JOURNAL OF ARTIFICIAL NEURAL NETWORKS, 1(1), 77–100 (1994)

Associative Switch for Combining Multiple Classifiers

Lei Xu

Peking University
Beijing, PRC

Adam Krzyżak
Ching Y. Suen

Concordia University
Montreal, Canada

The problem of combining the outputs of several classifiers is encountered in various applications of pattern recognition and has recently gained a lot of interest. In this paper, a neural net model, called associative switch, based on a new combination principle, is proposed for solving the problem. This switch consists of: (1) a number of knobs which gate the output channels of individual classifiers, and (2) a multilayer perceptron neural net trained by a backpropagation technique with a modified error criterion. When an unlabeled pattern is input to each individual classifier, it also enters a neural net for associatively recalling a code which controls the knobs to decide whether the output of each classifier could pass through as the final result. The problem of appropriately training the net to fulfill the associative control task is further addressed, and advantages of the modified error criterion are analyzed. Furthermore, this associative switch is used to tackle the problem of combining multiple classifiers for recognizing totally unconstrained handwritten numerals. The experiments show that the associative switch can improve the results of individual classifiers considerably.

multiple classifiers, backpropagation, Kullback-Leibler measure

The authors wish to thank Christine Nadal for providing the classification results of the four individual classifiers e_1, e_2, e_3, e_4 which form the basis of the experiments performed in Section 4. This research was supported by the Natural Sciences and Engineering Research Council of Canada, and the National Networks of Centers of Excellence program of Canada.

Correspondence and requests for reprints should be sent to Adam Krzyżak, Department of Computer Science, Concordia University, 1455 De Maisonneuve Blvd. West, Montreal, Canada H3G18.

1 INTRODUCTION

Recently, the combination of multiple classifiers has been regarded as a new direction for the development of highly reliable character recognition systems [1]. Preliminary results indicate that combination of several complementary classifiers leads to classifiers with improved performance [1, 2, 3, 4].

In our recent paper [5], we argued that the combination of multiple classifiers is a general problem which is important not only for character recognition but also for various applications of pattern recognition. There are at least three reasons justifying that claim.

1. In almost every application area there are many classification algorithms available based on various theories and methodologies. Each of these classifiers could attain a certain degree of success, but maybe none of them is totally perfect, or at least not as good as expected in practical applications. So there is a need to study the methodology of integrating the outputs of different classification algorithms so that better results can be obtained.
2. For specific recognition problems, usually many types of features can be used to represent and recognize patterns. These features can be represented in many different forms. It is very difficult to lump them together into one single classifier to make decisions. As a result, many classifiers are needed to handle different types of features. Again, the problem of combining multiple classifiers arises naturally.
3. Even for special application problems with only one feature type, it may be a good idea to divide the high-dimensional feature vector into several vectors of lower dimensions and input them to several classifiers, since it is well known that high-dimensional vectors will not only increase computational complexity but will also produce implementation and accuracy problems.

We recently summarized [5] problems of combining multiclassifiers into three categories according to the levels of information produced by various classifiers. Through the use of voting principle, Bayesian Theory, and Dempster–Shafer Theory, four approaches have been developed to tackle these problems. Application of these approaches to the problem of recognizing totally unconstrained handwritten numerals showed that combination of classifiers is significantly better than any individual classifier.

Since the recent renaissance of research in neural networks, many new approaches appeared, which can fulfill the task of pattern classification. The most popular ones among them are backpropagation [6, 7], LVQ (1,2,3) [8, 9], ART (1, 2, 3) [10, 11, 12], and Madaline (I, II, III) [13]. Due to many favorable characteristics of neural nets, especially the massive parallelism and their adaptive learning ability, these neural classifiers have recently been widely used to tackle many pattern recognition problems. The occurrence of various versions of this new kind of classifiers has created the need for an in-depth study on methods of integrating multiple classifiers. First, the different types of neural-net classifiers provide different classification results which may also complement each other, and an appropriate combination may produce a better result. Second, at present, almost all neural-net classifiers require the input data to be expressed by a vector in Euclidean space (i.e., they have the function similar to the conventional statistical classifiers). From a practical point of view, it may be helpful to combine the performances of neural-net classifiers and some conventional classifiers, especially those based on structural or syntactic methods.

In this paper, we will explore the possibility of using the neural-net approach to combine several classifiers. A new combination principle different from those given in paper [5] is proposed, and a novel technique called associative switch is developed to realize the principle. The switch is controlled by a feed-forward neural net trained by the backpropagation technique with a modified error criterion. When an unlabeled pattern is input to each individual classifier, it also goes to the neural net for associatively recalling a code which controls the switch to decide whether the result of each classifier could pass through as a final result. In Section 2, we will propose the new combination principle and explain the basic model of the associative switch, and then in Section 3 discuss how to train the switch to attain the right control ability, and analyze the advantage of the modified error criterion used here. In Section 4, we will show the results of applying this new approach to a problem of combining multiple classifiers for recognizing totally unconstrained handwritten numerals. Finally, we will conclude this paper in Section 5.

Recently, in the neural network literature, there has been an increasing interest in the study of combining multinetworks or modular or experts so that the global approximation task can be decomposed to several experts and fulfilled by these modular with better performance. A good review of such results is given in the recent papers [14, 15]. Our associative switch is different from the approaches of these papers in several aspects. In our switch, experts can be either neural-net classifiers, other conventional classifiers, or a mixture of different kinds of classifiers. Before training the switch network, these classifiers already exist and have been trained, and each of them is already capable to do classification. The switch is used in the process of classification to choose the classification result of a specific classifier for a specific input pattern. During the training of the switch, for each pattern, the training signal is the indication of the desired classifier (i.e., one which can produce the right classification of the pattern). In the approaches given in the papers [14, 15], experts are only neural nets. The final output of the whole net is the weighted combination of the outputs of each individual expert. Both the switch network (called gating network) and all the expert networks are trained simultaneously. For each pattern, the training signal is the desired output value (e.g., in classification, it is the right class label of the pattern). The purpose of our work is to build a switch which can select among the existing classifiers; while the goal of the approaches given in papers [14, 15] is to build a switch and all the expert networks at the same time so that the whole task can be decomposed appropriately among expert nets. Furthermore, our application here is also different from those considered in the papers [14, 15].

2 ASSOCIATIVE SWITCH TECHNIQUE FOR NEW COMBINATION PRINCIPLE

2.1 A New Principle for Combining Multiclassifiers

As pointed out in our paper [5], the problems of combining multiple classifiers could be classified into three types. Among them, Type 1 is one of the most general ones since it requires no extra information from each individual classifier except a class label assigned to the current input pattern and any classifier could provide such information. In this paper, we will concentrate on the problem of this type. A clearer description of the problem is given as follows.

Assume that there are K individual classifiers (or experts as called in the paper [1] e_k, $k = 1, \dots, K$). For an unlabeled input pattern x (expressed by a vector in the Euclidean space) each e_k will give x a label $j_k \in \Lambda \cup \{\phi\}$ with each label of $\Lambda = \{1, 2, \dots M\}$ representing a pattern class and ϕ denoting the rejection. Regardless of the internal structure of a classifier and regardless of theory and methodology it based on, we may simply regard e_k as a functional box which receives an input x and outputs a label j_k, in short denoted by $e_k(x) = j_k$. Then, the problem is to build a big functional box $E(x) = j$, $j \in \Lambda \cup \{\phi\}$ based on $e_k(x) = j_k$, $k = 1, \dots, K$, such that a better classification performance could be obtained.

