National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Recognition of Handwritten Numerals Using Neural Networks

Weiqian Dai

A Thesis

in

The Department

of

Computer Science

Presented in Partial Fulfillment of the Requirements for
the Degree of Master of Computer Science at
Concordia University
Montréal, Québec, Canada

December 1990

© Weiqian Dai, 1990

National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-64673-X

Canada

Abstract

Recognition of Handwritten Numeral

Using Neural Networks

Weiqian Dai

A new recognition system has been developed and implemented to solve the difficult real world problem of handwritten numeral character recognition. In the system Fourier descriptors were used as features and a backpropagation model of neural network as the pattern classifier.

A new backpropagaton learning algorithm for the neural network has been developed and its performance has been evaluated. The new algorithm was derived from a new evaluation function instead of least mean squares. Simulation results show that this new algorithm is superior to the standard backpropagation model. The problem which occurs in the training sequence in which the standard backpropagation model fails is solved by the new algorithm. The newly developed backpropagation model is a promising approach to handwritten numeral recognition.

Acknowledgements

I would like to take this opportunity to express my sincerest thanks to my thesis supervisors, Professor A. Krzyzak and Professor C. Y. Suen, for guidance, invaluable suggestions, criticisms and encouragement throughout the course of this research work.

To all my friends and colleagues in the Department of Computer Science, thank you for your encouragement, thoughtful discussion, and help.

I am also grateful for the financial assistance provided by research grants awarded to Professor A. Krzyzak and Professor C. Y. Suen by the Department of Education of Quebec and the Natural Sciences and Engineering Research Council of Canada.

I would also like to thank my parents, brother and sister for their patience and encouragement.

Finally, many thanks to my wife Jin Lu, for her caring and supporting.

Contents

1 Introduction **1**

1.1 Brain vs. Digital Computer 1

1.2 Modeling Approach to the Neural Network 2

1.3 Pattern Classification and Neural Networks 5

1.4 Introduction to Pattern Recognition System 6

1.4.1 Basic Structure of a Character Recognition System 6

1.4.2 Pattern Recognition System Using Neural Networks 7

1.5 Proposed Character Recognition System 8

1.6 Outline of This Thesis 8

2 Feature Selection **10**

2.1 Fourier Descriptors 11

2.1.1 Zahn and Roskies' Fourier Descriptors 11

2.1.2 Granlund's Fourier Descriptors 13

2.1.3 Reconstruction of Patterns 16

2.2 Internal Shape Descriptors 16

3 Neural Networks – Backpropagation Model **22**

3.1 Fundamentals of Artificial Neural Networks 22

3.1.1 The Biological Neuron 23
3.1.2 The Artificial Neuron 26
3.1.3 Activation Functions 27
3.2 Development of Neural Network 30
3.3 Perceptron and Representation 31
3.4 Multilayered Neural Network Model - Backpropagation Model 32
3.4.1 Representation of the Three Layer Neural Network 34
3.4.2 Multilayered Neural Network Learning Algorithm - Backpropagation Learning Algorithm 35
3.4.3 Implementation of Backpropagation Learning Algorithm . . . 37
3.4.4 Problems with the Backpropagation 39
3.5 Modification of Backpropagation Learning Algorithm 39

4 Simulation Results and Discussions **42**

4.1 Data Collection and Preprocessing 43
4.1.1 Data Collection 43
4.1.2 Data Preprocessing 43
4.1.3 Border Following Algorithm 43
4.2 Rejection Criterion for Numeral Recognition 47
4.3 Experimental Procedure 48
4.4 Results on the Standard Model 48
4.4.1 Number of Hidden Units 49
4.4.2 Number of FD's 49
4.4.3 Results from Training Relatively Large Training Samples . . . 51
4.5 Results from Modified Backpropagation Model 52

4.5.1 False Solution 54

4.5.2 Classification Results 57

5 Conclusions **59**

A Handwritten Character Recognition Verification System **67**

List of Figures

1.1 Block diagram of a recognition system 7

2.1 Polygonal representation of numerals. 12

2.2 Original numerals. 17

2.3 Outer contours of numerals from Figure 2.2. 18

2.4 Reconstructed numerals from Figure 2.2 using Zahn and Roskies' method with 30 FD's. 19

2.5 Reconstructed numerals from Figure 2.2 using Granlund's method with 30 FD's. 20

2.6 Parameters required to extract topological features from a numeral. . 21

3.1 Components of biological neuron. 23

3.2 Artificial neuron. 27

3.3 Artificial neuron with activation function. 28

3.4 Sigmoid function. 29

3.5 Two-layer artificial neural network 32

3.6 Multilayered Neural Network Model 33

3.7 Evaluation function for an output unit having corresponding to $t_{pj} = 0$. 40

3.8 Evaluation function for an output unit having corresponding to $t_{pj} = 1$. 41

4.1 Samples of collected numerals from the U.S. post office. 44

4.2 Recognition rate vs. number of hidden neurons for Granlund FD's alone for 200 training and 200 testing patterns. 50

4.3 Recognition rate vs. number of FD's for Granlund FD's for 200 training and 200 testing samples. 51

4.4 Recognition rate vs. number of hidden neurons for Granlund FD's with topological features for 200 training and 200 testing samples. . . 52

4.5 The false solution for original learning algorithm. 56

4.6 Improved convergence ability obtained with the modified learning algorithm. 56

4.7 Improved classification result from large training set obtained with the modified learning algorithm. 57

A.1 Block diagram of an interactive handwritten recognition verification system 68

List of Tables

4.1 8-neighborhood Pixel x_0 46

4.2 Confusion table for Zahn-Roskies' approach with topological features 53

4.3 Confusion table for Granlund's approach with topological features. . . 53

4.4 Confusion table for Granlund FD's without rejection. 54

4.5 Confusion table for Granlund FD's with rejection. 55

4.6 Confusion table for Zahn - Roskies FD's without rejection. 55

Chapter 1

Introduction

Research on the recognition of the handwritten numerals is a challenging field. A number of handwritten recognition systems have been proposed and tested. The problem of handwritten numerals recognition is both practical and complex. Even though it is easily formulated, a considerable amount of research has been devoted to solve it due to its complexity.

