A HIERARCHICAL NEURAL NETWORK ARCHITECTURE FOR HANDWRITTEN NUMERAL RECOGNITION

J. CAO,† M. AHMADI†,* and M. SHRIDHAR‡
†Department of Electrical Engineering, University of Windsor, Windsor, Ontario, Canada N9B 3P4
‡Department of Electrical and Computer Engineering, University of Michigan-Dearborn, Dearborn, MI 48128, U.S.A.

(Received 1 November 1994; in revised form 22 April 1996; received for publication 13 May 1996)

Abstract—This paper presents a hierarchical neural network architecture for recognition of handwritten numeral characters. In this new architecture, two separately trained neural networks connected in series, use the pixels of the numeral image as input and yield ten outputs, the largest of which identifies the class to which the numeral image belongs. The first neural network generates the principal components of the numeral image using Oja’s rule, while the second neural network uses an unsupervised learning strategy to group the principal components into distinct character clusters. In this scheme, there is more than one cluster for each numeral class. The decomposition of the global network into two independent neural networks facilitates rapid and efficient training of the individual neural networks. Results obtained with a large independently generated data set indicate the effectiveness of the proposed architecture. Copyright © 1997 Pattern Recognition Society. Published by Elsevier Science Ltd.

Handwritten character recognition Neural networks Clustering
Principal component analysis Pattern recognition Bayes learning

1. INTRODUCTION

Machine recognition of handwritten characters continues to be a topic of intense interest among many researchers, primarily due to the potential commercial applications in such diverse fields as document recognition, check processing, forms processing, address recognition etc.

The need for new techniques arises from the fact that even a marginal increase in recognition accuracy of individual characters can have a significant impact on the overall recognition of character strings such as words, postal codes, ZIP codes, courtesy amounts in checks, street number recognition etc.

A number of useful papers in this area may be found in references (1–5). Recognition techniques have been based on syntactic, structural, statistical or neural net methodologies. Each of these approaches has yielded reasonably high recognition accuracy; however, it is worth observing that it is more important to have algorithms that control the error rate by adjusting the rejection rate by using suitable thresholds. For example, a 1% error rate for single digits often translates to a 5% error rate for a five-digit numeral string such as a ZIP code. It is not uncommon to specify an error rate of less than 1% for ZIP codes, 0.1% for bank check courtesy amounts; such low error rates are achievable only if rejection rates can be set arbitrarily (about 10% rejection for 1% error and almost 60% rejection for 0.1% error).

Recently neural networks have been proposed for developing high accuracy, low error rate numeral recognition systems, using a variety of architectures. (5–7)

* Author to whom all correspondence should be addressed.

The basic problems associated with neural networks may be summarized as follows: (6–10)

1. The need to use a large number of representative samples is of crucial importance in training the neural network. This often results in a fairly slow learning rate. Also convergence of the learning algorithm is often not easy to achieve.
2. The neural network itself becomes excessively large, if the individual pixels of the character image are used as input. Even at a low spatial resolution of 16 x 16 for a character image, one has to deal with 256 input nodes. When this is combined with the nodes distributed across the hidden layers, one is faced with the awesome task of training a very large network.
3. The amount of training a neural network receives is crucial in achieving high recognition accuracy with test characters. It is possible to over-train or under-train a neural network, resulting in poor performance in real world recognition applications.

In this paper, the authors describe a hierarchical two-stage neural network architecture for handwritten numeral recognition. The first stage which may be viewed as a feature extractor is a two-layer neural network trained to yield the principal components (11,12) of the numeral image. The number of principal components p is typically much smaller than N, the number of pixels in the image. Oja’s (13) rule has been adopted for this stage.

The second stage clusters the principal components using an unsupervised learning algorithm and a weighted distance measure for evaluating closeness of a principal component vector to the cluster centers.
If one assumes a multivariate Gaussian distribution for the principal components, this distance measure is equivalent to a quadratic discriminant measure.