In our paper [5], several methods have been proposed for solving this problem. Those methods are based on two general principles. One is committee voting, and the other is evidence gathering for uncertainty reasoning. The two principles share one common point of trying to assemble or synthesize the outcomes of all the individuals as the final output. In this paper, we try to combine the results of multiple classifiers based on a new principle which is totally different from the above two principles. Simply speaking, the new principle is to select an expert to be totally responsible in the area of his expertise (i.e., for an input pattern x), we try to see if one of K individual classifiers is an expert at classifying x correctly, if not, we reject x; if there is one, we use its classification as the final outcome, if there are more than one, we could use the output of any one of them. It is quite clear that the key task for implementing the principle is how to correctly select an expert for each input pattern. If the task could be fulfilled well, the combined classification performance will of course be better than the performance of any one individual classifier since the combined $E(x)$ will give a right result only when one of K individuals could give the right result.

2.2 Associative Switch Model

We propose a technique called associative switch to fulfil the task of choosing an expert for each input. The model of the switch is shown in Figure 1(a).

It consists of K knobs sw_k, $k = 1, \dots, K$ with each sw_k installed on the output channel of classifier e_k to decide whether or not it is selected as the expert for the

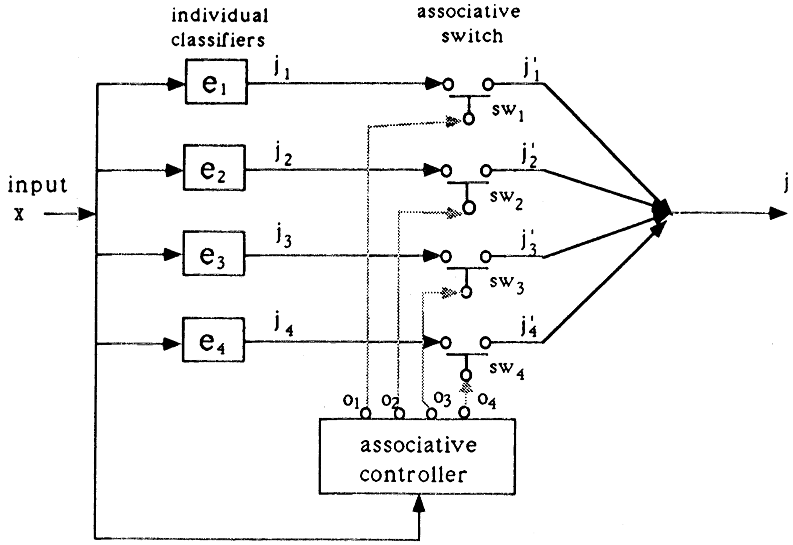


Figure 1(a). The basic model of associative switch for combining multiple classifiers.

present input (i.e., to let its output pass through). So, the output of each knob is given by

$$j'_k = \begin{cases} j_k, & \text{when } sw_k = \text{``}on\text{''} \\ \phi & \text{when } sw_k = \text{``}off.\text{''} \end{cases} \tag{1a}$$

K knobs are controlled by the output code $O = [o_1, \ldots, o_K]$, $0 \leq o_k \leq 1$, $k = 1, \ldots K$ of an associative controller in the following way

$$sw_k = \begin{cases} \text{``}on\text{''} & \text{when } o_k \geq o_t, \\ \text{``}off\text{''}, & \text{otherwise} \end{cases} \tag{1b}$$

where $0.5 < o_t \leq 1$ is a predefined threshold. When an unlabeled pattern x is input to individual classifiers e_k, $k = 1, \ldots, K$, it is also input to the associative controller for recalling code O which controls K knobs to either select one of the outputs of K classifiers as the final label j (when there is at least an expert for the present input) or block all the output channels of individual classifiers and assign ϕ to j (when there is no expert for the input). The associative controller could be any existing neural net of heteroassociative memory type [8]. In this paper, we use one-hidden layer feed-forward net architecture for our associative controller. The key requirement for the controller net is to map each input x into code O which satisfies a feasible condition defined by the following points:

1. For one input, the code O results in either all K knobs being "off" or one and only one knob being "on" (i.e., either no expert is found or one and only one expert is selected).

2. The "on" knob should connect the output channel of one of classifiers which labels x correctly (i.e., the selected expert should be a real expert).

Such mapping ability of the controller net is obtained through a learning process on a training pattern set in which class label of every pattern is known a priori. During the learning process, for each input pattern x, the desired mapping between x and O is designed from the known class label of x and the output $e_k(x)$, $k = 1, \ldots, K$, and then the internal connection weights of the controller net are modified to adapt to the desired mapping relation. In the ideal case, when the desired mapping relation for the training set has been completely learned by the controller net, and if the samples of the training set have the total representation such that the desired mapping relation on the training set is the same as on all the other patterns we are going to deal with, then each input x will associatively produce the right code O which lets the knobs correctly conduct the task of expert selection. In this case, the final output of the whole model could be expressed as

$$j = \begin{cases} j'_k, & \text{if } j'_k \neq \phi, \\ \phi, & \text{otherwise} \end{cases} \tag{1c}$$

2.3 Some Further Remarks

In general, it is very difficult for the controller net to completely learn the desired mapping for each input pattern we are going to deal with. Even when we assume that after training each pattern of the training set could recall a desired code designed during learning, some patterns encountered in the testing phase (or recall phase) may need the mapping relations which have not been met or learned before. Such difficulty can be partly overcome by the associative or generalization ability of a neural net. Namely, when an unknown pattern reaches a net trained on a set which does not contain the pattern, as long as the pattern is not too different from those patterns in the training set (how different patterns can neural net tolerate depends on its generalization ability), the net could still give a reasonable output code. Another measure to tackle the difficulty is given in Equation (1b), where the knobs are controlled not by the exact desired code but by the codes within an interval of tolerance. For these reasons, we could expect that the controller net could learn a good approximation of the desired mapping, thus it can select the outcomes of individual classifiers very well even when there may occasionally occur some wrong selections.

Due to the interval of tolerance in Equation (1b) as well as the possibility of wrong selection by the controller net during the testing phase, it may occasionally happen that more than one knob is "on" and the messages through these "on" knobs conflict. There are several remedies in such cases. The simplest one is to reject the present input (i.e., to turn off all the knobs). One could also regard that these cases form a combination problem with a small degree of complexity and treat them by some combining methods again (e.g., by appending the present associative switch with another one which has a controller net of smaller complexity). For simplicity, in this paper, we add a simple conflict resolver as shown in Figure 1(b). Let J_c be a subset consisting of all $j'_k \neq \phi$. Upon modifying Equation (1b), the output of the conflict resolver is given by

$$j = \begin{cases} j'_k, & \text{if } j'_k \in J_c \text{ and } \overline{\exists} j'_{k_1} \neq j'_k \text{ such that } r(j'_{k_1}) \geq r(j'_k), \\ \phi, & \text{otherwise} \end{cases} \tag{1d}$$

where $r(j_k)$ denotes the number of those elements which take value j_k in J_c.

Furthermore, in Figure 1(b), we have also supplemented one other element of Figure 1(a) (i.e., the input to the associative controller is $M(x)$ instead of x). The reason is that besides directly using the input pattern x to recall the controller net one could also use some mapping or coding $M(x)$ of x as the controller's input as long as $M(x)$ is expressed as a numeric vector. One way to obtain such $M(x)$ may be dimension reduction transform or feature selection. Another possibility is to let $M(x)$ be just the label vector $J(x) = [j_i, \ldots, j_K]$ (where we let $j_k = -1$ replace the nonnumeric notation $j_k = \phi$) consisting of the outputs of all individual classifiers. When the dimension of the input x is quite high, a large neural net is required as the associative controller, which costs a lot of storage and computing time. In such case, $J(x)$ may be a good alternative to x since the number of individual classifiers is usually much smaller than the dimension of x.

