CHAPTER 6

CLASSIFIER DESIGN

As discussed in the privous chapter, feature extraction process will help in highlighting the characteristics of a connected compoent (glyph) which are similar in all the glyphs similar to that and differ significantly from the features of glyphs different from it. The step to follow the feature extraction is to use these features to identify the glyph. In other words, we have to classify the glyph as one of the known ones.

In this chapter we have discussed mathematical basis of the classification and two of the classification techniques viz. Nearest Neighbour Classifier and General Regression Neural Network. The work on general regression was done in collaboration and has been published as a joint publication [15]. A part of it is also included in the thesis of Yajnik. A. [47].

6.1 Introduction

Classification means assignment of a class or category label to given data, here, an image. The essence of pattern classification is to devise a mechanism for learning the mapping from training set * which does this assignment and then use the resulting knowledge to classify any given pattern as per the learnt categories [7].

There are many different ways to represent pattern classifiers. One of the most useful is in terms of a set of real valued *discriminant functions* $g_i(x), i =$

*a set of samples where class labels are known

$1, 2, \cdots, c$ see [11]. The classifier is said to assign a feature vector $\mathbf{x}$ to class ω_i if

$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \quad \text{for all } j \neq i. \tag{6.1}$$

The effect of this decision rule is that it divides the feature space into c decision regions, $\mathcal{R}_1, \ldots, \mathcal{R}_c$. The regions are separated by *decision boundaries* or surfaces in the feature space [11]. In case of a tie, additional information is used to classify an image uniquely.

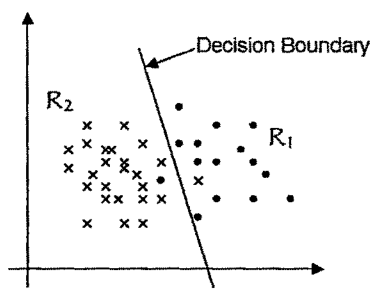

Figure 6.1: Decision Regions for Two-Class Classification Problem

It is important to note that it may not be possible to assign correct class label to all possible data sets and it may be assigned label of some other class. In other words, this data sets is *confused* with the other one. This may be due to various reasons like limitations of feature extractor to extract the minor differentiating features, limitation of the classifier to use that differentiating features. Objects that produce such confused data sets constitute *confusion set.*

Confusioı Set

It is important to study these sets to devise strategy to resolve such confusion and thereby improving the performance of OCR beyond a certain extent.

6.2 Mathematical Preliminaries

In this section we have discussed necessary mathematics.

6.2.1 Artificial Neural Networks

An Artificial Neural Network (ANN) is a massively parallel-distributed processor made up of simple processing units that have a natural propensity for storing experiential knowledge and making it available for use [20]. It resembles the brain in the way in which knowledge is acquired by the network from its environment through a learning process and inter neuron connection strengths, known as synaptic weights, are used to store the acquired knowledge. The computational units in an ANN are called *neurons*. Like its biological counter part neurons in the ANN are also connected and the strength of the connection is modeled by the weight on each of this connections. Diagrammatically the neural network may be represented as a directed graph as in Fig. 6.2 Neuron

