Letters

Multiple Network Fusion Using Fuzzy Logic

Sung-Bae Cho and Jin H. Kim

Abstract— **Multiplayer feedforward networks trained by minimizing the mean squared error and by using a one of c teaching function yield network outputs that estimate posterior class probabilities. This provides a sound basis for combining the results from multiple networks to get more accurate classification. This paper presents a method for combining multiple networks based on fuzzy logic, especially the fuzzy integral. This method non–linearly combines objective evidence, in the form of a network output, with subjective evaluation of the importance of the individual neural networks. The experimental results with the recognition problem of on-line handwriting characters show that the performance of individual networks could be improved significantly.**

I. Introduction

Recently the concept of combining multiple networks has been actively exploited for developing highly reliable neural network systems [1]-[5]. One of the key issues of this approach is how to combine the results of the various networks to give the best estimate of the optimal result. There are a number of possible schemes for automatically optimizing the choice of individual networks and/or combining architectures.

A straightforward approach is to decompose the problem into manageable ones for several different subnetworks and combine them via a gating network that decides which of the subnetworks should be used for each case. Hampshire and Waibel [6] have described a system of this kind that can be used when the decomposition into subtasks is known prior to training, and Jacobs *et al.* [3] have also proposed a supervised learning procedure for systems composed of many separate networks, each of which learns to handle a subset of the complete set of training instances. The subnetworks are local in the sense that the weights in one expert are decoupled from the weights in other subnetworks. However, there is still some indirect coupling because if some other network changes its weights, it may cause the gating network to alter the responsibilities that get assigned to the subnetworks.

An alternative is to independently generate a number of networks for possible generalizers and utilized all of them for obtaining robust output. While a usual scheme chooses one best network from amongst the set of candidate networks based on a winner-takes-all strategy, this approach keeps multiple networks and runs them all with an appropriate collective decision strategy. This is different from the aforementioned "adaptive mixtures of local experts" [3], in the sense that here networks do not decompose the task, but learn globally the same task with different points of view. A general result from the previous works is that averaging separate networks improves generalization performance for the mean squared error [5]. If we have networks of different accuracy, however, it is obviously not good to take their simple average or simple voting.

To give a solution to the problem, this paper presents a fusion method that considers the difference of performance of each network in combining the networks, which is based on the notion of fuzzy logic, especially the fuzzy integral. This method combines the outputs of separate networks with importance of each network, which is subjectively assigned as the nature of fuzzy logic. Also, we demonstrate the superior performance of the presented method and compare with conventional averaging methods by thorough experiments. Although a serious theoretical investigation is beyond the scope of this paper, we will demonstrate the effectiveness of the method by experimental results on a difficult OCR problem.

II. Multiple Network Structure

Suppose that we have a two-layered 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 of classes, and H is an appropriately selected number.[1] The network is fully connected between adjacent layers. The outputs of the neural network are not just likelihoods or binary logical values near zero or one. Instead, they are estimates of Bayesian *a posteriori* probabilities of a classifier[2] [7].

The operation of this network can be thought of as a nonlinear decision-making process; Given an unknown input $X = (x_1, x_2, \cdots, x_T)$ and the class set $\Omega = (\omega_1, \omega_2, \cdots, \omega_c)$, each output neuron estimates the probability $P(\omega_i \mid X)$ of belonging to this class by

$$P(\omega_i \mid X) \approx f\left\{\sum_{k=1}^{H} w_{ik}^{om} f\left(\sum_{j=1}^{T} w_{kj}^{mi} x_j\right)\right\} \tag{1}$$

where w_{kj}^{mi} is a weight between the jth input neuron and the kth hidden neuron, w_{ik}^{m} 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.

This kind of network 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 will generalize poorly. Increasing the size and number of hidden layers most often does not lead to any improvements. 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.

The basic idea of the presented network scheme is to develop n independently trained neural networks with relevant features, and to classify a given input pattern by utilizing combination methods to decide the collective classification [1], [8] (Fig. 1). Because the

Manuscript received June 4, 1993; revised August 13, 1994. This work was supported in part by a grant from the Korea Science and Engineering Foundation (KOSEF).