Some of the difficulties associated with neural networks are avoided by this hierarchical approach, since each network is trained independently. Also each network is essentially a two-layer network; therefore these networks can be trained very efficiently without the convergence problems associated with multilayer networks.

The organization of this paper is as follows. In Section 2, the neural network model is described. In Section 3, experimental results are presented, followed by brief conclusions in Section 4.

2. SYSTEM ARCHITECTURE

The recognition system uses two neural networks that can be trained independently to perform the basic recognition task. The neural network structure is illustrated in Fig. 1. The first network which functions as a feature extractor generates “p” principal components of the character image, using Oja’s rule. The second neural network which functions as a classifier assigns the output of the first network to one of the clusters (numeral sub-classes). The second network is essentially an unsupervised clustering network that uses a weighted distance measure to derive the clusters. The recognition system described here can be considered as a neural implementation of a statistical classifier. Descriptions of the two neural networks are provided in Sections 2.1 and 2.2.

2.1. Principal component extraction

The authors propose a neural network to extract the first “p” principal components of the character image. The techniques described in this section are based on the works of Oja and Sanger, who used a neural network to generate the principal components for zero-mean random vectors. The use of principal components in the recognition system offers several advantages:

(i) The principal components are mutually uncorrelated.
(ii) A small subset of the principal components contains most of the relevant information.

These advantages are important in the development of practical recognition systems, as a significant data reduction is achieved by using only a small subset of the principal components. The principal components may be generated by a two-layer feed-forward neural network with N input nodes (one node for each image pixel) and p output nodes (one for each principal component). In this configuration, the i-th output \( y_i \) is given by

\[
y_i = \sum_{j=1}^{N} w_{ij} x_j = w_i^T x,
\]

where \( x_j \) is the j-th element of input vector \( X \) and \( w_{ij} \) is the connection weight between input node j and output node i.

Two rules are generally used in adjusting the weights during the learning phase. Sanger’s learning rule for adjusting the weights is given by

\[
\Delta w_{ij} = \eta y_i \left( x_j - \sum_{k=1}^{l} y_k w_{kj} \right),
\]

where \( \eta \) is the learning rate.

Oja proposed a different learning rule for adjusting the weights and is given by

\[
\Delta w_{ij} = \eta y_i \left( x_j - \sum_{k=1}^{p} y_k w_{kj} \right).
\]

Of the two rules, Sanger’s rule leads to true principal components as defined by Karhounen-Loeve (KL) transform, where the variances of the consecutive principal components become progressively smaller. Oja’s rule, on the other hand, generates components that span the same subspace as the components generated by Sanger’s rule. However, the variances of the components are nearly equal. In this paper, the authors adopted Oja’s rule as it was more appropriate for pattern recognition applications. Since Oja’s rule does not yield principal components as defined by KL transform, the components generated by using Oja’s rule may be viewed as principal features. Since only a two-layer network is used, rapid convergence of the weight vector is assured.

2.2. The Bayes incremental clustering neural net (BICNN)

The second stage of the recognition system uses the principal features generated by the first network and groups them into distinct clusters. The authors propose the use of conditional probability function to guide the clustering process. Assuming Gaussian distribution for the principal features, conditional probability of a feature vector belonging to a given sub-cluster is evaluated. If this probability is low with respect to each of the existing clusters, then a new cluster is initiated.
After all the training samples have been used, merger operations are applied to reduce the number of clusters. Figure 2 shows the architecture of the Bayes incremental learning neural network. For each numeral class there are M processing elements (PE) initially and each PE represents one sub-cluster. A K-means algorithm is initially used to classify the training samples of handwritten numerals into M sub-clusters for each class according to the feature vectors. M is typically chosen to be 3.

The probability distribution of the feature vectors within each sub-cluster is evaluated, assuming a multivariate Gaussian distribution. As each feature is assigned to one of the sub-clusters, the parameters of the probability distribution are updated. The following equations define the conditional probabilities and the update rules for the parameters of the distribution.