In this work, a new handwritten recognition system is presented, which uses neural network as pattern classifier. The modified back-propagation model is presented and results are discussed.

Recognition of the handwritten numerals is a standard task performed by human. To get an idea how humans do it we have to explore mechanism of the human brain.

1.1 Brain vs. Digital Computer

Conventional digital computers are extremely good at executing sequences of instructions that have been precisely formulated, with the "stored program" representing the processing steps that need to be done. The human brain, on the other hand, performs well at such tasks as vision, speech, information retrieval, and complex spatial and temporal pattern recognition in the presence of noise and distortion – tasks that

are very difficult for sequential digital computers to do.

A Visual pattern recognition task such as reading numerals or distinguishing shapes – a task that can be easily accomplished by human beings – presents significant difficulties to those attempting to design information processors to do the same thing. The solution to this design dilemma apparently resides in the brain itself.

How does the brain accomplish this? The human brain has more than 10 billion neural cells, which have complicated interconnections and constitute a large scale network. Hence, uncovering the neural mechanisms of the higher functions of the brain is not easy. However, the recording can be made from, at most, a few cells simultaneously. Although we can obtain fragmentary knowledge, understanding the mechanism of the network as a whole proves difficult. It is not yet understood how this massively parallel interconnected system of neurons allows us to store, represent, retrieve, and manipulate data such as numeral patterns. Up to now, we do not know how the brain stores the image data and how it learns to recognize the patterns like handwritten numerals. However, we can do this thing well.

1.2 Modeling Approach to the Neural Network

Neurobiologists and neuroanatomists have made substantial progresses. Painstakingly studying the structure and function of the human nervous system, they came to understand much of the brain's "wiring", but little of its operation. As their knowledge grew, the complexity of the human brain was found to be surprising.

Improved understanding of the functioning of the neurons and the pattern of its interconnection have allowed researchers to develop mathematical models to test their theories. Experiments can now be conducted on digital computers without involv-

ing human or animal subjects, thereby solving many practical and ethical problems. From early works it became apparent that these models not only mimic the functions of the brain, but they were also capable of performing useful functions in their own right. Hence, two mutually reinforcing objectives of neural modeling have been defined: first, to understand the physiological and psychological functioning of the human neural system; and second, to produce computational systems (artificial neural networks) that perform brainlike functions. It is the latter objective which forms the major focus of this thesis.

Therefore, a modeling approach using neural network models continues to gain importance, as pointed out by Fukushima [10]. In the modeling approach, there is a need to study how to interconnect neurons to synthesize a *brain model*, which is a network having the same functions and abilities as the brain.

When synthesizing a brain model, it is necessary to follow the physiological evidence as faithfully as possible. For parts not yet clear, however, a hypothesis is constructed and the model should follow the hypothesis. Then the behavior of the model is analyzed and simulated. The simulation results are compared with that of the brain. If any discrepancy is found in the behavior between the model and the brain, the initial hypothesis should be changed and the model should be modified accordingly. The behavior of the model is tested again. This procedure is repeated until the model behaves in the same way as the brain. Although one still needs to verify the validity of the model by physiological experiment, it is probable that the brain uses the same mechanism in the same way.

Once a model has been completed, simplification of the model will make it easy to understand the essential algorithm of information processing in the brain. This

algorithm can be directly used as a design principle for new information processing.

The basic artificial neural networks are biologically inspired: that is, they are composed of elementary functions of the biological neuron. These elements are then organized in a way that may (or may not) be related to the anatomy of the brain. Despite this superficial resemblance, artificial neural networks exhibit a surprising number of the brain's characteristics. For example, they learn from experience, generalize from previous examples to new ones, and abstract essential characteristics from inputs which may contain irrelevant data.

Imagine a computer that learns. Information is fed into it, along with examples of the conclusions it should be reading or feedback on how it is doing - or the machine may even be left on its own. The computer simply runs through the material again and again, making mistakes but learning from them, until finally it attains the proper state to carry out the task successfully. Such behavior is quite human, and naturally so: for the design of the machine's information processing system, a neural network, was inspired by the structure of the human brain–its nerve cells, their interconnections, and their interactions- and by comparing of what the brain can do.

Artificial neural networks have made many impressive demonstrations of its capabilities: a network has been trained to convert text to phonetic representations, which are then converted to speech by other means [34]; and a neural network-based image-compression system has been devised [4]. All these networks use the backpropagation network, perhaps the most successful of the current algorithms.

It has to be noted that the study of the modeling approach to neural networks have been given many terms in the literature: artificial neural networks, connectionist models, parallel distributed processing models, and self-organizing systems. The term

"artificial neural networks" has been adopted as the preference throughout this thesis.

1.3 Pattern Classification and Neural Networks

Pattern recognition is a very broad field that includes various subjects such as character reading, medical diagnosis, weather prediction, and speech recognition.

Recognition of handwritten patterns, after more than three decades, is still a challenge for today's researchers [37, 36]. The difficulty is not only due to the infinite variations and great distortions of characters from one individual to another, but also there are variations from one experiment to another, even when dealing with the same individual.

Numerous character recognition systems have been proposed and implemented in many ways. The recognition techniques vary widely according to the features chosen, the way these features are extracted, and the classification scheme used. Both theoretical and practical aspects of classification schemes have been studied extensively by Suen et al[37].

Because the character recognition problem is related to the replication of human functions, it is natural to consider the neural network approach to the character recognition system.

A number of people have investigated this technique and some researchers have developed several models which are capable of recognizing visual patterns. For a good introduction to artificial neural networks for pattern classifications see Lippmann [22].

1.4 Introduction to Pattern Recognition System

Automatic pattern recognition has been a very active field during the past two decades. The increasing attention given to this research field is due to the demand of large volumes of information processing. The ultimate goal is to develop a flexible system which can classify characters in varying positions, orientations and dimensions in the image plane. Additionally, such a recognition system should be tolerant to a certain degree of random variations in shapes.