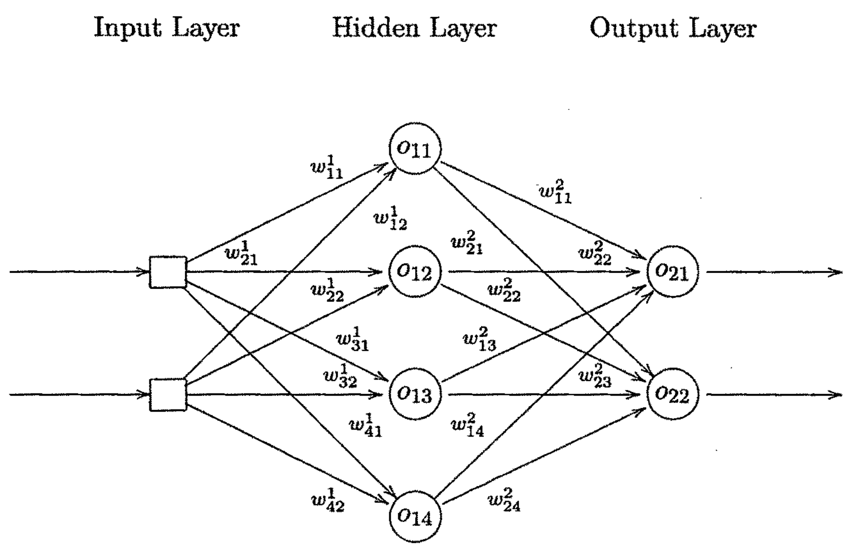

Figure 6.2: Artificial Neural Network

In Fig. 6.2 the circle represents the processing unites in the ANN. A typical ANN can have neurons arranged in different layers viz. an input layer, one or more hidden layers and an output layer. Input layer neurons (shown as squares) could be a non processing neurons i.e. they pass the input to the next layer without any change of the value. The neurons in the layers other than input layer are processing neurons. w^k_{ij} gives weight on the link between i^{th} neuron of k^{th} layer and j^{th} neuron of the $(k-1)^{th}$ layer. Output of i^{th} neurons in k^{th} layers other than input layer, o_{ki}, can be given by

$$o_{ki} = \mathcal{T}\left(\sum_{j=1}^{n} w^k_{ij} x_j\right) \tag{6.2}$$

where, x_j is the output from the j^{th} neuron in the $k-1$ layer and $\mathcal{T}$ is known as *transfer function*. Transfer Function

This type of networks where the output of neurons in one layer is used to compute the output of neurons in the next layer and which does not have any cycle / loop are called *feed forward networks*. Multi Layer Perceptron (MLP), Feed Forward Network

Radial Basis Function(RBF) Networks are commonly used example of this feed forward type of networks. ANNs with a cycle in their graph representation are called *recurrent networks*. Recurrent Network

Learning or *training* is the process by which ANN adapts itself to exhibit the relationship between input and output in the data used for this purpose. This is achieved through one of the learning algorithms. Most of this algorithms can be viewed as an implementation of optimization process where in they try to minimize the difference (say e_i) between the computed output at each of the output neuron (say c_i) and the known output, say y_i. Mathematically, Learning

$$\text{minimize } E = \sum_{i=1}^{n} (c_i - y_i)^2, \quad \text{where } n \text{ is total number of outputs} \qquad (6.3)$$

The data set with known output for the given set of inputs is called *training set*. The set of input and corresponding output is referred to as a *pattern* in machine learning. At each iteration of the learning algorithm input data from one of the patterns being trained is fed to the input neurons and output is computed at the output neurons by doing the computation at each processing neurons as mentioned in the eq. 6.2. Then the weights are updated according to the rule specified in the algorithm to achieve targeted error minimization. Once the error reaches below the allowable tolerance limit the training is stopped and the network is called *trained* at that stage. One of the most popular algorithms to train MLPs is called *error back propagation* algorithm. The performance of the trained network is checked against the *test set*, the set of patterns not used for training purpose. Training Set Pattern Test Set

6.3 Classifier

Researchers have tested various classification techniques while developing OCR for Indic scripts / languages. [29] documents the use of nearest neighbor classifier for Telugu OCR where as first effort for Gujarati character recognition by Sameer Antani *et. al.* employed K-Nearest Neighbor (K-NN) classifier. Use of Support Vector Machine (SVM) Based hierarchical classifier for Malayalam OCR is documented in [28]. In our studies we have mainly focused on nearest neighbor classifier and we have compared the result with a variant of Artificial Neural Network (ANN) viz. General Regression Neural Network.

6.3.1 K-Nearest Neighbor Classifier

In lay man sense the K-NN classifier finds K most similar patterns to the test sample from the training set. The similarity between testing and training samples is determined through a distance measure. The most commonly used distance measure is Euclidean distance between two vectors $p(x_1, x_2, \ldots, x_n)$ and $q(y_1, y_2, \ldots, y_n)$ in an n-dimensional space which is defined as

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \tag{6.4}$$

The training of K-NN involves storing the features of the training samples along with their identifiers in a file. It is achieved through following steps :

1. Compute feature vector,say $\mathbf{F_{test}}$, from the test sample in the same way as the features for training samples were computed.