S. B. Cho is with ATR Human Information Processing Research Laboratories, Kyoto 619-02, Japan.

J. H. Kim is with KAIST Center for Artificial Intelligence Research, Taejeon, 305-701, Republic of Korea.

IEEE Log Number 9408102.

[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 technique, such as Fahlman's *Cascade Correlation.*

[2] Note here that the outputs are not probabilities *per se*, but rather *estimates* of probabilities (hopefully good ones, but not necessarily so). Since the neural networks considered here were trained using a simple gradient technique, it is possible for the network to become struck in local minima, or for the network's limited size to preclude it from computing an accurate estimate.

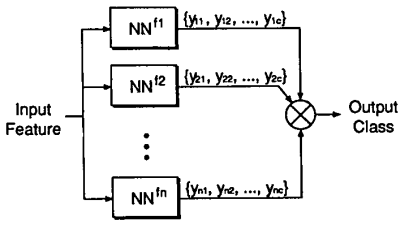

Fig. 1. The multiple network architecture with fusion method.

outputs of neural networks are estimates of Bayesian *a posteriori* probabilities as mentioned earlier, the classification of an input X is actually based on a set of real value measurements

$$P(\omega_i \mid X), \qquad 1 \le i \le c. \tag{2}$$

They represent the probabilities that X comes from each of the c classes under the condition X. In the multiple network scheme, each network k estimates by itself a set approximations of those true values as follows

$$P_k(\omega_i \mid X), \qquad 1 \le i \le c, \quad 1 \le k \le n. \tag{3}$$

One simple approach to combine the results on the same X by all n networks is to use the following average value as a new estimation of combined network

$$P(\omega_i \mid X) = \frac{1}{n}\sum_{k=1}^{n} P_k(\omega_i \mid X), \qquad 1 \le i \le c. \tag{4}$$

We can think of such a combined value as an averaged Bayes classifier. This estimation will be improved if we give the judge the ability to bias the outputs based on *a priori* knowledge about the reliability of the networks

$$P(\omega_i \mid X) = \sum_{k=1}^{n} r_k P_k(\omega_i \mid X), \qquad 1 \le i \le c \tag{5}$$

where

$$\sum_{k=1}^{n} r_k = 1. \tag{6}$$

Another alternative is to use the maximum value of $P_k(\omega_i \mid X)$ denoted by $P_m(\omega_i \mid X)$, to replace the correspondent average value. Since $\sum_{i=1}^{c} P_m(\omega_i \mid X) \neq 1$, we use the following normalized values as the new estimations

$$P(\omega_i \mid X) = \frac{P_m(\omega_i \mid X)}{\sum_{j=1}^{c} P_m(\omega_j \mid X)}, \qquad 1 \le i \le c. \tag{7}$$

III. Fuzzy Integral for Network Fusion

The fuzzy integral is a nonlinear functional that is defined with respect to a fuzzy measure, especially g_λ-fuzzy measure introduced by Sugeno [9]. The ability of the fuzzy integral to combine the results of multiple sources of information has been established in several previous works [10]–[12]. In the following, we shall introduce some definitions of it and present an effective method for combining the outputs of multiple networks with regard to subjectively defined importances of individual networks. For further details on the method, see the recent publication made by the authors [13].

TABLE I

Subset A	$g_{0.0305}(A)$
$\emptyset$	0
$\{y_1\}$	0.34
$\{y_2\}$	0.32
$\{y_3\}$	0.33
$\{y_1, y_2\}$	0.6633
$\{y_2, y_3\}$	0.6532
$\{y_1, y_3\}$	0.6734
$\{y_1, y_2, y_3\}$	1.0

Definition 1: A set function $g: 2^Y \rightarrow [0, 1]$ is called a fuzzy measure if

1) $g(\emptyset) = 0$, $g(Y) = 1$,
2) $g(A) \le g(B)$ if $A \subset B$,
3) If $\{A_i\}_{i=1}^{\infty}$ is an increasing sequence of measurable sets, then

$$\lim_{i\to\infty} g(A_i) = g\Big(\lim_{i\to\infty} A_i\Big).$$

From this definition, Sugeno introduced the g_λ-fuzzy measure satisfying the following additional property

$$g(A \cup B) = g(A) + g(B) + \lambda g(A)g(B)$$

for all $A, B \subset X$ and $A \cap B = \emptyset$, and for some $\lambda > -1$.