Before closing this section, we would like to make it clear that the above associative controller is different from the traditional logical code translator or decoder. For a logical decoder, although its output could satisfy the two points of the feasible condition mentioned earlier, its input cannot directly be the input pattern x itself, but a logical code of x. When recalling, this input logical code

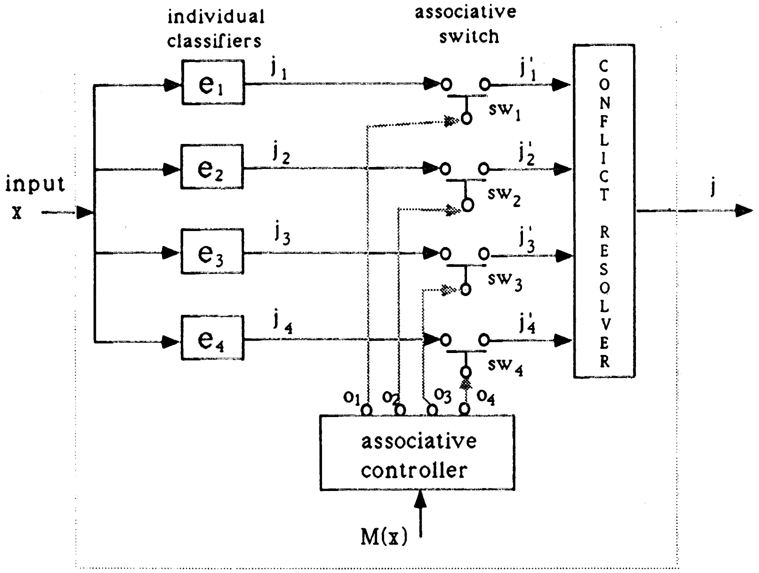


Figure 1(b). The model of associative switch with a conflict resolver.

should be exactly right, otherwise a small error will result in a totally different output code and will thus produce a completely wrong control on sw_k, $k = 1, \ldots, K$. In contrast, the associative controller, being a neural net, permits its input to be either directly the input pattern x itself or its mapping $M(x)$ (including also logical coding). An error bearing or distorted input or even a partial input will still call out the right output code or a code which is near the right one (i.e., the controller has associative recalling ability and is robust enough to tolerate noises and distortions).

3 THE CONTROLLER NET AND ITS LEARNING

3.1 Backpropagation Method

The backpropagation method is a learning technique proposed for training the multiple layer perceptron neural net [6, 7]. Here, by taking the one-hidden layer feed-forward net as example, we briefly summarize the method as follows.

The one-hidden layer net is shown in Figure 2, the input layer consists of n units each of which corresponds to a component of input pattern vector $x = [x_1, \ldots, x_n]$. These input units are fully connected to n_h units of the hidden layer. Again, all the hidden units are fully connected to K units with the output o_k, $k = 1, \ldots, K$.

For each unit, its input is usually given by

$$y_j^{(r)} = \sum_{i=1}^{n^{(r-1)}} w_{ij}^{(r)} o_j^{(r-1)} + \theta_j^{(r)} \tag{2a}$$

where y_j is called the potential of unit j. It makes the unit produce an output o_j through an activation function given by

$$o_j^{(r)} = \frac{1}{1 + e^{-ay_j^{(r)}}} \tag{2b}$$

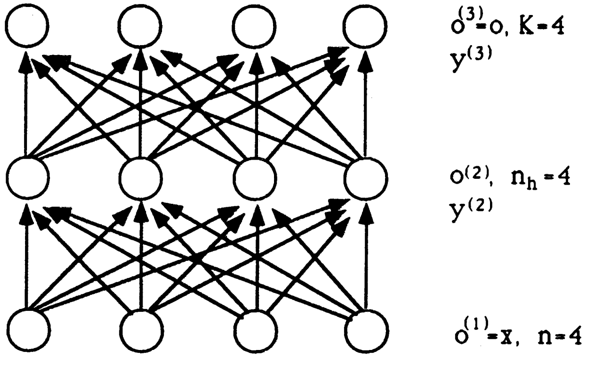


Figure 2. A one hidden layer forward net as the associative controller.

where $r = 1, 2$ denotes the hidden and output layers respectively with $o_k = o_k^{(2)}$, $o_i^{(0)} = x_i$ and $n^{(0)} = n$, $n^{(1)} = n_h$, $n^{(2)} = K$. $w_{ij}^{(r)}$ is the weight of the connection from unit i in the layer $r - 1$ to the unit j in the r layer, $\theta_j^{(r)}$ is a variable bias with similar function to a threshold. $a > 0$ affects the steepness of the activation function. High a values give a steplike curve and lower values give a smooth curve.

To build the desired input-output mapping, the net is trained on a training set in which the desired output to each input pattern is known a priori. When an input x reaches the first layer, it is passed forward through Equations (2a) and (2b) to generate an output $O(x, W, \theta) = [o_1(x, W, \theta), \ldots, o_K(x, W, \theta)]$ depending on the present input x, connection weights W and bias θ. The error between the current output and the desired one $O^d = [o_1^d, \ldots, o_K^d]$ is calculated by

$$J_2 = \frac{1}{2} \sum_{k=1}^{K} (o_k^d(x) - o_k(x, W, \theta))^2. \tag{3a}$$

Its derivatives $\frac{\partial J_2}{\partial w_{ij}^{(r)}}, \frac{\partial J_2}{\partial \theta_j^{(r)}}$ with respective to each of all the connections $w_{ij}^{(r)}$ and biases $\theta_j^{(r)}$ are computed to modify these parameters in the gradient descent manner with a learning rate α, that is,

$$\Delta w_{ij}^{(r)} = -\alpha \frac{\partial J_2}{\partial w_{ij}^{(r)}}, \Delta \theta_j^{(r)} = -\alpha \frac{\partial J_2}{\partial \theta_j^{(r)}}. \tag{3b}$$

Each time, an input pattern is presented to the net randomly (or in some specific order) from the training set, the above procedure is repeated once, until a convergence is reached or the error is reduced below a predefined tolerance level. This way of training is called the online or adaptive type. An alternative way, called batch type, is also often used. It calculates the error J_2 after a batch of input patterns $x_i, \ldots, x_{N_b}$ are input to the net by the following Equation (3c) instead of Equation (3a)

$$J_2 = \frac{1}{2N_b} \sum_{p=1}^{Nb} \sum_{k=1}^{K} (o_k^d(x_p) - o_k(x_p, W, \theta))^2 \tag{3c}$$

and then reduces the error by Equation (3b).

The key point of the above learning is the calculation of all the derivatives $\frac{\partial J_2}{\partial w_{ij}^{(r)}}$ and $\frac{\partial J_2}{\partial \theta_j^{(r)}}$. The task is completed by the backpropagation technique [6, 7]. Based on the chain rule, it starts at the output layer and goes down to the input layer for calculating derivatives as follows:

$$\frac{\partial J_2}{\partial w_{ij}^{(r)}} = \frac{\partial J_2}{\partial y_j^{(r)}} o_j^{(r-1)}, \frac{\partial J_2}{\partial \theta_j^{(r)}} = \frac{\partial J_2}{\partial y_j^{(r)}} \tag{4a}$$

for $r = 2$

$$\frac{\partial J_2}{\partial y_j^{(r)}} = \frac{\partial J_2}{\partial o_j^{(r)}} \frac{\partial o_j^{(r)}}{\partial y_j^{(r)}} = -a[1 - o_j(x, W, \theta)]o_j(x, W, \theta)[o_j^d - o_j(x, W, \theta)], \tag{4b}$$

and for $r = 1$

$$\frac{\partial J_2}{\partial y_j^{(r)}} = \sum_{j=1}^{n(r+1)} \frac{\partial J_2}{y_j^{(r+1)}} \frac{\partial y_j^{(r+1)}}{\partial o_j^{(r)}} \frac{\partial o_j^{(r)}}{\partial y_j^{(r)}} = \sum_{j=1}^{n(r+1)} a(1 - o_j^{(r)})o_j^{(r)} w_{ij}^{(r+1)} \frac{\partial J_2}{\partial y_j^{(r+1)}}, \tag{4c}$$

where in Equation (4c), $\frac{\partial J_2}{\partial y_j^{(r+1)}} = \frac{\partial J_2}{\partial y_j^{(2)}}$ is already available after the calculation of Equation (4b) (i.e., by Equation (4c)), the required derivatives of the lower layer could be computed based on those of the upper layers through backward propagation.