1.4.1 Basic Structure of a Character Recognition System

Figure 1.1 shows the block diagram of a basic character recognition system. It consists of the three main components, namely, sensor, image preprocessor and feature extractor, and pattern classifier. Sensing involves the digitization of character image by an optical scanner and binarization if necessary. The image preprocessor and feature extractor can be divided into two stages: preprocessing and feature extraction. Preprocessing mainly involves enhancement and filtering of the digitized image, preparing it for the next stage. In the feature extraction stage, numerical measurements or structural properties representing the characteristics of the input images are extracted. The extracted feature will give maximal information about the pattern. In the selection process we preserve those features which exhibit good discriminatory qualities. The third component of the system involves the classification and identification of the character images into their respective classes. The extracted feature information is fed into a categorizer which makes a classification of the unknown samples.

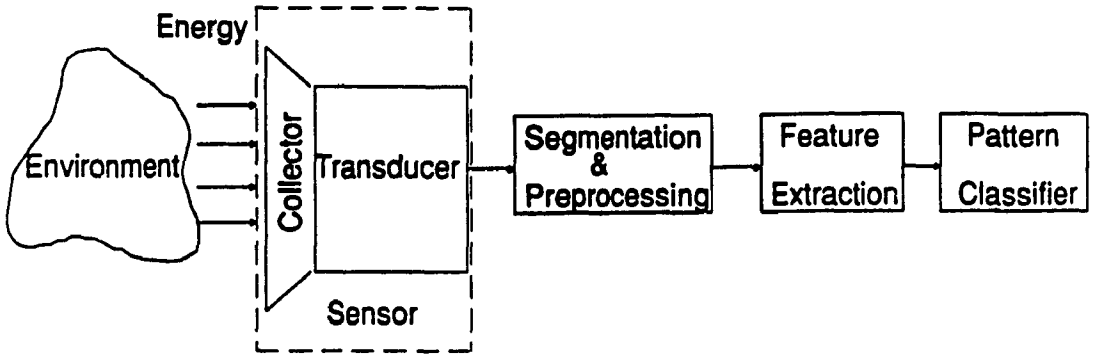


Figure 1.1: Block diagram of a recognition system

1.4.2 Pattern Recognition System Using Neural Networks

In the study of the architecture of neurons in the brain and sensory nerves, it has been suggested [30] that humans perceive shape by performing comparisons to forms already stored in the brain. This comparison is performed invariant of various shape transformations such as scaling of size, translation, and rotation. The neurons can implement operations that are invariant to these transformations [14].

Recent advances in the field of artificial neural nets have opened the door to a new approach to pattern recognition problems. These nets offer several advantages over the traditional recognition techniques, most notably high computation rates and greater degree of fault tolerance provided by massive parallelism.

As mentioned before, traditional image recognition approaches require three phases: image preprocessing: feature extraction to reduce the dimension of the problem to a computationally manageable size and classification of the image using its features.

Our approach shares the first and the second stage with the traditional structures. However, in the classification stage, a neural network is used in place of a conventional classifier. There have been attempts to merge the second and third stages into one when using neural network. This may eliminate the need for explicit feature extraction. However, biological studies of the human visual system suggest that it relies on features extracted from the visual stimuli, e.g. edges and texture content. Therefore, we believe that utilizing features instead of the image itself is an appropriate action. Thus, to investigate any possible gain achieved by using a neural net, one must compare the obtained performance with that of traditional classifiers. The experimental results obtained in our study show that we can actually do better with a neural network which justifies its use in place of a conventional classifier.

1.5 Proposed Character Recognition System

The principal aim of this thesis is to define a new character recognition method which can simulate human ability to learn from the experience (training) and identify the characters automatically. The basic structure of the system includes three parts:

1. preprocessing - noise elimination.

2. feature extractor - extraction of the Fourier descriptor and topological features.

3. pattern classifier - neural network approach to pattern classification.

1.6 Outline of This Thesis

In this thesis, a new handwritten character recognition system has been implemented and tested on a data base of handwritten numerals provided by the US post office.

Chapter 2 explains how the features of characters are extracted and why they are sufficient for the handwritten character recognition system.

Chapter 3 presents the background of a neural network and the history of a backpropagation model. It also explains the necessity to modify the backpropagation learning model for pattern recognition.

Chapter 4 presents and discusses the simulation results for the standard and modified model. The simulation results prove that the modified model is superior to the standard model.

Chapter 5 concludes the thesis and points to the future research directions.

Chapter 2

Feature Selection

Recognition of handwritten characters was and still remains a difficult problem due to the large degree of variability of the data [35]. Not only there is a change and distortion of characters from one individual person to another, but also there are variations from one instance to another, even when written by the same individual. A large number of techniques have been applied to shape recognition problems [20]. Shape analysis, boundary line encoding [8] and polygonal approximation [28] techniques are usually quite simple, however, they are not rotationally invariant. Another related approach involves the reduction of shape information through certain contour scanning or tracing schemes [21]. This approach yields descriptors that are invariant with respect to translation, shift and size. The development of slope and curvature codes for shape description leads to more general concept of intrinsic equation [9], in which the information code along a curve may be considered as a function of the arc length.

Since each character can be, in general, represented by a closed contour of line segments, we can obtain useful information by tracing the boundary of the character periodically. Fourier descriptor are obtained from expanding the boundary or related quantities into Fourier series. Resulting features may be chosen so that they

are invariant with respect to translation, rotation, shift and size of similar shapes [29, 44, 11]. Moreover, these features lead to a significant compression of shape information. In this chapter, two kinds of Fourier descriptors are investigated, and the reconstruction of handwritten numerals from the Fourier descriptor is also discussed.

Unfortunately, Fourier descriptors alone are not sufficient to discriminate all characters for the following reasons:

- Fourier descriptors describe only the outer contour of a binary pattern. Certain numerals, like "7" and "9", "1" and "0", have similar outer contours.
- The rotational invariance property of the Fourier descriptors does create difficulties in dealing with numerals that are similar in shape and their differences can be attributed to rotation and/or reflection, such as 2 and 5, or 6 and 9.

In order to distinguish these ambiguous characters, additional features such as internal topological features (hole), are used.