Example 1: Consider the following case of $Y = \{y_1, y_2, y_3\}$ together with density values $g^1 = 0.34$, $g^2 = 0.32$, and $g^3 = 0.33$. Using the equation (13) (which will be introduced below), the Sugeno measure g must have a parameter λ satisfying $0.0359\lambda^2 + 0.3266\lambda - 0.001 = 0$. The unique root greater than -1 for this equation is $\lambda = 0.0305$, which produces the following fuzzy measure on the power set of Y as found in Table I.

As expected, the subset of criteria $\{y_1, y_3\}$ is more important for confirming the hypothesis than either subsets $\{y_1, y_2\}$ or $\{y_2, y_3\}$.

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

$$\begin{aligned} h(y) \circ g(\cdot) &= \max_{E\subseteq Y}\Big[\min\Big(\min_{y\in E} h(y),\, g(E)\Big)\Big] \\ &= \max_{\alpha\in[0,1]}\left[\min\left(\alpha,\, g(F_\alpha)\right)\right] \end{aligned} \tag{8}$$

where

$$F_\alpha = \{y \mid h(y) \ge \alpha\}. \tag{9}$$

To get some intuition for the fuzzy integral we consider the following interpretation: $h(y)$ measures the degree to which the concept h is satisfied by y. The term $\min_{y\in E} h(y)$ measures the degree to which the concept h is satisfied by all the elements in E. Moreover, the value $g(E)$ is a measure of the degree to which the subset of objects E satisfies the concept measured by g. Then, the value obtained from comparing these two quantities in terms of the min operator indicates the degree to which E satisfies both the criteria of the measure g and $\min_{y\in E} h(y)$. Finally, the max operation takes the biggest of these terms. One can interpret the fuzzy integral as finding the maximal grade of agreement between the objective evidence and expectation.

The calculation of the fuzzy integral when Y is a finite set is easily given. Let $Y = \{y_1, y_2, \cdots, y_n\}$ be a finite set and let $h: Y \rightarrow [0, 1]$ be a function. Suppose $h(y_1) \ge h(y_2) \ge \cdots \ge h(y_n)$, (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

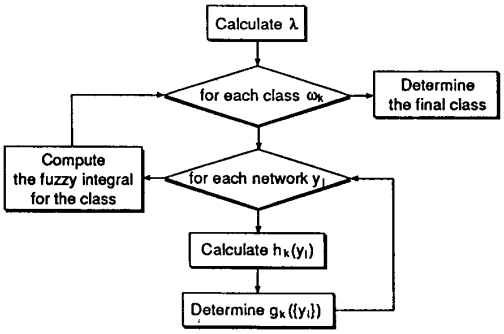

Fig. 2. Algorithm of the network fusion by the fuzzy integral.

$$e = \max_{i=1}^{n} [\min (h(y_i), g(A_i))] \tag{10}$$

where $A_i = \{y_i, y_2, \cdots, y_i\}$.

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

$$g(A_1) = g(\{y_1\}) = g^1 \tag{11}$$

$$g(A_i) = g^i + g(A_{i-1}) + \lambda g^i g(A_{i-1}), \qquad \text{for } 1 < i \leq n. \tag{12}$$

λ is given by solving the equation

$$\lambda + 1 = \prod_{i=1}^{n} (1 + \lambda g^i) \tag{13}$$

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 ith density, g^i, is interpreted as the degree of importance of the source y_i toward the final evaluation. In this definition, we can easily see that if g^1 increases then $g(A_i)$ also increases and hence the new fuzzy integral must be greater than or equal than the previous value. In general, if g^k increases for some $1 < k \leq n$, then $g(A_i)$, $i = 1, \cdots, k-1$, decreases and $g(A_i)$, $i > k$ increases. Therefore, if the function h or measure g increases then the integral increases.