3.2 A Modified Error Criterion and Its Advantages

Although the least squares error criterion Equation (3a) or Equation (3c) is widely used in neural net literature, we propose to replace it by another criterion given by

$$J_{KL} = -\sum_{k=1}^{K} \{[1 - o_k^d(x)] \ln [1 - o_k(x, W, \theta)] + o_k^d(x) \ln o_k(x, W, \theta)\} \tag{5a}$$

or

$$J_{KL} = -\sum_{p=1}^{Nb} \sum_{k=1}^{K} \{[1 - o_k^d(x_p)] \ln [1 - o_k(x_p, W, \theta)] + o_k^d(x_p) \ln o_k(x_p, W, \theta)\}. \tag{5b}$$

Under this criterion, all the formulas for backpropagation given in Section 3.1 above are still valid by replacing J_2 by J_{KL} except that Equation (4b) should be changed into the following

$$\frac{\partial J_{KL}}{\partial y_j^{(r)}} = \frac{\partial J_{KL}}{\partial o_j^{(r)}} \frac{\partial o_j^{(r)}}{\partial y_j^{(r)}} = -[o_j^d - o_j(x, W, \theta)], \qquad \textit{for } r = 2 \tag{5c}$$

(i.e., a factor of $[1 - o_j(x, W, \theta)]o_j(x, W, \theta)$ in Equation (4b) is eliminated).

Krzyżak et al. have shown [16] that this elimination can partly solve the problem of local minima encountered in the least squares criterion. They applied the modified criterion to handwritten character recognition problem with a significant speed-up in the training process. In Hinton's paper [17], by imagining that a real-valued o_k is randomly converted into a binary value with the probability o_k being one, and by interpreting o_j^d as the corresponding desired probability for the output unit j being one, the minimization of J_{KL} in Equation (5a) or (5b) is interpreted as minimizing a cross-entropy. In the following, we will explore new properties of this criterion.

In a classification problem, for each input x, the desired output is a binary vector which has only one component being one indicating the class that x belongs to

$$o_j^d(x) = \begin{cases} 1 & \text{if } x \in \omega_j, \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where ω_j expresses class j.

Let $P(\omega_j)$ denote a priori probability of class ω_j and $P(\omega_j|x)$ is a posteriori one given x. We can prove the following theorem which shows favorable features of criterion Equation (5).

Theorem. The minimization of the expectation $E(J_{KL})$ is equivalent to minimizing $E_{KL} + \bar{E}_{KL}$ with respect to W and θ, where

$$E_{KL} = -\sum_{j=1}^{C} P(\omega_j) \int_x P(x|\omega_j) \ln \frac{P(\omega_j|x)}{P(\omega_j|x, W, \theta)}\, dx \tag{7a}$$

$$\bar{E}_{KL} = -\sum_{j=1}^{C} P(\bar{\omega}_j) \int_x P(x|\bar{\omega}_j) \ln \frac{P(\bar{\omega}_j|x)}{P(\bar{\omega}_j|x, W, \theta)}\, dx \tag{7b}$$

with

$$P(\omega_j|x, W, \theta) = o_j(x, W, \theta),\ P(\bar{\omega}_j|x, W, \theta) = 1 - o_j(x, W, \theta) \tag{7c}$$

where $\bar{\omega}_j$ denotes the complementary of ω_j (i.e., $x \in \bar{\omega}_j$ is equivalent to $x \notin \omega_j$).

Proof. By taking expectation on the both sides of Equation 5(a), we get

$$\begin{aligned} E(J_{KL}) &= E\left\{-\sum_{j=1}^{K} [1 - o_j^d(x)] \ln [1 - o_j(x, W, \theta)]\right\} + E\left\{-\sum_{j=1}^{K} o_j^d(x) \ln o_j(x, W, \theta)\right\} \\ &= \sum_{j=1}^{K} - E\left\{[1 - o_j^d(x)] \ln [1 - o_j(x, W, \theta)] + \sum_{j=1}^{K} - E[o_j^d(x) \ln o_j(x, W, \theta)]\right\} \end{aligned}$$

Since $o_j^d(x) = 1$ for $x \in \omega_j$ and $o_j^d(x) = 0$ for $x \notin \omega_j$, we further have

$$\begin{aligned} E\{[1 - o_j^d(x)] \ln [1 - o_j(x, W, \theta)]\} &= P(\bar{\omega}_j)E\{[1 - o_j^d(x)] \ln [1 - o_j(x, W, \theta)]|\bar{\omega}_j\} \\ &\quad + P(\omega_j)E\{[1 - o_j^d(x)] \ln [1 - o_j(x, W, \theta)]|\omega_j\} = P(\bar{\omega}_j)E\{\ln [1 - o_j(x, W, \theta)]|\bar{\omega}_j\} \end{aligned}$$

Similarly

$$\begin{aligned} E[o_j^d(x) \ln o_j(x, W, \theta)] &= P(\bar{\omega}_j)E[o_j^d(x) \ln o_j(x, W, \theta)|\bar{\omega}_j] \\ &\quad + P(\omega_j)E[o_j^d(x) \ln o_j(x, W, \theta)|\omega_j] = P(\omega_j)E[\ln o_j(x, W, \theta)|\omega_j]. \end{aligned}$$

So, collecting the above, we have

$$E(J_{KL}) = -\sum_{j=1}^{K} P(\overline{\omega}_j)E\{\ln[1 - o_j(x, W, \theta)]|\overline{\omega}_j\} - \sum_{j=1}^{K} P(\omega_j)E\{\ln[1 - o_j(x, W, \theta)]|\overline{\omega}_j\}$$

$$= -\sum_{j=1}^{K} P(\overline{\omega}_j) \int_x P(x|\overline{\omega}_j) \ln P(\overline{\omega}_j|x, W, \theta) - \sum_{j=1}^{K} P(\omega_j) \int_x P(x|x\omega_j) \ln P(\omega_j|x, W, \theta)$$

since it follows from Equation (7c) that

$$P(\omega_j|x, W, \theta) = o_j(x, W, \theta),\ P(\overline{\omega}_j|x, W, \theta) = 1 - o_j(x, W, \theta).$$

Thus, it is not difficult to show that $E(J_{KL}) - (E_{KL} + \overline{E}_{KL})$ equals

$$\sum_{j=1}^{K} P(\omega_j) \int_x P(x|\omega_j) \ln P(\omega_j|x)dx + \sum_{j=1}^{K} P(\overline{\omega}_j) \int_x P(x|\overline{\omega}_j) \ln P(\overline{\omega}_j|x)dx$$

which is independent of W, so we proved that the minimization of $E(J_{KL})$ given by Equation (5a) is equivalent to minimizing $E_{KL} + \overline{E}_{KL}$ with respect to W and θ.

Notice that

$$E\left\{-\sum_{p=1}^{Nb}\sum_{k=1}^{K} \{[1 - o_k^d(x_p)] \ln [1 - o_k(x_p, W, \theta)] + o_k^d(x_p) \ln o_k(x_p, W, \theta)\}\right\}$$

$$= E\left\{-\sum_{j=1}^{K} [1 - o_j^d(x_p)] \ln [1 - o_j(x_p, W, \theta)]\right\} + E\left\{-\sum_{j=1}^{K} o_j^d(x_p) \ln o_j(x_p, W, \theta)\right\}.$$

It is easy to see that the above proof is also true for the case that J_{KL} is given by Equation (5b).

.QED.

It is interesting to note that E_{KL} is just the Kullback–Leibler information measure for posterior probability estimation since it could also be rewritten as $E_{KL} = \Sigma_{j=1}^{K} \int_x P(x, \omega_j) \ln \dfrac{P(\omega_j|x)}{P(\omega_j|x, W)} dx$. Thus, the minimization of E_{KL} lets the sequence of the top layer outputs $o_j(x, W, \theta)$, $j = 1, \ldots, K$ be an estimate of the sequence of posterior probabilities $P(\omega_j|x)$, $j = 1, \ldots,$ under the Kullback–Leibler information measure.