2.1 Fourier Descriptors

In this section we will present two different types of Fourier Descriptors to encode the outer contour of an image. Training and testing results obtained from these FD's will be presented in Chapter 4.

2.1.1 Zahn and Roskies' Fourier Descriptors

Fourier descriptors given in [44] are defined as follows: assume that γ is a clockwise-oriented simple closed curve with parametric representation $Z(l) = (x(l), y(l))$ where l is the arc length and $0 \leq l \leq L$, see Figure 2.1. Denote the angular direction of γ at point l by $\theta(l)$. Define the cumulative angular bend function $\phi(l)$ as the net amount

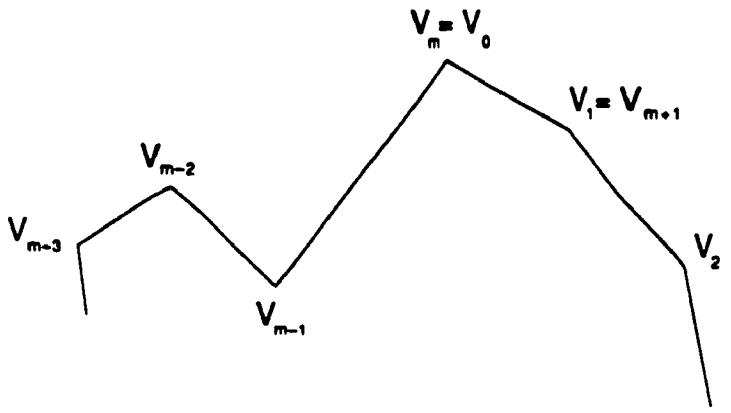


Figure 2.1: Polygonal representation of numerals.

of angular bend between starting point $l = 0$ and point l. So $\phi(l) = \theta(l) - \theta(0)$ except for possible multiples of 2π and $\phi(L) = -2\pi$. Note that if γ winds in a spiral then $|\phi(l)|$ can achieve values larger than 2π. Finally, we define $\phi^*(t)$ as

$$\phi^*(t) = \phi(\frac{Lt}{2\pi}) + t \tag{2.1}$$

where t ranges from 0 to 2π. Note that $\phi^*(t)$ is invariant under translation, rotation, and changes of the perimeter L (scale). Upon expanding $\phi^*(t)$ into a Fourier series we get

$$\phi^*(t) = \mu_0 + \sum_{k=1}^{\infty}(a_k \cos kt + b_k \sin kt). \tag{2.2}$$

In polar form the expansion is

$$\phi^*(t) = \mu_0 + \sum_{k=1}^{\infty} A_k \cos(kt - \alpha_k) \tag{2.3}$$

where (A_k, α_k) are polar coordinates of (a_k, b_k). These number (A_k and α_k) are Fourier descriptors FD's for curve γ.

Suppose that γ is a polygonal curve with m vertices $V_0, ..., V_{m-1}$ and that the edge (V_{i-1}, V_i) has length Δl_i, angular bend at vertex V_i is $\Delta\phi_i$ and $L = \sum_1^{m-1} \Delta l_i$. With these definitions (see Figure 2.1) it is not hard to verify that

$$\phi(l) = \sum_{i=1}^{k} \Delta\phi_i \tag{2.4}$$

$$for \sum_{i=1}^{k} \Delta l_i \leq l \leq \sum_{i=1}^{k+1} \Delta l_i$$

and

$$\mu_0 = -\pi - \frac{1}{L} \sum_{k=1}^{m} l_k \Delta\phi_k \tag{2.5}$$

$$a_n = \frac{-1}{n\pi} \sum_{k=1}^{m} \Delta\phi_k \sin \frac{2\pi n l_k}{L} \tag{2.6}$$

$$b_n = \frac{1}{n\pi} \sum_{k=1}^{m} \Delta\phi_k \cos \frac{2\pi n l_k}{L} \tag{2.7}$$

Where

$$l_k = \sum_{i=1}^{k} \Delta l_i.$$

Normalization

By definition of ϕ^* the amplitudes A_n are invariant to translation, rotation and changes in size. Since α_n's carry information about the starting point, it is useful to normalize the starting point. To normalize the starting point we scan the image of a numeral from left to right and top to bottom and take the first black point as the starting point of the boundary γ.

2.1.2 Granlund's Fourier Descriptors

The FD's given in Granlund [11] are defined in a different way than in [44]. Assume again that γ is a simple, closed curve with representation $Z(l) = (x(l), y(l))$, where l

is the arc length along γ. A point moving clockwise along the boundary generates the complex periodic function $u(l) = x(l) + jy(l)$ with period L. The FD's are defined as

$$a_n = \frac{1}{L}\int_0^L u(l)e^{-j(2\pi/L)nl}dl \tag{2.8}$$

and

$$u(l) = \sum_{n=-\infty}^{\infty} a_n e^{jn(2\pi/L)l} \tag{2.9}$$

We can easily obtain FD's for a polygonal curve. Using the notation of Figure 2.1, we have

$$a_n = \frac{L}{4n^2\pi^2}\sum_{k=1}^{m}(b_{k-1} - b_k)e^{-jn(2\pi/L)l_k} \tag{2.10}$$

where

$$l_k = \sum_{i=1}^{k}|V_i - V_{i-1}|. \tag{2.11}$$

$$b_k = \frac{V_{k+1} - V_k}{|V_{k+1} - V_k|} \tag{2.12}$$

and V_0 is the starting point.

Normalization

In order to normalize the characters for scaling, rotation, position, and the starting point, the coefficients are normalized to a "standard" orientation. The Fourier coefficients can be transformed as follows: $a_0 = 0$ and $a_n' = a_n * se^{j(\phi+n\alpha)}$ for $n \neq 0$. The parameters ϕ and α are chosen so that a_1' and a_{-1}' have zero phases and s is a scale facter which equals to $1/|a_1|$. This is equivalent to rotation and shifting the starting point so that the a_1' and a_{-1}' components describe a fundamental ellipse with its major axis along the x-axis.

