>
<th>Data index</th>
<th>Actual class</th>
<th>Partial decision</th>
<th>Fuzzy integral decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>N1: 0.8956</td>
<td>N2: 0.8956</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>N1: 0.9956</td>
<td>N2: 0.9956</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>N1: 0.9956</td>
<td>N2: 0.9956</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>N1: 0.9956</td>
<td>N2: 0.9956</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>N1: 0.9956</td>
<td>N2: 0.9956</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>N1: 0.9956</td>
<td>N2: 0.9956</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>N1: 0.9956</td>
<td>N2: 0.9956</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>N1: 0.9956</td>
<td>N2: 0.9956</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>N1: 0.9956</td>
<td>N2: 0.9956</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>N1: 0.9956</td>
<td>N2: 0.9956</td>
</tr>
</tbody>
</table>

TABLE IV
THE RESULT OF RECOGNITION RATES (1)

<table>
<thead>
<tr>
<th>Subject</th>
<th>N1</th>
<th>N2</th>
<th>N3</th>
<th>Majority</th>
<th>Error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeral</td>
<td>73.2</td>
<td>69.8</td>
<td>70.8</td>
<td>74.0</td>
<td>77.4</td>
</tr>
<tr>
<td>Uppercase</td>
<td>59.0</td>
<td>69.8</td>
<td>74.0</td>
<td>77.4</td>
<td>77.6</td>
</tr>
<tr>
<td>Lowercase</td>
<td>59.0</td>
<td>69.8</td>
<td>74.0</td>
<td>77.4</td>
<td>77.6</td>
</tr>
</tbody>
</table>

TABLE V
THE RESULT OF RECOGNITION RATES (2)

<table>
<thead>
<tr>
<th>Subject</th>
<th>N1</th>
<th>N2</th>
<th>N3</th>
<th>Majority</th>
<th>Error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeral</td>
<td>82.6</td>
<td>81.2</td>
<td>81.0</td>
<td>84.9</td>
<td>86.8</td>
</tr>
<tr>
<td>Uppercase</td>
<td>73.9</td>
<td>71.8</td>
<td>72.1</td>
<td>74.0</td>
<td>79.2</td>
</tr>
<tr>
<td>Lowercase</td>
<td>73.9</td>
<td>71.8</td>
<td>72.1</td>
<td>74.0</td>
<td>79.2</td>
</tr>
</tbody>
</table>

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].

ACKNOWLEDGMENT

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

REFERENCES