$\overline{E}_{KL}$ also resembles the Kullback–Leibler information measure for estimating posterior probability $P(\overline{\omega}_j|x)$, but it is not since

$$P(\overline{\omega}_j \cup \overline{\omega}_k) \neq P(\overline{\omega}_j) + P(\overline{\omega}_k)$$

(i.e., $P(\overline{\omega}_j)$, $j = 1, \ldots, K$ do not represent a probability distribution on the basic element space $\{\omega_1, \ldots, \omega_K\}$). However, if we regard $\overline{\omega}_j$ as a basis element and consider the probability measure on set $\{\overline{\omega}_1, \ldots, \overline{\omega}_K\}$, and introduce the following definitions

$$P'(\overline{\omega}_j) = \frac{P(\overline{\omega}_j)}{\Sigma_{j=1}^{K} P(\overline{\omega}_j)} = \frac{P(\overline{\omega}_k)}{K - 1}, j = 1, \ldots, K, \tag{8a}$$

$$P'(\overline{\omega}_j|x) = \frac{P(\overline{\omega}_j|x)}{K - 1}, j = 1, \ldots, K, \tag{8b}$$

then, we have the Kullback–Leibler information measure

$$\overline{E}'_{KL} = \sum_{j=1}^{K} P'(\overline{\omega}_j) \int_x P(x|\overline{\omega}_j) \ln \frac{P'(\overline{\omega}_j|x)}{P'(\overline{\omega}_j|x, W, \theta)} dx. \tag{9a}$$

Substituting Equation (7) into Equation (9a) gives

$$\overline{E}_{KL} = (K - 1)\overline{E}'_{KL} \tag{9b}$$

So we can see that minimization of $\overline{E}_{KL}$ is equivalent to letting the sequence $\dfrac{1 - o_j(x, W, \theta)}{K - 1}$, $j = 1, \ldots, K$ be an estimate of the sequence of posterior probabilities $P'(\omega_j|x) = P(\omega_j|x)/(K - 1)$, $j = 1, \ldots, K$ under the Kullback–Leibler information measure.

Although by only minimizing either E_{KL} or $\overline{E}_{KL}$ one could let $o_j(x, W, \theta)$, $j = 1, \ldots, K$ be an estimate of the sequence of posterior probabilities $P(\omega_j|x)$, $j = 1, \ldots, K$, the resulting estimation may not satisfy the following basic constraint for a probability measure

$$\sum_{j=1}^{K} o_j(x, W, \theta) = 1. \tag{10a}$$

However, it follows from the above theorem that the minimization of $E(J_{KL})$ is equivalent to minimizing E_{KL} and $\overline{E}_{KL}$ simultaneously. From $P(\overline{\omega}_j|x) = 1 - P(\omega_j|x)$ and Equation (7b), we see that these simultaneous minimizations try to let $o_j(x, W, \theta)$ approach $P(\omega_j|x)$ and $1 - o_j(x, W, \theta)$ approach $P(\overline{\omega}_j|x)$ simultaneously. This is equivalent to letting $o_j(x, W, \theta)$, $j = 1, \ldots, K$ learn the relations

$$P(\omega_j|x) = 1 - P(\overline{\omega}_j|x), j = 1, \ldots, K$$

which implicitly keeps the constraint in Equation (10a) satisfied.

Recall a well-known result [18] that minimization of $E(J_2)$ given in Equation (3a) or (3b) is equivalent to minimizing $\Sigma_{j=1}^{K} E[(o_j(x, W, \theta) - P(\omega_j|x))^2]$ (i.e., letting the sequence $o_j(x, W, \theta)$, $j = 1, \ldots, k$ be the least square estimate of the sequence $P(\omega_j|x), j = 1, \ldots, K)$. We could see that criterion J_{KL} given by Equation (5a) or (5b) is superior to J_2 also in the sense of probability estimation since the Kullback–Leibler information measure is better than the least square measure for probability estimation.

3.3 Training Neural Net as an Associative Controller

In this section we will discuss how to train the one hidden layer net given in Figure 2 as an associative controller by the backpropagation method under the criterion of Equation (5a) or (5b). The key task is the design of the desired output codes for training the controller net on a data set, because once we have the desired output codes the training procedure is just the direct application of the backpropagation method discussed in Sections 3.1 and 3.2.

To design the appropriate output codes (i.e., the input and output mapping relations), we look at the information available in a training set for the problem of multiple classifiers combination. For each sample x in the training set, although its class label $j \in \Lambda = \{1, 2, \ldots M\}$ is known a priori, one cannot design the desired output code for x by Equation (6) alone. The reason is that the task of the controller net is not to classify x into some class, but rather to control K knobs in Figure 1 so that the right classification made by the individual classifiers pass through as the final result. Thus, the desired output should be designed based on the classification results $e_k(x) = j_k, k = 1, \ldots, K$. The following gives a simple way to produce the desired output code according to three different cases:

Case 1. If $j_k \neq j$ for all the $k = 1, \ldots, K$ (i.e., there is no individual classifier giving the right classification), then we let $o_j^d(x) = 0, k = 1, \ldots, K$.

Case 2. If there is only one k such that $j_k = j$ (i.e., there is only one individual classifier giving the right result), then for $j = 1, \ldots, K$ we let

$$o_j^d(x) = \begin{cases} 1, & \text{for } j = k \\ 0, & \text{otherwise} \end{cases}$$

Case 3. When there is more than one individual classifier giving the right result (i.e., there is a subset $S_K \subseteq \{1, 2, \ldots, K\}$ for each $j' \in S_K, j' = j$). In this case, we arbitrarily or randomly choose one j' among S_K and for $j = 1, \ldots, K$, let

$$o_j^d(x) = \begin{cases} 1, & \text{for } j = j' \\ 0, & \text{otherwise} \end{cases}$$

It is clear that the above design for the desired output code reflects our ideal requirement for realizing the new combining principle described in Section 2.1 and the feasible condition for the associative model given in Section 2.2. In fact, the above design is equivalent to the procedure which regroups all the samples of the training set into K classes (i.e., the problem of combining the result of K individual classifiers on the samples of M original classes has been transformed into a problem of classifying the samples of K new classes). Under these new classes, Equation (6) holds and the theorem given in Section 3.2 is applicable. Thus, one can directly use criterion of Equation (5a) or (5b) to train the controller net given in Figure 2 by the procedure of backpropagation method introduced in Section 3.1 and in Equation (5c).

In Case 3, there is also an alternative way of choosing a desired output code among several possibilities. One could simply take all the choices, that is, let

$$o_j^d(x) = \begin{cases} 1, & \text{for all } j' \in S_k \\ 0, & \text{otherwise.} \end{cases} \tag{11a}$$

Referring to the model of Figure 1, one could find that this will also make the associative switch work. The reason is that instead of requiring the output code O to control K knobs so that only one of the outputs of K classifiers could pass through as the final label j, we can require that each code O turns "on" all the knobs of the output channels of these classifiers which label x correctly and turns "off" all the other knobs. It should be noted that under this design of the desired output, the theorem in Section 3.2 is not directly applicable. However, we can still use criterion of Equation (5a) or (5b) to improve the local minimum problem encountered by criterion of Equation (3a) or (3b).

Besides the above, there is also another method which can improve the original one given in Case 3. For the original method, the desired output code is chosen randomly or by arbitrarily picking one among several choices. However, the different choices will usually affect the final result. In fact, the different choices are equivalent to regrouping x into the different classes, which will result in the different regrouping of the training set. In general, the input-output mapping relation of some regrouped K classes may be simpler than that of others, in turn, the simpler mapping relation will need a neural net of smaller size and a shorter training period as well. Thus, it is important to make selection carefully. In the following, a policy called the *minimal disturbance* will be proposed for this purpose.