There are two problems related to normalization. First, the magnitude of a_{-1} may be zero. In that case the phase angle becomes indeterminate. Second, there are

two possible orientations, which satisfy the zero phase angle of a_1' and a_{-1}' condition. An additional 180° rotation and $-L/2$ starting point shift would also satisfy the zero phase condition. In order to mitigate these difficulties we present alternative normalization schema [12]. A new set of Fourier coefficients a_n' is defined as follows:

- Set $a_0' = 0$.
- $a_n' = \frac{a_n}{|a_1|} e^{j(nt_0+\alpha)}$, for all $n \neq 0$, where t_0 and α are chosen below.

  The above transformation scales and shifts the phase of original Fourier coefficients.

- Find the largest coefficients a_{k^*} in the set $\{a_k, -4 \leq k \leq 6, k \neq 0, 1\}$ and set

$$t_0 = \frac{\phi(a_1) - \phi(a_{k^*})}{k^* - 1}, \tag{2.13}$$

$$\alpha = -\frac{k^* \phi(a_1) - \phi(a_{k^*})}{k^* - 1}, \tag{2.14}$$

  where $\phi(a_k)$ is the phase of a_k.

  The curve $\gamma(l) = a_1 e^{j2\pi l/L} + a_{k^*} e^{j2\pi k^* l/L}$ has $|k-1|$ fold symmetry. So there are $|k^* - 1|$ relative starting point shifts, multiples of $\frac{2\pi}{k^*-1}$ that satisfy the zero phase condition. Hence, rotate and shift the starting point so that

$$\bar{a_n} = a_n' e^{jn\frac{2\pi m}{k^*-1} - j\frac{2\pi m}{k^*-1}}, m = 0, 1, ..., (|k^* - 1| - 2). \tag{2.15}$$

  to maximize the following criterion:

$$\sum_n Re\{\bar{a_n}\}|Re\{\bar{a_n}\}|. \tag{2.16}$$

  This criterion effectively chooses the normalization that orients the axis of one of the main lobes of the $|k^* - 1|$-fold functional shape $a_1 e^{jt} + a_{k^*} e^{jk^* t}$, where $t = 2\pi l/L$, along the positive x-axis and the starting point on the contour corresponding to that lobe is the farthest point from the origin.

2.1.3 Reconstruction of Patterns

One of the main advantages of the Fourier analysis of characters is significant data compression. We have found that for almost all of the characters analyzed in this study, only 30 FD's had significant values, while the remaining coefficients were negligibly small.

Figure 2.2 illustrates the original and Figures 2.4 – 2.5 reconstructed characters using the above normalized Fourier descriptors. The 30 normalized FD's were used to reconstruction of numerals. In figure 2.4, 15 pairs of A_k, α_k were used. In figure 2.5, 15 complex harmonics were taken. It is clear that the reconstructed numerals preserve the inherent features of original characters.

2.2 Internal Shape Descriptors

Topological properties are useful for global descriptions of regions in the image plane. Simply defined, topology is a study of the properties of a figure that are not affected by any deformation, as long as there is no tearing or joining of the figure. These are sometimes called rubber-sheet distortions. If we define as a topological descriptor the number of holes in the region, it is evident that this property will not be affected by stretching or rotation. Topological features are determined from the internal holes described by two numbers, the minimum and maximum distance from the top of the numeral to the hole taken relatively to the height of the numeral (see Figure 2.6).