Let $\Omega = \{\omega_1, \omega_2, \cdots, \omega_c\}$ be a set of classes of interest. Note that each ω_i may, in fact, be a set of classes by itself. Let $Y = \{y_1, y_2, \cdots, y_n\}$ 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 a 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.

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 induced from data set. The g^i's define the fuzzy density mapping. Hence λ is calculated using (13) and thereby the g_λ-fuzzy measure g is constructed. Now, using (10) to (13), the fuzzy integral can be calculated. Finally, the class ω_k with the largest integral value is chosen as the output class. Fig. 2 illustrates the details of how the consensus is formed.

Example 2: Using Example 1, how the consensus decision is performed by the fuzzy integral can now be described for a two class problem, which discriminates handwriting characters 6 and 4. Suppose that we obtain the network outputs for an input image as shown in Fig. 3, $h(y_1) = 0.6$, $h(y_2) = 0.7$, and $h(y_3) = 0.1$, for

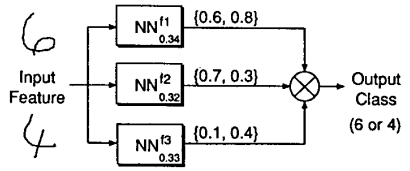

Fig. 3. A simple example of network outputs for a two class problem.

TABLE II

Class	$h(y_i)$	$g(A_i)$	$H(E)$	$\max[H(E)]$
1	0.7	$g(\{y_2\}) = g^2 = 0.32$	0.32	
	0.6	$g(\{y_2, y_1\}) = g^2 + g^1 + \lambda g^2 g^1 = 0.66$	0.6	√
	0.1	$g(\{y_2, y_1, y_3\}) = 1.0$	0.1	
2	0.8	$g(\{y_1\}) = g^1 = 0.34$	0.34	
	0.4	$g(\{y_1, y_3\}) = g^1 + g^3 + \lambda g^1 g^3 = 0.67$	0.4	√
	0.3	$g(\{y_1, y_3, y_2\}) = 1.0$	0.3	

class 1. For class 2, $h(y_1) = 0.8$, $h(y_2) = 0.3$, and $h(y_3) = 0.4$. Table II shows how the consensus is formed, where $H(E) = \min(h(y_i), g(A_i))$. Finally, the class 1 is selected as output. In the meantime, in case we use the weighted average instead of the fuzzy integral, the class 2 is chosen as the correct class because the class 2 yields 0.5 $(0.34 \times 0.8 + 0.32 \times 0.3 + 0.33 \times 0.4)$ whereas the class 1 produces 0.46 $(0.34 \times 0.6 + 0.32 \times 0.7 + 0.33 \times 0.1)$. This example shows how the minute differences of g^i conspire so as to dramatically change the performance compared to simple averaging.

IV. Experimental Results

To give an idea of practical application of the presented method to pattern recognition, a data set of handwriting characters was used as a source of both training and test samples. Handwriting characters were input to the computer (SUN workstation) by a Photron FIOS-6 440 LCD tablet, which samples at the rate of 80 dots per second. The tasks were to classify Arabic numerals, uppercase letters, and lowercase letters collected from 13 different writers. The writers were told to draw the numerals and letters into prepared square boxes to facilitate segmentation.

An input character consists of a set of strokes, each of which begins with a pen-down movement and ends with a pen-up movement. Several preprocessing algorithms were applied to successive data points within each stroke to reduce quantization noise and fluctuations due to the writer's pen motion. The processes used were wild point reduction, dot reduction, hook analysis, three point smoothing, peak preserving filtering, and N point normalization [14]. Data points, representing single characters, were resampled with a fixed number of regularly spaced points. Then, a sequence of preprocessed data points was approximated by a sequence of 8 directional straight-line segments—the chain code, as used by Freeman [14].