Let $r = |S_K|$ denote the number of elements in S_K and $O^d(x, j), j = 1, \ldots, r$ all the possible choices of a desired output code for an input x. Although all $O^d(x, j), j = 1, \ldots, r$ satisfy the required condition in Section 2.2 for the associative switch, the current errors $J_{KL}(j), j = 1, \ldots, r$ obtained by putting $O^d(x, j), j = 1, \ldots, r$ into Equation (5a) in the current net will be different. First, the smaller error means that x is regrouped into one class which has learned quite a lot during the past and thus a smaller effort is presently needed to adapt x, which implicitly indicates that some choice may give a simpler input-output mapping relation. Second, the smaller error also means that a smaller modification of the current weights W and bias θ will be made for adapting the input x, thus less learning time could be expected. Clearly, these two considerations suggest to select the j' in S_k such that $J_{KL}(j') = Min_{j \in S_K} J_{KL}(j)$ (i.e., to make the minimal modification of the current net or to give the minimal disturbance to what was learned in the past). For short, we call this the minimal disturbance policy.

To implement this policy, we insert the process for designing the desired output into the whole process of training. That is, for each randomly picked x, if it does not correspond to Case 3, then one assigns x a desired output code as described in the above Cases 1 and 2, and modifies the connecting weights W and bias θ by the backpropagation method described in Section 3.1 and Equation (5c). Otherwise, we try to select j' such that $J_{KL}(j') = Min_{j \in S_K} J_{KL}(j)$, and assign x a desired output code by

$$o_j^d(x) = \begin{cases} 1, & \text{for } j = j' \\ 0, & \text{otherwise} \end{cases}$$

which is then used for a modification made on the net by the backpropagation to adapt the present input x. In real implementation, one actually only needs to choose j' such that $o_{j'}(x, W, \theta) = Max_{j \in S_K} o_j(x, W, \theta)$, since the problem of finding j' such that $J_{KL}(j') = Min_{j \in S_K} J_{KL}(j)$ is equivalent to finding j' such that $o_{j'}(x, W, \theta) = Max_{j \in S_K} o_j(x, W, \theta)$. This fact is made obvious by the following equation

$$J_{KL}(j) = -\ln o_j(x, W, \theta) - \sum_{k \neq j}^{K} \ln[1 - o_k(x, W, \theta)] \tag{11b}$$

which is a simplified version of Equation (5a).

4 COMPUTER EXPERIMENTS ON COMBINING MULTIPLE CLASSIFIERS FOR RECOGNIZING TOTALLY UNCONSTRAINED HANDWRITTEN NUMERALS

4.1 Individual Classifiers and Database

The four classifiers proposed in the paper of Suen et al. [1] are considered as individual classifiers in the experiments. As in the paper [1], the four classifiers are named *expert #1*, *expert #2*, *expert #3*, and *expert #4*, and are denoted by e_1, e_2, e_3, e_4. The first three are based on the features extracted from the skeletons, while e_4 is based on the features derived from contours. See the paper [1] for more details.

The data used here come from the U.S. zip code database of the Concordia OCR research team. This database contains run-length coded binarized digits. The samples were originally collected from the dead-letter envelopes by the U.S. Postal Service at different locations, some of these samples are shown in Figure 3.

After some preprocessing (see paper [1] for details), 4,000 samples (400 × 10 digits, that is, each of the 10 numeral classes contains 400 samples) were used for training the four experts, and then, a new set of 2,000 samples (200 × 10 digits) was used for testing. The following results were obtained from the testing set.

In Table 1.1, Recogn., Subtsti., Reject., and Reliab. are abbreviations of recognition, substitution, rejection, and reliability rates, respectively. The reliability rate is defined by

$$Reliability = \frac{Recognition}{100\% - Rejection}. \tag{12}$$

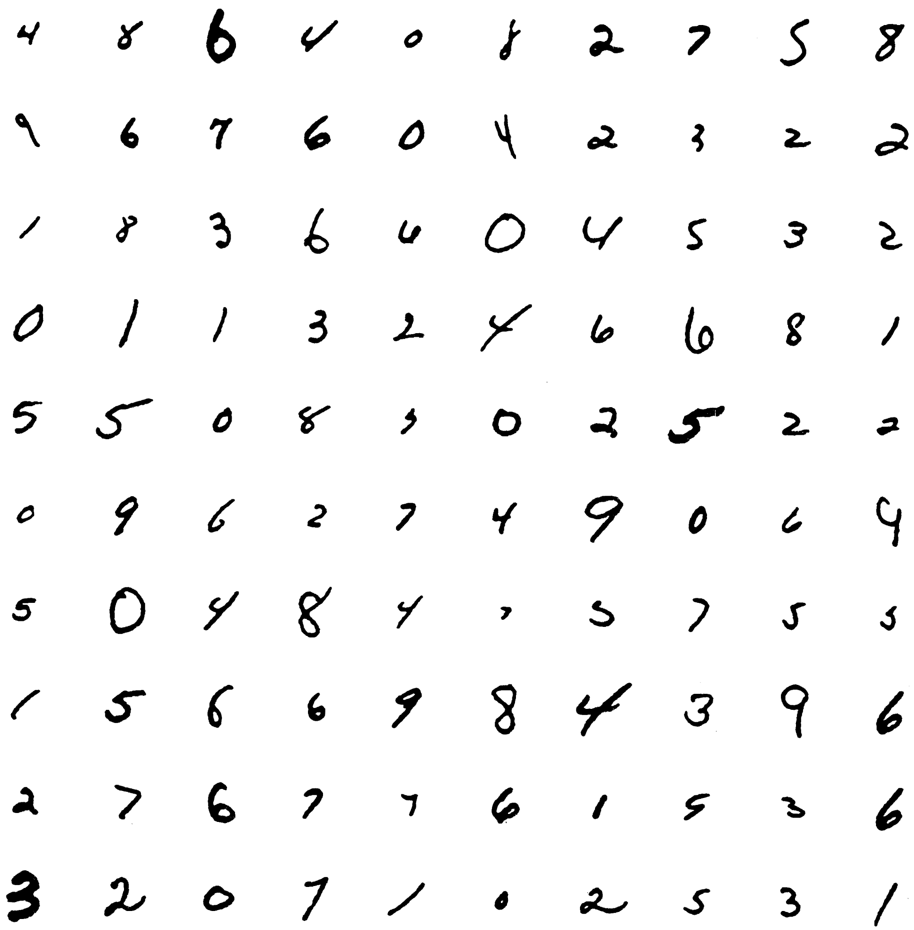


Figure 3. Samples of numerals collected from the U.S. Post Office.

Table 1.1. The Results of Four Experts

	(%) Recogn.	(%) Substi.	(%) Reject.	(%) Reliab.
e_1	86.05	2.25	11.70	97.45
e_2	93.10	2.95	3.95	96.98
e_3	92.95	2.15	4.90	97.74
e_4	93.90	1.60	4.50	98.32

In addition, if e_1 assigns a subset of labels to an input, the input is regarded as being rejected in Table 1.1. Moreover, in the following, this nonunique recognition of e_1 is also always regarded as a rejection except for some cases specifically indicated.

In addition, for some practical reasons, we presently could only access the classification results for the above mentioned 2,000 test samples. We have to further divide them into two sets, each contains 1,000 samples (100 samples for each digit), the first set is used as the training set to train the associative switch, the second set is used to test the combination performances of the associative switch. For convenience of comparing the combination performances with those of individual classifiers, we decompose the results of Table 1.1 into Table 1.2 and Table 1.3 to show the performances of individual classifiers on the 1,000 samples of the first set and the second set, respectively.

By comparing the above two tables, one could see that the performances of e_i, $i = 1, 2, 3, 4$ on the first set are better than those on the second set (i.e., we selected the difficult half of the original 2,000 samples for testing our associative switch).