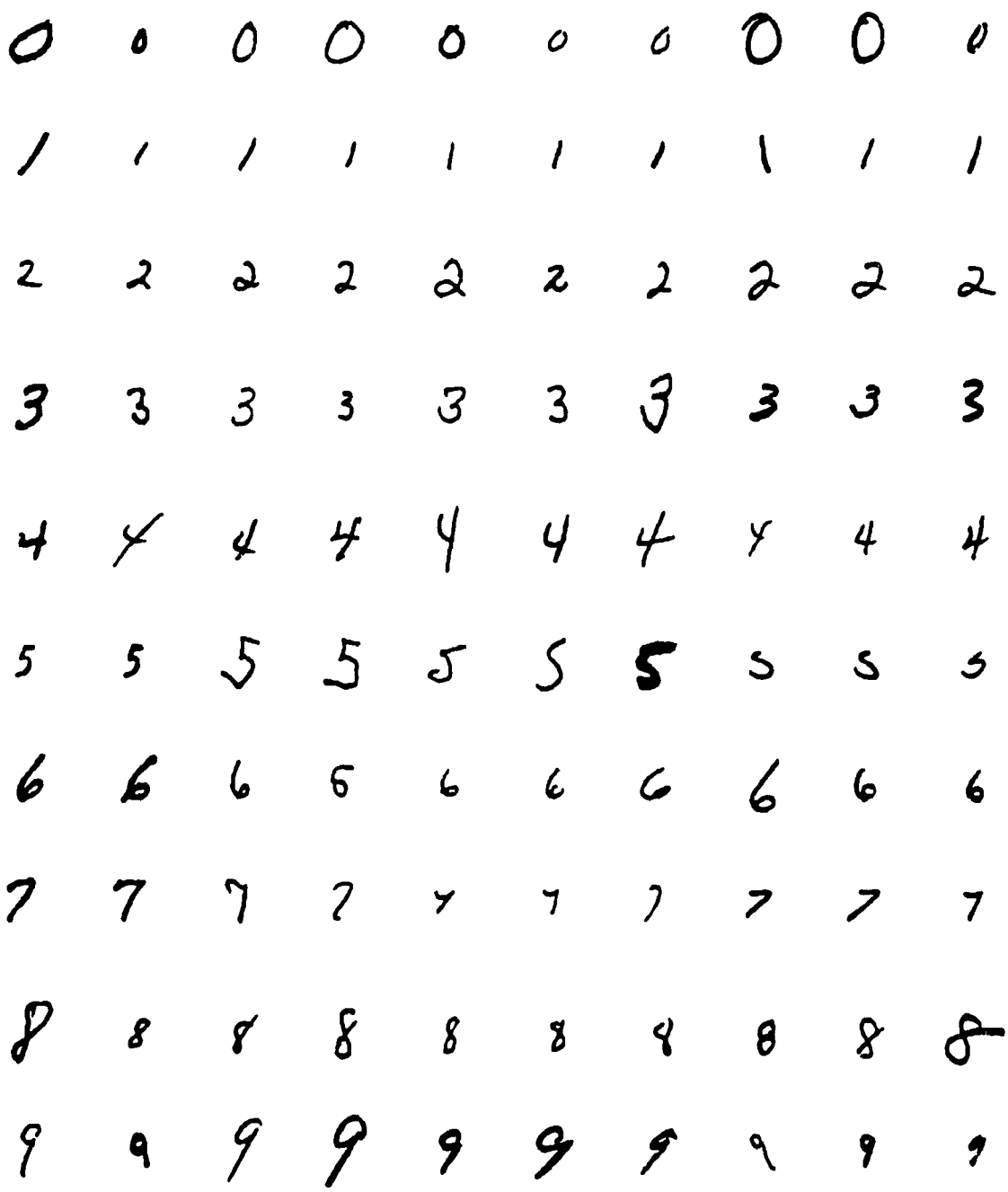


Figure 2.2: Original numerals.

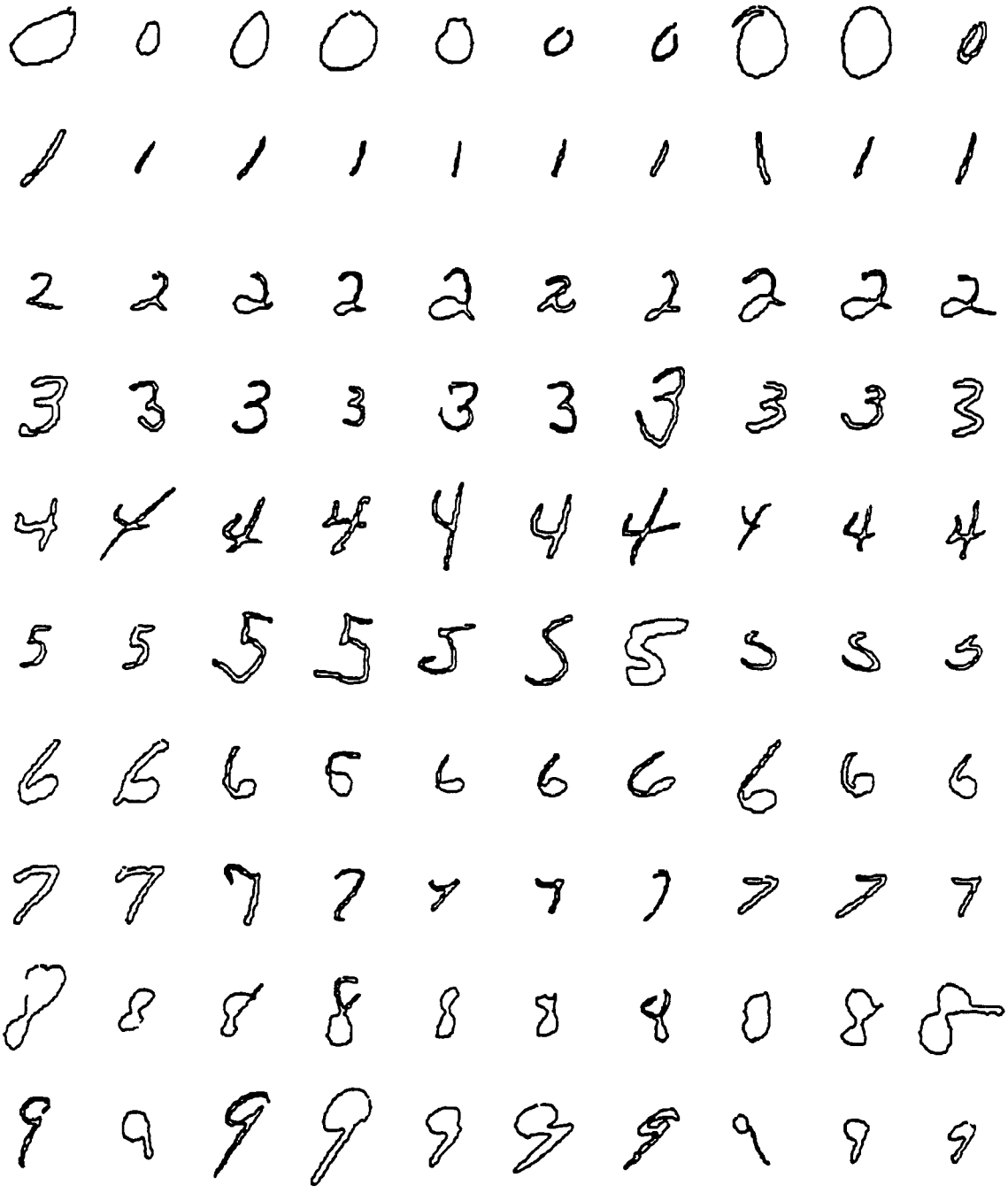


Figure 2.3: Outer contours of numerals from Figure 2.2.