2. Compute distance d_i of this features from each of the feature vectors $\mathbf{F^i_{train}}$ of the training samples

$$d_i = distance(\mathbf{F_{test}}, \mathbf{F^i_{train}}) \tag{6.5}$$

3. Arrange the distances with corresponding category labels in increasing order of distance.

4. Record the labels of first K entries in the sorted list.

5. Perform a voting to find the class / category label of the test sample.

Here, if $K = 1$ then this method is known as *nearest neighbor classifier.*

Nearest Neighbor Classifier

6.3.2 General Regression Neural Networks

Most of the researchers employ Neural Network Architectures using iterative learning techniques like Back propagation [5], [38], [46], RBF [21], Dynamic Neural Networks (DNN)[33]. The basic disadvantage of these techniques is that the algorithms take a large number of iterations to converge to desired solution. GRNN, in comparison, is single pass neural network architecture, and converges much faster to the desired solution. General Regression Neural Network (GRNN) was first proposed for speech recognition [3], [2] and used there with good success. However, it is for the first time that we have used it as a classifier for Gujarati Character Recognition.

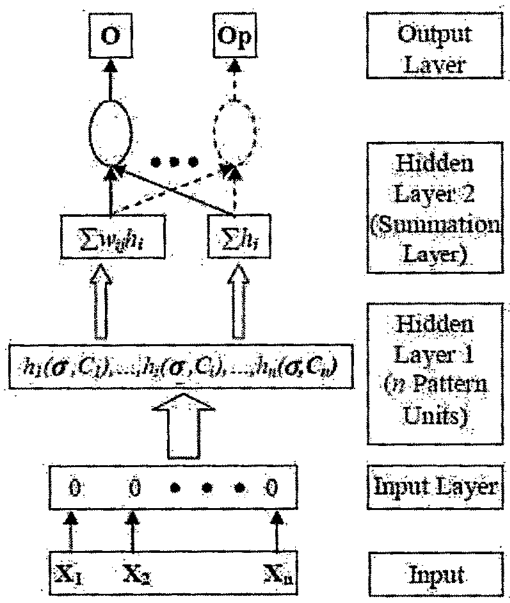

Figure 6.3: GRNN Architecture [2]

GRNN can be described as follows [37]:
Let w_{ij} be the desired output corresponding to input training vector X_i and j^{th} output. Then,

$$y_j = \frac{\sum_{i=1}^{n} w_{ij} h_i}{\sum_{i=1}^{n} h_i}. \tag{6.6}$$

$$where, \quad h_i = \exp\left(-\frac{D_i^2}{2\sigma^2}\right). \tag{6.7}$$

$$D_i^2 = (X - X_i)^T (X - X_i), \quad \sigma = spread\ .$$

The estimate y_j can be visualized as a weighted average of all the observed values, w_{ij}, where each observed value is weighted exponentially according to its Euclidean distance from input vector X and n is the number of patterns available in the input space.

According to eq. (6.6) and eq. (6.7) the topology of a GRNN consists of (Fig. 6.3) :

- The input layer (input cells), which is fully connected to the pattern layer.

- The pattern layer which has one neuron for each pattern. It computes the pattern functions which is expressed in (6.7).
- The summation layer has two units N and D corresponding to numerator and denominator of equation (6.6) respectively. w_{ij}, the desired output of j^{th} output neuron of i^{th} training pattern, is multiplied with corresponding exponential term $h_i(\sigma, C_i)$, C_i is a center for i^{th} training pattern X_i. Value of N for j^{th} output neuron is computed by summing this multiplication for all training patterns. Denominator D is computed by considering $w_{ij} = 1$ in the procedure used for computing N.
- Finally the output unit divides N by D to provide the result.

A detailed study of GRNN and its application to Gujarati character recognition is given by Yajnik A. *et. al.* in [47]. In this experiment we have taken $C_i = X_i$. Here, we have compared the performance of GRNN with nearest neighbor classifier and also studied their confusion set to take final decision on selection of classifier for Gujarati character recognition.

6.4 Experiment Setup

To carry out testing experiments we need two data sets for training the classifier and then testing the performance of the trained classifier. It should be kept in mind that we need to create data sets for glyphs in all the three zones of the text line. Also the target document for OCR can come from different sources which may have different fonts and style of the font. We have collected the segmented glyphs from various sources like books printed by different publishers and also some pages printed with modern computerized font and type setting. Further, we scale the testing and training glyphs to a common size of 32×32 pixels.

We are using the zone information as one of the features to limit the search to the glyphs from the same zone as the test glyphs to be classified. In other words, we design and train three different classifiers one for each glyphs of a particular zone.

Further, it may be noted that Gujarati consonant /pa/ has the same shape as the shape of Gujarati numeral 5 and same is the case with consonant /ra/ and Gujarati numeral 2. Hance, we have merged the classes for glyphs in these pairs. That is /pa/ and 5 will not be distinguished at the classifier level and same for /ra/ and 2.