4.2 Experiments on the Combination by Associative Switch

For the above specific problem, we have $K = 4$ and $M = 10$ for the associative switch model given in Figure 1(b). In addition, a mapping $M(x)$ of x, instead of x itself, is used as the input to the controller net. As suggested in Section 2.2, this

Table 1.2. Results of Individual Classifiers on the 1,000 Samples of the First Set

	(%) Recogn.	(%) Substi.	(%) Reject.	(%) Reliab.
e_1	87.0	1.5	11.5	98.31
e_2	94.4	2.4	3.2	97.52
e_3	95.0	1.2	3.8	99.79
e_4	94.8	0.9	4.3	99.06

Table 1.3. Results of Individual Classifiers on the 1,000 Samples of the Second Set

	(%) Recogn.	(%) Substi.	(%) Reject.	(%) Reliab.
e_1	85.1	3.0	11.9	96.59
e_2	91.8	3.5	4.7	96.33
e_3	90.9	3.1	6.0	96.70
e_4	93.0	2.3	4.7	97.59

$M(x)$ is simply the label vector $J(x) = [j_1, \ldots, j_4]$ with $j_k = -1$ replacing the nonnumeral notation $j_k = \phi$. Each j_k is the label assigned to x by classifier e_k. We choose $J(x)$ for practical reason: the four classifiers e_1, e_2, e_3, e_4 are based on different methods and features, the common pattern input to the four is actually the binary matrix of digits, which is usually of high dimension when represented as a vector x. The direct use of such x as the input to the controller will greatly increase the size of the neural net, thus some feature extraction or dimension reduction transform is needed to produce $M(x)$. For simplicity, we just use $J(x)$ since it is already available without any extra work.

In our experiments, the initial values of the connection weights $w_{ij}^{(r)}$ and the bias $\theta_j^{(r)}$ are given by random numbers from a uniform distribution on $[-0.5, 0.5]$. For simplicity, the learning rate is fixed at $\alpha = 0.15$, and the desired output codes are designed by Equation (11a). The preliminary results are quite promising, they show that the proposed associative switch works well. For illustration, we list some results below.

Table 2.1 gives the combination results of the associative switch with different values of the threshold o_t in Equation (1b). Here, the controller net is quite simple, it only contains four units in the hidden layer. The listed results are obtained after the net has been trained by 20,000 learning steps, where one step means one whole process of randomly picking a pattern x from the training set and modifying the present W and θ to adapt x. For convenience of making a comparison with the results of the individual classifiers given in Tables 1.2 and 1.3, here we also rewrite the results of individual classifier e_3 for the training set and e_4 for the testing set since e_3 and e_4 performs best among the four individual classifiers on each set, respectively.

The improvements obtained in combination approach are easily noticeable in Tables 2.1 and 2.2. To show details of classification results, we also give the confusion matrices corresponding to $o_t = 0.6$ in Tables 2.3 and 2.4, respectively.

5 CONCLUSIONS

In our previous paper [5], we argued that combination of several individual classifiers is a general problem which is often encountered in various application areas

Table 2.1. The Combination Results on the Training Set

o_t	(%) Recogn.	(%) Substi.	(%) Reject.	(%) Reliab.
0.9	95.7	0.2	4.1	99.79
0.8	95.8	0.2	4.0	99.79
0.7	95.8	0.2	4.0	99.79
0.6	97.3	0.2	2.5	99.79
0.5	97.3	0.2	2.5	99.79
e_3 :	95.0	1.2	3.8	99.79

Table 2.2. The Combination Results on the Testing Set

o_t	(%) Recogn.	(%) Substi.	(%) Reject.	(%) Reliab.
0.9	94.3	0.2	5.5	99.79
0.8	94.7	0.3	5.0	99.68
0.7	94.8	0.4	4.8	99.58
0.6	95.9	0.4	3.7	99.58
0.5	95.9	0.4	3.7	99.58
e_4 :	93.0	2.3	4.7	97.59

Table 2.3. The Combination Result on the Training Set

i|*o*	0	1	2	3	4	5	6	7	7	9	rej.
0 :	95	0	0	0	0	0	1	0	0	0	4
1 :	0	99	0	0	0	0	0	0	0	0	1
2 :	0	0	95	0	0	0	0	0	0	0	5
3 :	0	0	0	98	0	1	0	0	0	0	1
4 :	0	0	0	0	98	0	0	0	0	0	2
5 :	0	0	0	0	0	97	0	0	0	0	3
6 :	0	0	0	0	0	0	97	0	0	0	3
7 :	0	0	0	0	0	0	0	97	0	0	3
8 :	0	0	0	0	0	0	0	0	99	0	1
9 :	0	0	0	0	0	0	0	0	0	98	2

Recogn.: 97.3% Substi.: 0.2% Reject.: 2.5% Relab.: 99.79%

Table 2.4. The Combination Result on the Testing Set

i|*o*	0	1	2	3	4	5	6	7	7	9	rej.
0 :	96	0	0	0	0	0	0	0	0	0	4
1 :	0	97	0	0	0	0	0	0	0	0	3
2 :	0	0	93	0	0	0	0	0	0	0	7
3 :	0	0	0	93	0	0	0	0	0	0	7
4 :	0	0	0	0	94	0	0	0	0	0	6
5 :	0	0	0	0	0	98	0	0	0	0	2
6 :	1	0	0	0	0	0	97	0	0	0	2
7 :	0	0	1	0	1	0	0	97	0	0	1
8 :	0	0	0	0	0	0	0	0	98	0	2
9 :	0	0	0	0	0	0	0	0	1	96	3

Recogn.: 95.9% Substi.: 0.4% Reject.: 3.7% Relab.: 99.58%

of pattern recognition. In the present paper, we further argued that availability of various neural net classifiers may also raise the need for a study of integrating techniques for multiple classifiers. We thus explored the possibility of using the neural-net approach to perform the task of combining multiple classifiers. In our paper [5], several combination methods were proposed based on two general combination principles: the committee voting and the evidences gathering with uncertainty reasoning. The two principles share one common point of trying to assemble or synthesize the outcomes of all the individuals as the final output. In this paper, we proposed a new principle which only selects one expert for the total responsibility. This new principle is realized by a novel technique called *associative switch*. The switch is controlled by a multilayer perceptron neural net trained by backpropagation technique with a modified error criterion. When an unlabeled pattern is input to each individual classifier, it also goes to the neural net for associatively recalling a code which controls the switch to decide whether the result of each classifier could pass through as a final result. We studied how to appropriately train the net to fulfil the associative controlling task, and analyzed the advantage of the modified error criterion used here. Furthermore, we applied this associative switch to tackle the problem of combining multiple classifiers for recognizing totally unconstrained handwritten numerals. The experimental results have shown that the associative switch works well and can produce considerably better classification results.

REFERENCES

[1] C.Y. Suen, C. Nadal, T.A. Mai, R. Legault, and L. Lam, "Recognition of Totally Unconstrained Handwritten Numerals Based on the Concept of Multiple Experts," in *Frontiers in Handwriting Recognition*, C.Y. Suen, Ed. Montreal: Concordia University, 1990, pp. 131–143.

[2] C. Nadal, R. Legault, and C.Y. Suen, "Complementary Algorithms for the Recognition of Totally Unconstrained Handwritten Numerals," in *Proc. 10th Int. Conf. on Pattern Recognition*, Vol. 1, 1990, pp. 434–449.

[3] J.J. Hull, A. Commike and T.K. Ho, "Multiple Algorithms for Handwritten Character Recognition," in *Frontiers in Handwriting Recognition*, C.Y. Suen, Ed. Montreal: Concordia University, 1990, pp. 117–129.

[4] T.K. Ho, J.J. Hull, and S.N. Srihari, "Combination of Structural Classifiers," in *Proc. 1990 IAPR Workshop on Syntactic and Structural Pattern Recognition*, 1990, pp. 123–137.

[5] L. Xu, A. Krzyżak, and C.Y. Suen, "Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition," *IEEE Trans. Systems, Man and Cybernetics*, Vol. 22, pp. 418–435, May/Jun. 1992.

[6] P. Werbos, "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences," Doctoral dissertation, Harvard University, Cambridge, MA, 1974.