The conditional probability is defined as:

\[ P_{mk}(Y_k \mid C_m) = \frac{1}{(2\pi)^{N/2} \left( \prod_{n=1}^{N} \sigma_{mn} \right)} \exp \left[ -\frac{1}{2} \sum_{n=1}^{N} \left( Y_{kn} - \mu_{mn} \right)^2 \sigma_{mn}^{-2} \right], \]

where \( Y_k \) is the input vector, i.e. the output of the PCA net (Fig. 3), of the kth pattern; \( Y_{kn} \) is the nth element of \( Y_k \); \( \mu_{mn} \) is the nth element of the mean vector of cluster \( m \); \( \sigma_{mn} \) is the nth element of the variance of cluster \( m \) and \( C_m \) is the numeral class to which the cluster "m" belongs.

The update rules are

\[ \mu_{mn} = \frac{\mu_{mn} T_m}{T_m + 1} + \frac{Y_{mn}}{T_m + 1}, \]

(5)

\[ \left( \sigma_{mn}^2 \right)^{-1} = \frac{\left( \sigma_{mn}^2 + \mu_{mn}^2 \right) T_m + \left( Y_{mn}^2 \right)}{T_m + 1} - \left( \mu_{mn}^2 \right)^{-1}, \]

(6)

where \( T_m \) is the total number of training patterns that belong to the mth sub-cluster.

Each feature vector obtained in stage 1 is assigned to the sub-cluster with the highest conditional probability as determined from equation (4). A new sub-cluster is constructed, if the conditional probability of the feature vector with respect to each existing sub-cluster is below a specified threshold.

The mean and the variance for the new sub-cluster are defined as

\[ \mu_{(M+1)n} = Y_{kn}, \]

(7)

\[ \sigma_{(M+1)n}^2 = \frac{\sum_{n=1}^{M} \sigma_{mn}^2}{M}, \]

(8)

where \( M \) is the total number of sub-clusters in a given class and \( M+1 \) denotes the new sub-cluster.

Another operation during the learning phase is the merging of within-class sub-clusters, when the corresponding distances between the clusters are below a specified threshold. When two sub-clusters are merged, the parameters of the probability distribution are updated. The algorithm is shown in Table 1. The inter-cluster distance is defined as

\[ d(\omega_1, \omega_2) = \sqrt{\frac{(u_1-u_2)^2}{\sigma_1^2} + \frac{(u_1-u_2)^2}{\sigma_2^2}}. \]

(9)

In this definition, \( u_1 \) and \( u_2 \) are the means of the two clusters and \( \sigma_1 \) and \( \sigma_2 \) are the variances of the two clusters. In the authors' study with handwritten numerals, the learning phase yielded 188 sub-clusters. However, after applying the merging algorithm, the number of clusters was reduced to 90.

<table>
<thead>
<tr>
<th>Table 1. The cluster merging algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster_merging()</td>
</tr>
<tr>
<td>do</td>
</tr>
<tr>
<td>min_dis = min(intercluster_distances);</td>
</tr>
<tr>
<td>if(min_dis &lt; merge_thres)</td>
</tr>
<tr>
<td>( \rho_{AB} = \frac{\rho_{AA} T_A + \rho_{BB} T_B}{T_A + T_B} )</td>
</tr>
<tr>
<td>( \sigma_{AB}^2 = \frac{1}{T_A + T_B} \left( T_A (\sigma_{AA}^2 + \mu_{AA}^2) + T_B (\sigma_{BB}^2 + \mu_{BB}^2) \right) - \mu_{AB}^2 )</td>
</tr>
<tr>
<td>( T_{AB} = T_A + T_B );</td>
</tr>
<tr>
<td>active_nodes--;</td>
</tr>
<tr>
<td>while(min_dis &lt; merge_thres);</td>
</tr>
<tr>
<td>}</td>
</tr>
</tbody>
</table>