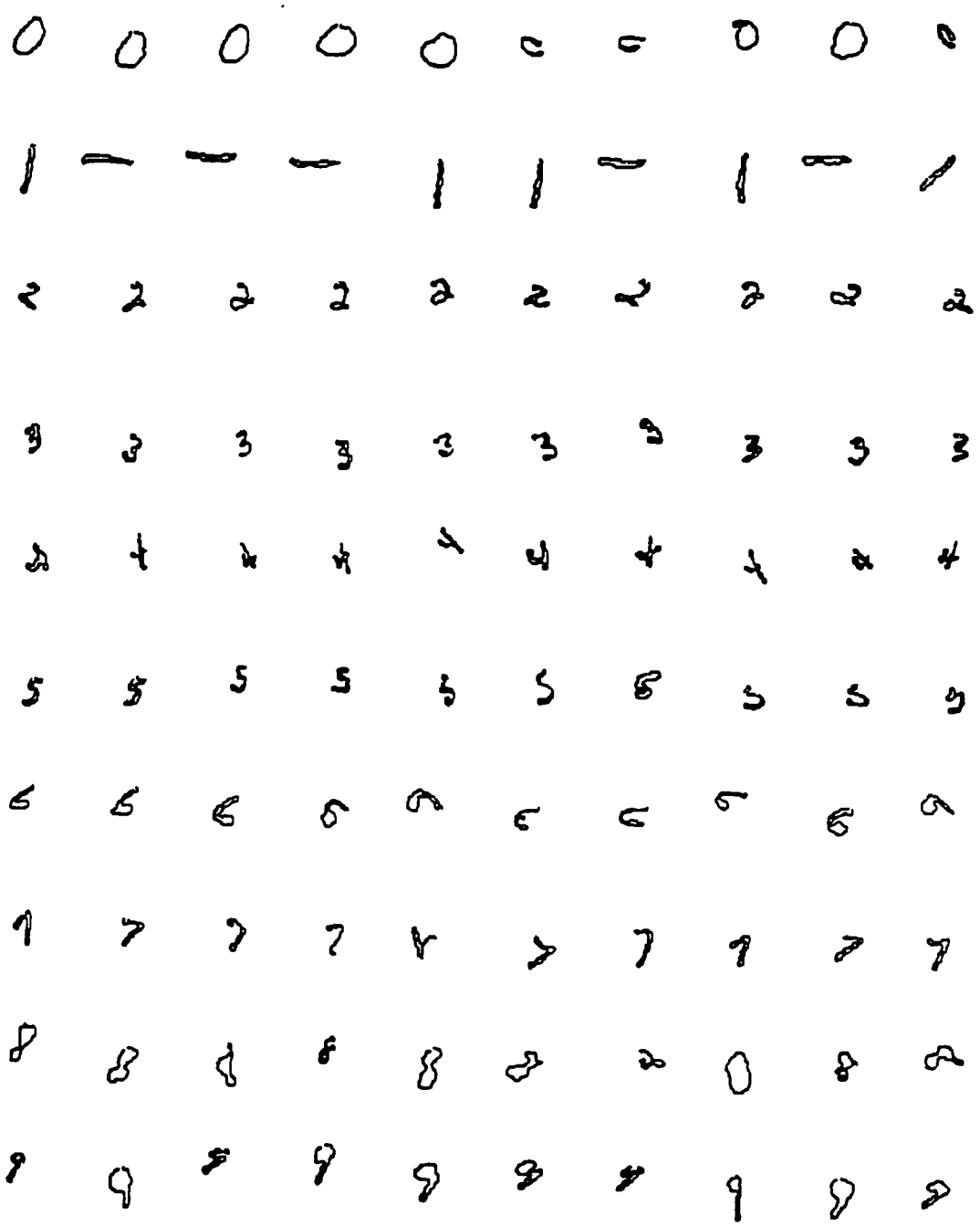


Figure 2.4: Reconstructed numerals from Figure 2.2 using Zahn and Roskies' method with 30 FD's.

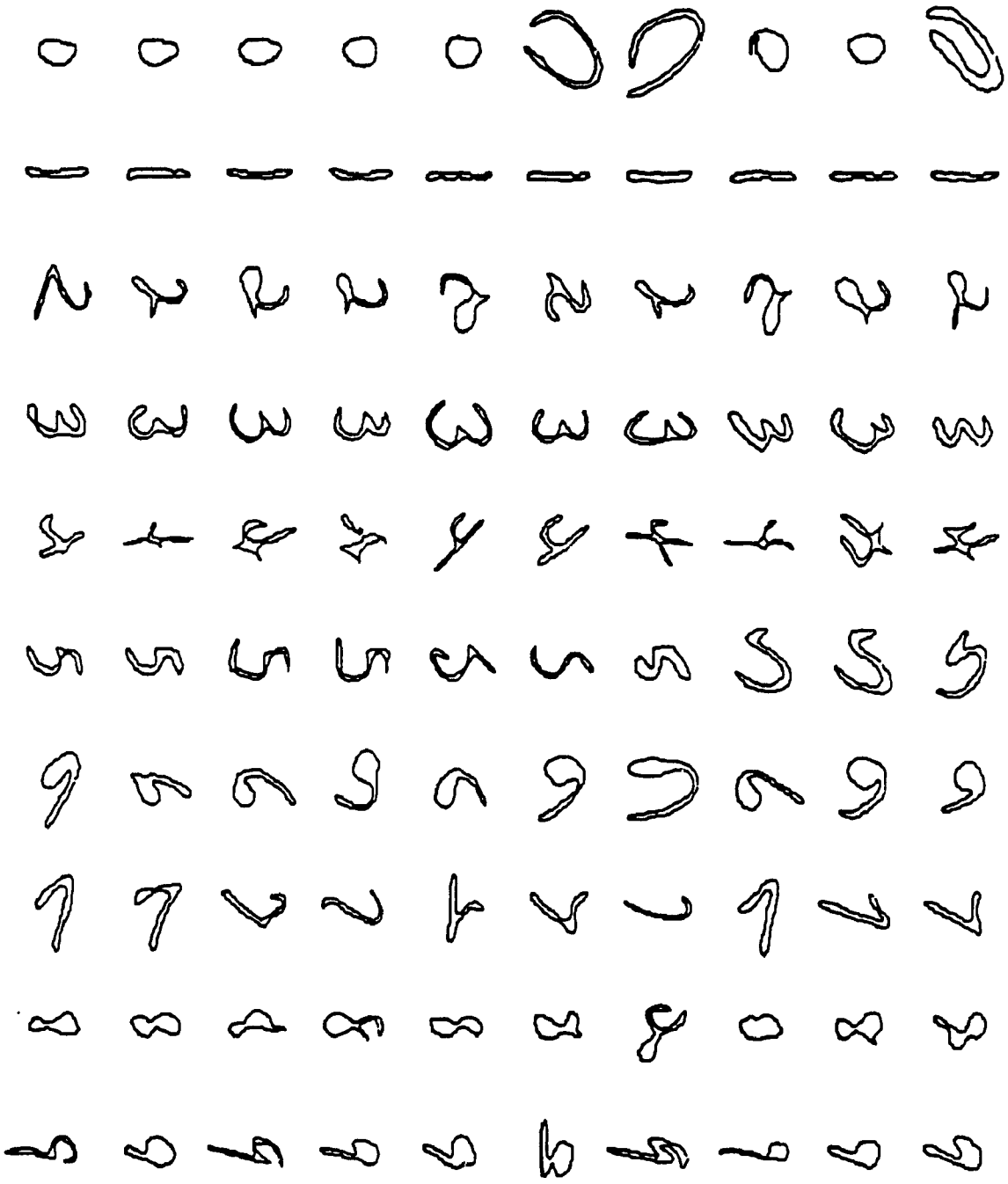


Figure 2.5: Reconstructed numerals from Figure 2.2 using Granlund's method with 30 FD's.