The distance measure used for classifying glyphs with DCT based features is standard Euclidean Distance and for fringe map features we use the special distance measure called Fringe Distance, which is also used for Telugu character recognition in [29]. Fringe distance, FD_i between the feature vector for the test glyph $\mathbf{F_{test}}$ and a feature vector of i^{th} trained glyphs $\mathbf{F^i_{train}}$ is given by

$$FD_i(\mathbf{F_{test}}, \mathbf{F^i_{train}}) = \frac{d(\mathbf{F_{test}}, \mathbf{F^i_{train}}) + d(\mathbf{F^i_{train}}, \mathbf{F_{test}})}{\text{Number of 0s in } \mathbf{F^i_{train}}}, \text{ where,} \tag{6.8}$$

$$d(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{32}\sum_{j=1}^{32} \mathbf{X}[i][j], \text{ for all } i,j \text{ for which } \mathbf{Y}[i][j] = 0 \tag{6.9}$$

We have carried out experiments with the following setup for features and classifiers

1. Fringe Map features with Nearest Neighbor classifier for complete set of glyphs with different proportion of training and testing glyphs
2. Fringe Map features with K-Nearest Neighbor classifier for complete set of glyphs with different proportion of training and testing glyphs
3. DCT with Nearest Neighbor Classification
4. Comparison of performance of NN classifier with GRNN using Wavelets feature a sample study.

6.5 Results and Discussion

Results of various experiments are presented and discussed here. In each experiment, out of total number of samples available some portion is used for training the classifier where as the other is used for testing. Further, for testing the repeatability of experiment for a particular distribution of training and testing is carried out number of times and each time the selection is randomly selected.

1. **Set up - 1**
   Feature : Fringe Map
   Classifier : Nearest Neighbor
   Number of Tests Per Distribution : 10
   Number of Samples: 14438
   Number of Classes:238

Training - Testing Distribution : 70 - 30%	
Training Samples : 10106	Testing Samples : 4332

Table 6.1: Classification Accuracy(in%)

Minimum	Maximum	Average
97.6	98.1	97.8

Training - Testing Distribution : 60 - 40%	
Training Samples : 8662	Testing Samples : 5776

Table 6.2: Classification Accuracy(in%)

Minimum	Maximum	Average
97.4	98.0	97.7

2. **Set up - 2**
   Feature : Fringe Map
   Classifier : K-Nearest Neighbor (K=5) with majority voting
   Number of Tests Per Distribution : 10
   Number of Samples: 14438
   Number of Classes:238

Training - Testing Distribution : 70 - 30%	
Training Samples : 10106	Testing Samples : 4332

Table 6.3: Classification Accuracy(in%)

Minimum	Maximum	Average
95.3	97.3	95.9

Training - Testing Distribution : 60 - 40%	
Training Samples : 8662	Testing Samples : 5776

Table 6.4: Classification Accuracy(in%)

Minimum	Maximum	Average
95.0	97.3	95.5

3. **Set up - 3**
   Feature : DCT (80 coefficients)

Classifier : Nearest Neighbor Classifier
Distance : Euclidean Distance
Number of Classes: 10 (Middle Zone)
Accuracy : 88.43%

4. **Set up - 4 (from [15])**
   Feature : Wavelets (16×16 low-low coefficients)
   Classifier : General Regression Neural Network

Table 6.5: Classification Accuracy

Number of Samples: 4173		Number of Classes: 119 (Middle Zone)
Training Samples	Testing Samples	% Accuracy
2802	1371	97.5

Table 6.6: Classification Accuracy

Number of Samples: 6125		Number of Classes: 200 (Middle Zone)
Training Samples	Testing Samples	% Accuracy
4287	1838	88.7

It is clear from the accuracy results of various combinations of feature extraction and classification methods that fringe map features with nearest neighbor classifier gives better accuracy for large number of glyph classes.

There are some trivial confusion sets like /gha/,/dha/,/tha/, /pa/,/sHa/ etc. due to the similarity in their shape.

6.6 Conclusions

It is shown here that researchers have used various methods for classification for indic script glyphs. Literature until this studies reports the results of ANN and nearest neighbor classifier on only a subset of the total number of glyphs required for designing Gujarati OCR. Here, we have considered full glyph set for our experiment with fringe map features and nearest neighbor classifier as no other classifier showed better accuracy for large number of classes.