Here cluster A, and cluster B are to be merged to form cluster AB. \( T_A, T_B \) and \( T_{AB} \) denote the number of samples in sub-clusters A, B and AB, respectively. \( \rho_{AA}, \rho_{BB} \) and \( \rho_{AB} \) are the mean vectors of clusters A, B and AB, respectively. \( \mu_{AA}, \mu_{BB} \) and \( \mu_{AB} \) are the nth elements of \( \rho_{AA}, \rho_{BB} \) and \( \rho_{AB} \), respectively. \( \sigma_{AA}, \sigma_{BB} \) and \( \sigma_{AB} \) are the nth elements of the variances of clusters A, B and AB, respectively.
3. EXPERIMENTAL RESULTS

3.1. Data collection

The data used in this paper consists of 13,200 handwritten digit characters provided by CGA Alcatel and the U.S. Postal Service. Figure 4 shows some of the samples from this data set. Among these, 6800 (680 samples for each class) samples were used to train the system, and the remaining 6400 samples (640 samples for each class) were used to evaluate the performance.

3.2. Preprocessing

The preprocessing of the images of handwritten samples included slant correction, size normalization, smoothing and mapping. The binary image pattern was first passed through a slant correction process. The slant of a character is estimated by the direction chain codes of its contours as defined in equation (10).

$$\theta = \tan^{-1}\left(\frac{n_3 + n_2 - n_1}{n_1 - n_3}\right)$$

where $n_i$ is the number of chain code elements at angle of $i\times45^\circ$, $i = 1, 2, 3$. After slant correction, the image was scaled to a size of $80 \times 64$ pixels. A non-linear smoothing algorithm was then applied to the normalized image. In this process, if five consecutive neighbors (8-neighbor) of a black pixel at $(i,j)$, are all identical and equal to “b” (0 or 1), then $x(i,j)$ was set to “b”. During the smoothing process, small cavities on the character boundary were filled and small bumps were removed. A neighborhood $(3 \times 3)$ averaging algorithm was then repeatedly applied on the image six times. This process essentially yielded a Gaussian filtered image. Also finally, the processed image was down sampled to a size of $8 \times 8$ pixels. The mapping was performed by taking the average gray level in each of the $8 \times 8$ zones. The resulting $8 \times 8$ gray level image was used as the 64-element input vector to the system. Figure 5 illustrates the steps of the preprocessing.

3.3. Test results

The proposed system was trained and tested separately on the above-mentioned data. The classification for a test pattern is made based on the Euclidean distance measure between the test pattern and each of the clusters. The test sample is assigned the class of the cluster that is closest to the test pattern.

In many practical applications, it is essential to keep the error rate very low (< 0.5%). This is often achieved by allowing a certain amount of rejection of the test samples. The test pattern is often rejected, if the smallest distance (belonging to class $j$) and the second smallest distance (belonging to class $i \neq j$) differ by less than a threshold $T$. The classification is accepted, however, if $j = i$. Otherwise, the test pattern is rejected when $j \neq i$. Error rates can be arbitrarily lowered by increasing the threshold value $T$. However, there will be a consequent increase in the rejection rate. The decision rule is illustrated in Table 2.

![Fig. 4. Typical samples from the handwritten data set.](image-url)

<table>
<thead>
<tr>
<th>The Decision rule for the proposed system</th>
</tr>
</thead>
<tbody>
<tr>
<td>For an input pattern $x$, order clusters $\omega_1, \omega_2, \omega_3, \ldots, \omega_K$ such that, $d(x, \omega_1) \leq d(x, \omega_2) \leq \ldots \leq d(x, \omega_K)$, the decision $s(x)$ is defined by $s(x) = \begin{cases} j &amp; \text{if}\left(\left(\omega_1 \in C_j \cap (\omega_2 \notin C_j)\right) \cup \left(\omega_1 \in C_j \cap (D &gt; T)\right)\right), \ \text{reject} &amp; \text{otherwise} \end{cases}$</td>
</tr>
<tr>
<td>$C_j$ represents numeral class $j$, $j \in {0, 1, \ldots, 9}$, $d(x, \omega_j)$ is the distance between the test pattern and the $j$th cluster, $D =</td>
</tr>
</tbody>
</table>