[7] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning Internal Representation by Error Propagation," *ICS Rep. 8506*, Institute of Cognitive Science, University of California at San Diego, La Jolla, CA, Sep. 1985.

[8] T. Kohonen, *Self-Organization and Associative Memory*. Berlin: Springer-Verlag, 1988.

[9] T. Kohonen, "The Self-Organizing Map," *Proc. IEEE*, Vol. 78, pp. 1464–1480, Sep. 1990.

[10] G.A. Carpenter and S. Grossberg, "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine," *Computer Vision, Graphics, and Image Processing*, Vol. 37, pp. 54–115, Jan. 1987.

[11] G.A. Carpenter and S. Grossberg, "Art 2: Self-Organization of Stable Category Recognition Codes for Analog Output Patterns," *Applied Optics*, Vol. 26, pp. 4919–4930, Dec. 1987.

[12] G.A. Carpenter and S. Grossberg, "Art 3: Hierarchical Search. Chemical Transmitters in Self-Organizing Pattern Recognition Architectures," in *Proc. Int. Joint. Conf. on Neural Networks*, Vol. 2, 1990, pp. 30–33.

[13] B. Widrow and M.A. Lehr, "30 Years of Adaptive Neural Networks: Perceptron, Madaline, Backpropagation," *Proc. IEEE*, Vol. 78, pp. 1415–1442, Sep. 1990.

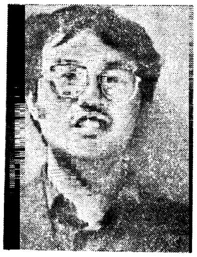
[14] R.A. Jacobs and M.I. Jordan, "A Competitive Modular Connectionist Architecture," in *Advances in Neural Information Processing Systems*, Vol. 3, R.P. Lippmann, J.E. Moody, and D.S. Touretzky, Eds. San Mateo, CA: Morgan Kaufmann, 1991, pp. 767–773.

[15] S.J. Nowlan and G.E. Hinton, "Evaluation of Adaptive Mixtures of Competing Experts," in *Advances in Neural Information Processing Systems*, Vol. 3, R.P. Lippmann, J.E. Moody, and D.S. Touretzky, Eds. San Mateo, CA: Morgan Kaufmann, 1991, pp. 774–780.

[16] A. Krzyżak, W. Dai, and C.Y. Suen, "Classification of Large Sets of Handwritten Characters Using Modified Back Propagation Model," *Proc. Int. Joint. Conf. on Neural Networks*, Vol. 3, 1990, pp. 225–231.

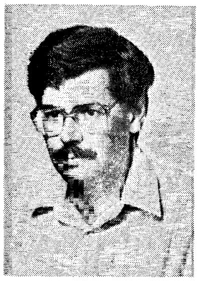
[17] G.E. Hinton, "Connectionist Learning Procedures," *Artificial Intelligence*, Vol. 40, pp. 185–235, Sep. 1989.

[18] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.



Lei Xu has been an associate professor of Peking University (Beijing, P.R. China) since September 1988, and has spent a year as a visiting scientist at the Harvard Robotics Lab., Harvard University (USA). Since September 1993 he has been with the Department of Computer Science, The Chinese University of Hong Kong. He received his Bachelor Degree from the Department of Electrical Engineering, Harbin Institute of Technology in 1982, and his Master and Doctoral degrees both in Pattern Recognition and Signal Processing from Tsinghua University in 1984 and 1987, respectively. From June 1987 to June 1988, he was a post-doctoral fellow at the Department of Mathematics, Peking University. He also visited the Department of Information Technology, Lappeenranta University of Technology, Finland, and the Department of Computer Science, Concordia University, Canada as Senior researcher and visiting scholar, each for one year, respectively. He is an author of over 80 papers on neural networks, pattern recognition and computer vision, signal processing and artificial intelligence, as well as their applications to OCR and to automatic seismic signal processing and interpretation. His present research interests are mainly focused on neural networks and computer vision.

He also has had the experiences of serving as a reviewer for *INNS Journal Neural Networks, IEEE Trans. on Neural Networks, Artificial Intelligence, International Journal of Visual Communication and Image Representation*, and several Chinese high-level scientific journals, as well as for the international conference on Neural Networks INNC-90 (Paris) and for NIPS-92. Dr. Xu was one of 40 winners of the First FOK YING TUNG EDUCATION FOUNDATION PRIZE for young teachers of the universities in the whole nation of P.R. China, 1988. He was also the second of the ten winners of the 2nd BEIJING YOUNG SCIENTISTS PRIZE awarded by Beijing Association for Science and Technology in 1988. Moreover, he was the first of the five winners of the excellent paper award for young researchers in the *1988 National Conference of the Chinese Automation Society*, May, 1988.



Adam Krzyżak was born in Szczecin, Poland, on May 1, 1953. He received his M.Sc. and Ph.D. degrees in computer engineering from the Technical University of Wroclaw, Poland, in 1977 and 1980, respectively. In 1980, he became an Assistant Professor in the Institute of Engineering Cybernetics, Technical University of Wroclaw, Poland. From November 1982 till July 1983 he was a Postdoctorate Fellow receiving the International Scientific Exchange Award in the School of Computer Science, McGill University, Montreal, PQ, Canada. Since August 1983, he has been with the Department of Computer Science, Concordia University, Montreal, where he is currently an Associate Professor. He spent the spring semester of 1992 at Technion-Israel Institute of Technology on Lady Davis Scholarship. He has published over 90 papers in the areas of pattern recognition, image processing, computer vision, identification and estimation of control systems as well as on various applications of probability theory and statistics. He is an editor of the book, Computer Vision and Pattern Recognition (Singapore: World Scientific, 1989) and an Associate Editor of the Pattern Recognition Journal.

Ching Y. Suen received his M.Sc. (Eng.) degree from the University of Hong Kong and a Ph.D. degree from the University of British Columbia, Canada. In 1972, he joined the Department of Computer Science of Concordia University, Montreal, Canada, where he became Professor in 1979 and served as Chairman from 1980 to 1984. Presently he is the Director of CENPARMI, the new research Center for Pattern Recognition and Machine Intelligence of Concordia. During the past 15 years, he was also appointed to visiting positions in several institutions in different countries. Professor Suen is the author/editor of ten books on subjects ranging from *Computer Vision and Shape Recognition, Frontiers in Handwriting Recognition*, to *Computational Analysis of Mandarin and Chinese*. His latest book is entitled *Operational Expert System*

Applications in Canada, published in December 1991 by Pergamon. Dr. Suen is the author of more than 250 papers and his current interests include pattern recognition and machine intelligence, expert systems, optical character recognition and document processing, and computational linguistics. An active member of several professional societies and Fellow of the IEEE, Dr. Suen is an Associate Editor of several journals including *Pattern Recognition, Pattern Recognition Letters, International Journal of Pattern Recognition and Artificial Intelligence*, and *Signal Processing*. He is the Founder and Editor-in-Chief of *Computer Processing of Chinese and Oriental Languages*, an international journal of the Chinese Language Computer Society. He had also been an Associate Editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence and is still serving as a member of its Advisory Board. During the past 15 years, Professor Suen has served as Chairman of the Character and Mark Recognition Committee of the Canadian Standards Association which developed several Canadian Standards on optical character recognition. He is the Past President of the Canadian Image Processing and Pattern Recognition Society, Governor of the International Association for Pattern Recognition, and President of the Chinese Language Computer Society. He has been a consultant to numerous industrial companies, and has given many lectures outside Concordia University, in Canada and abroad, at universities and in industries. He has organized many international conferences on subjects of his specialties, and was Founder and Chairman of the first International Workshop on Frontiers in Handwriting Recognition held in Montreal, April 1990; and also Co-Founder and Co-Chairman of the First International Conference on Document Analysis and Recognition held in France, September 1991, and has been Co-Chairman of the Second ICDAR held in Japan, Oct. 1993.