To evaluate the performance of the proposed method, we implemented three different networks, each of which is a two-layered neural network having a 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 each case, the network makes a decision based on its resolution. For example, NN_1 uses sparsely sampled inputs, and in doing so is able to overcome variations in input noise. NN_3, by comparison, uses a finer view of the input image. The selection of the features is largely adhoc and no attempt was made to find an optimal scheme, although this is an important issue in character recognition schemes. Our objective here is to evaluate and

TABLE III
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	83.0	80.4	78.0
2	0.1	0.3	0.2	83.2	80.6	77.6
3	0.2	0.1	0.3	83.6	80.4	77.6
4	0.2	0.3	0.1	83.6	81.0	75.6
5	0.3	0.1	0.2	85.0	81.6	78.0
6	0.3	0.2	0.1	84.8	80.8	77.2

TABLE IV
FUZZY DENSITIES AND THE CORRESPONDING λ's

Subject	g^1	g^2	g^3	λ
Numerals	0.3450	0.3349	0.3249	-0.0149
Uppercases	0.3447	0.3312	0.3240	0.0003
Lowercases	0.3370	0.3321	0.3312	-0.0009

compare different fusion methods through an example which has a certain complexity and practical significance.

Each of the three networks was trained by the EBP algorithm with 40 samples per class, validated with another 500 samples, and tested on 10 sets of additional samples collected from different 10 writers. The training process was stopped when the recognition rate over the validation set was optimized. This process and early stopping mechanism were adopted mainly for prevented networks from overtraining. The initial parameter values used for training were: Learning rate is 0.4 and momentum parameter is 0.6. From the result of Eaton *et al.* [15], we have selected a value of the learning rate to be decreased gradually to allow stable convergence of training. An input vector is classified as belonging to the output class associated with the highest output activation. Each of the following experiments consisted of 10 trails in which the different data were made from different writers.

First of all, the behavior of the fuzzy integral of a function h with respect to a g_λ-fuzzy measure, g is examined. Table III shows these results. Here, each case shows a set of fuzzy densities corresponding to three networks and the recognition rates of numerals, uppercase letters, and lowercase letters using the fuzzy integral on the three networks. Generating (13) and solving for λ gives $\lambda = 3.109$.

From this table we can expect that the recognition performance might be improved as modifying the g values. We assigned the fuzzy densities g^i, the degree of importance of each network, based on how good these networks performed on validation data, though there are a lot of possibilities to obtain these densities. We computed these values as follows

$$g^i = \frac{p_i}{\sum_j p_j} \cdot d\text{sum} \tag{14}$$

where p_i is the performance of network NN_i for the validation data and $d$sum is the desired sum of fuzzy densities. The real values of these densities with the corresponding λ are shown in Table IV.

Table V shows the simulation results obtained on numerals, uppercase letters, and lowercase letters, respectively. All results are averaged over ten different sets of the data. In Table V, NN_1 to NN_3 represent the three individual networks, and NN_{fi} a combined network of the three networks with the fuzzy integral. Although the network learned the training set almost perfectly in all three cases, the performances on the test sets are quite different.

A comparison of the proposed method with the conventional methods is given in Table VI. As can be seen, the overall classification rates for the fuzzy integral were higher than those for the average, the weighted average, and the maximum method. In addition, the integral evaluation need not sum to one, so that lack of evidence and negative evidence can be distinguished. It is also seen from Table VI that the mean recognition rate of the proposed method is higher than that of the other conventional methods. The following test, the paired t-test, can further support to determine whether the fuzzy integral method is superior to the conventional method or not.