The decision rule for the BP Neural Network

$$d(x) = \begin{cases} j & \text{if}\left(\left(\omega_j = \max_{i \neq j} \omega_i\right) > T_1\right) (\max_{i \neq j} \omega_i < T_2), \\ \text{reject} & \text{otherwise} \end{cases}$$

The test pattern is assigned to class $j$, where $\omega_j$ is the maximum of the ten outputs, provided $\omega_j$ is larger than $T_1$ and $\omega_i(i \neq j)$ is less than $T_2$. |
A hierarchical neural network architecture

![Image of handwritten numerals](image)

Fig. 5. Preprocessing steps: (a) slant correction; (b) smoothing; (c) gray scale mapping.

![Image of digit mapping to neural network](image)

Fig. 7. Architecture of the BP neural net.

![Graph showing performance of the BP neural net](image)

Fig. 8. Performance of the BP neural net.

![Graph comparing performance of BP net and proposed system](image)

Fig. 9. Relative performance of a BP neural net and the proposed system.

4. CONCLUSIONS

We have presented a Bayes incremental learning neural network classifier for handwritten recognition. By using a Principal Component Analysis neural net as a feature extraction sub-net, the dimension of the input vector is significantly reduced and the de-correlation of data vector is performed efficiently. With uncorrelated and small size input vector, the algorithm of Bayes learning is significantly simplified and hence, improves the learning speed.

Experimental results with a large real world data set show that the proposed system outperforms the back-propagation learning neural network. However, it should

values of $T_2$. This figure enables the selection of optimal thresholds for a specified error rate. Figure 9 shows the comparison between the BP neural network and our proposed system. The optimal error rate to rejection rate curve of BP net is used in this comparison. Notice that in the lower rejection rate range, the proposed system outperforms the BP neural net, while both classifiers perform similarly in the high rejection range. Error rates below 1% with 10–15% rejection are easily achieved.

Due to the unsupervised nature of the PCA net and BICN net, the proposed system has a faster learning rate than the BP net. In fact the required learning time for the proposed system is only minutes in comparison with days for the BP net.
be noted that the test results are based on a fairly low resolution (8×8) data vectors. By increasing the dimension of the data vector, the recognition performance is expected to be further improved.

Acknowledgements — This work was partially supported by the Natural Sciences and Engineering Research Council of Canada and the MICRONET (A Canadian Network of Centers of Excellence) and Michigan Research Excellence and Economic Development Fund (REEDF).

REFERENCES


About the Author — JUN CAO received his B.Eng. in Radio Engineering and M.Eng. in Communication and Electronic Systems from Beijing University of Post and Telecommunications, People’s Republic of China, in 1982 and 1985, respectively. He received his Ph.D. from the Department of Electrical Engineering, University of Windsor in 1995 and is now employed at C-CUBE in California.

About the Author — M. AHMADI received B.Sc.(EE) degree from Arya Mehr University in Tehran, Iran, and Ph.D. degree from Imperial College of London University, London, U.K., in 1970 and 1977, respectively. He has been with the Department of Electrical Engineering, University of Windsor, Canada, since 1980 and holds the rank of Professor. His research interests include design, stability and realization of 2-D filters, Pattern Recognition and Machine Vision. Dr Ahmadi co-authored the book on Digital Filtering in 1-D and 2-D, Design and Applications, published by Plenum, 1989. He has published over 230 articles in the above area. Dr Ahmadi is a Fellow of the Institution of Electrical Engineers in England (FIEEE) and is a Senior of IEEE.

About the Author — M. SHRIDHAR received his M.Sc. in Electrical Engineering from Polytechnic Institute of Brooklyn, New York, in 1968, and Ph.D. in electrical engineering from the University of Aston in Birmingham, U.K., in 1970. From 1969 to 1985, Dr Shridhar was a faculty member in the Electrical Engineering Department of the University of Windsor, Canada. Since 1986, he has been with the University of Michigan-Dearborn where he is currently a Professor and Chairman of the Electrical and Computer Engineering Department. His research interests are in the area of image processing, pattern recognition and computer vision.