For a given test problem, let f_i^a denote the solution at convergence for method a using test data i. To test whether methods a and b have

TABLE V
MEANS AND STANDARD DEVIATIONS OF RECOGNITION RATES OF THE THREE INDIVIDUAL NETWORKS AND THE FUZZY INTEGRAL (%)

	Numerals		Uppercases		Lowercases	
Networks	Mean	S.D.	Mean	S.D.	Mean	S.D.
NN_1	82.69	6.36	68.10	8.95	73.95	7.73
NN_2	81.25	7.16	68.63	9.14	71.80	8.86
NN_3	81.05	7.15	66.26	10.60	72.15	9.30
NN_{fi}	88.11	7.14	76.16	9.85	80.35	7.24

TABLE VI
COMPARISON OF THE FUZZY INTEGRAL WITH OTHER FUSION METHODS ON THE RECOGNITION RATES (%)

Numerals

Data Set	Average	Weighted Average	Maximum	Fuzzy integral
1	92.2	92.22	90.56	96.11
2	86.67	86.67	85.56	86.67
3	83.89	83.89	82.78	85.00
4	94.44	95.00	92.22	95.56
5	90.00	90.00	90.56	91.67
6	86.11	86.11	84.44	88.33
7	73.33	73.33	71.11	73.33
8	82.22	82.78	82.78	83.33
9	83.89	83.89	82.22	85.00
10	96.67	96.67	95.00	96.11
Mean	86.94	87.06	85.72	88.11

Uppercase Letters

Data Set	Average	Weighted Average	Maximum	Fuzzy integral
1	86.34	87.37	82.11	88.42
2	82.11	82.63	78.95	82.11
3	76.84	77.37	74.21	78.95
4	85.26	83.68	83.16	87.37
5	78.42	78.42	75.79	78.42
6	74.74	75.26	68.95	76.32
7	58.42	60.00	55.26	60.53
8	78.42	78.42	71.58	78.42
9	71.05	71.58	66.32	71.58
10	61.05	61.05	57.37	59.47
Mean	75.27	75.58	71.37	76.16

Uppercase Letters

Data Set	Average	Weighted Average	Maximum	Fuzzy integral
1	82.00	82.00	82.00	85.00
2	84.00	83.50	80.00	84.50
3	89.50	90.00	85.50	88.50
4	85.50	85.50	88.00	87.00
5	75.50	76.50	71.00	79.00
6	59.50	61.00	60.50	64.00
7	79.50	80.50	79.50	80.50
8	80.00	80.50	77.50	83.50
9	71.50	72.50	71.50	75.00
10	75.50	75.50	73.50	76.50
Mean	78.25	78.75	76.90	80.35

TABLE VII
THE PARIED t-TEST (DEGREE OF FREEDOM = 9, $t_{0.05} = 1.833$, $t_{0.025} = 2.262$, $t_{0.01} = 2.821$. "A" STANDS FOR AVERAGE METHOD, AND "WA" FOR WEIGHTED AVERAGE. "YES" INDICATES THAT THE HYPOTHESIS IS REJECTED FOR THE TASK AT THE ASSOCIATED LEVEL OF SIGNIFICANCE)

Task	t	Significance Level		
		5%	2.5%	1%
Numerals (A)	-2.908	Yes	Yes	Yes
Numerals (WA)	-2.575	Yes	Yes	No
Uppercase (A)	-2.193	Yes	No	No
Uppercase (WA)	-1.300	No	No	No
Lowercase (A)	-3.806	Yes	Yes	Yes
Lowercase (WA)	-3.361	Yes	Yes	Yes

the same mean solution value, we compute the following statistic

$$t = \frac{\sqrt{n}\bar{x}}{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}} \tag{15}$$

where $n = 10$, $x_i = f_i^a - f_i^b$, and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$. (In this case the method b is of the fuzzy integral.) From this value we can reject the null hypothesis that $H_0 : \bar{x} \leq 0$ in favor of the alternative that $\bar{x} > 0$ with significance level α, where $\alpha = \Phi(t)$ and $\Phi(t)$ can be obtained from the table of percentage points of the t-distribution.

Since it follows t-distribution, an α point can be computed as the threshold t_α, where α could be 95, 99.95 or 99.9%. Then, if

$$|t| > t_\alpha \tag{16}$$

the null hypothesis is rejected at a 100%–α level of significance, i.e., the fuzzy integral method is superior to the conventional method. Otherwise, the null hypothesis is accepted, i.e., we cannot say the fuzzy integral method improves the performance significantly.

Table VII shows the results of the test with $n = 10$ for all three tasks. In this comparison, f_i^b is of the fuzzy integral, and f_i^a of the average method as mentioned in (4) or the weighted average method in (5). These methods were chosen for comparisons because they had produced the best results among the conventional methods mentioned in this paper. In this comparison, the degree of freedom is $(n - 1) = 9$, and the threshold t_α with $\alpha = 95$, 97.25, and 99% is 1.833, 2.262 and 2.821, respectively. It is seen from Table VII that all the values of t, except for the weighted average of the uppercase letter task, are greater than t_α with $\alpha = 95\%$. Therefore, for all the cases except that, "no-improvement" hypothesis is rejected at a 5% level of significance. Similarly, other cases can be tested. This is strong evidence that the proposed method is superior to the conventional methods.

V. CONCLUSION

This paper has presented the multiple network scheme and proposed a fusion method based on the fuzzy integral. One of the important advantages of the method is that not only are the classification results combined, but that the relative importance of the different networks is also considered. The experimental results for classifying a large set of on-line handwriting characters show that it improves the generalization capability significantly. It is relevant to note here that the present paper has dealt with the comparison of the methods within the same framework of neural network architecture, so called the multiple neural networks. However, it might also be possible to compare with any other neural network architectures as long as there is a fair way to measure the performance, which remains for further study.

REFERENCES

[1] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, pp. 993–1001, 1990.

[2] C. Scofield, L. Kenton, and J. Chang, "Multiple neural net architectures for character recognition," in *Proc. Compcon*, pp. 487–491, San Francisco, CA: IEEE Computer Society Press, 1991.

[3] 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.

[4] D. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, pp. 241–259, 1992.

[5] M. P. Perrone and L. N. Cooper, "When networks disagree: Ensemble methods for hybrid neural networks," *Neural Networks for Speech and Image Processing*, R. J. Mammone, Ed., Chapman-Hill, London, 1993.

[6] J. B. Hampshire II and A. Waibel, "Connectionist architectures for multi-speaker phoneme recognition," *Advances in Neural Information Processing Systems*, vol. 2, pp. 203–210, 1990.

[7] M. D. Richard and R. P. Lippmann, "Neural network classifiers estimate Bayesian *a posteriori* probabilities," *Neural Computation*, vol. 3, pp. 461–483, 1991.

[8] S. Shlien, "Multiple binary decision tree classifiers," *Pattern Recognition*, vol. 23, pp. 757–763, 1990.

[9] M. Sugeno, "Fuzzy measures and fuzzy integrals: a survey," *Fuzzy Automata and Decision Processes*. Amsterdam: North Holland, 1977, pp. 89–102.

[10] K. Leszeynski, P. Penczek, and W. Grochulskki, "Sugeno's fuzzy measures and fuzzy clustering," *Fuzzy Sets and Systems*, vol. 15, pp. 147–158, 1985.

[11] 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.

[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] S.-B. Cho and J. H. Kim, "Combining multiple neural networks by fuzzy integral for robust classification," *IEEE Trans. Syst., Man, Cybern*, vol. 25, no. 1, 1995.

[14] C. C. Tappert, C. Y. Suen, and T. Wakahara, "The state of the art in on-line handwriting recognition," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 12, no. 8, pp. 787–808, 1990.

[15] H. A. C. Eaton and T. L. Olivier, "Learning coefficient dependence on training set size," *Neural Networks*, vol. 5, pp. 238–288, 1992.

Sensitivity of Equilibrium Points in Continuous-Time Hopfield's Network

Renzo Perfetti

***Abstract*—The sensitivity of continuous-time Hopfield neural network is investigated. The relative sensitivity of hyperbolic equilibrium points, with respect to changes in the interconnections, is computed. Then it is shown that the minimum sensitivity of equilibria corresponds to the minimum scattering of weights about their mean value. This permits a selection among the many available synthesis methods for associative memories.**

Manuscript received June 4, 1993; revised September 10, 1993.

The author was with the Info-Com Dept., Università "La Sapienza," Rome, Italy and is now with the Istituto di Elettronica, Università di Perugia, 06100 Perugia, Italy.

IEEE Log Number 9213985.