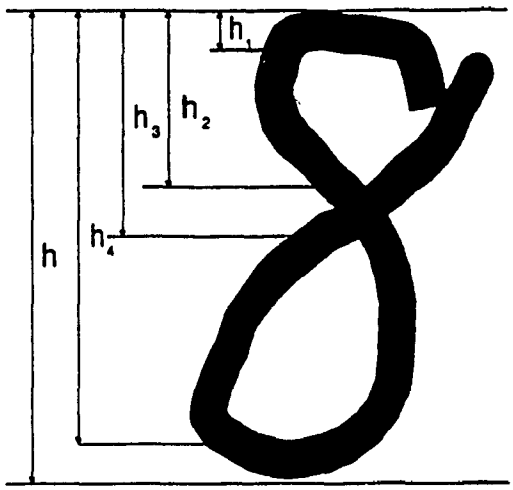


Figure 2.6: Parameters required to extract topological features from a numeral.

Chapter 3

Neural Networks – Backpropagation Model

In this chapter, the basic knowledge of the artificial neural networks is presented, followed by a review of development of the perceptron model. The backpropagation model, adapted in this work, is then discussed in detail. At the end of this chapter the modified backpropagation model is presented.

3.1 Fundamentals of Artificial Neural Networks

Artificial neural networks are biologically inspired; that is, researchers are usually taking into account the organization of brain when considering network configurations and algorithms. Knowledge about the brain's overall operation is so limited that there is a little guidance for those who try to emulate it. Hence, network designers must go beyond current biological knowledge, to seek structures that can perform useful functions. In many cases, this necessary shift discards biological plausibility; the brain becomes a metaphor; networks are produced that are organically infeasible or require a highly improbable set of assumptions about brain anatomy and functioning.

Before we can start the discussion of the artificial network, the human nervous system is described - an entity that successfully performs the tasks to which our

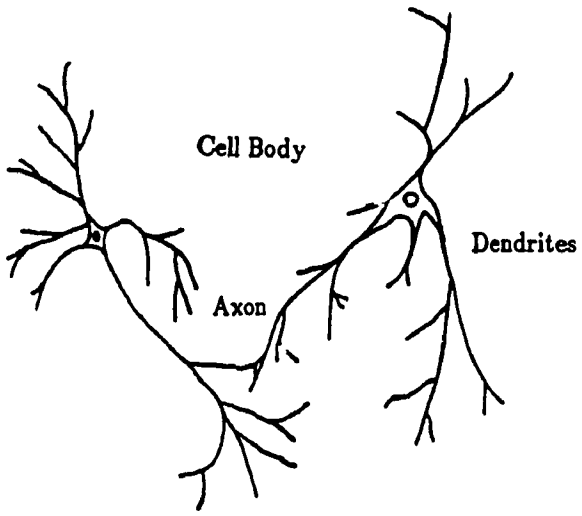


Figure 3.1: Components of biological neuron.

artificial systems only aspire.

The human nervous system, built of cells called neurons, is a very complex system. An estimated 10^{11} neurons participate in perhaps 10^{15} interconnections over transmission paths. Each neuron shares many characteristics with the other cells in the body, but has unique capabilities to receive, process, and transmit electrochemical signals over the neural pathways that comprise the brain's communication system.

3.1.1 The Biological Neuron

The neuron is the fundamental building block of the nervous system. It is a cell similar to all cells in the body; however, certain critical specializations allow it to perform all of the computational and communication functions within the brain.

Figure 3.1, shows the structure of a pair of typical biological neurons. Each neuron consists of three sections: The cell body, the dendrites, and the axon, each

with separate but complementary functions.

Functionally, the dendrites receive signals from other cells at connection points called *synapses*. From there, the signals are passed on to the cell body where they are essentially averaged with other such signals. If the average over a short time interval is sufficiently large, the cell "fires", producing a pulse down its axon that is passed on to succeeding cells. Despite its apparent simplicity, this computational function accounts for most of the known activity of the brain. Underlying it, however, is a complex electrochemical system.

The Cell Body

The neurons in the adult brain do not regenerate; they must last a lifetime. This means that all of the components must be continuously replaced and the materials renewed as needed. Most of these maintenance activities take place in the cell body, where a wide variety of complex molecules are manufactured. In addition, the cell body manages the energy economy of the neuron and regulates a host of other activities within the cell. The outer membrane of the neuron's cell body has the unique capability of generating nerve impulses, a vital function of the nervous system and central to its computational abilities.

Dendrites

Most input signals from other neurons enter the cell by way of dendrites, a bushy branching structure emanating from the cell body. On the dendrites are synaptic connections where signals are received, usually from other axons. In addition, there are numerous synaptic connections from axon to axon, axon to cell body, and dendrite to dendrite. The function of these is little understood, but too widespread to be

insignificant.

Unlike electrical circuits, there is usually no physical or electrical connection made at the synapse. Instead, a narrow gap called the *synaptic cleft* separates the dendrite from the transmitting axon. Specialized chemicals that are released by the axon into the synaptic cleft diffuse across to the dendrite. These chemicals, called neural transmitters, pass into specific receptor sites on the dendrite and enter the cell body.

More than thirty neural transmitters have been identified. Some are excitatory and tend to cause the cell to "fire" and produce an output pulse. Others are inhibitory and tend to suppress such a pulse. The cell body combines the signals received over its dendrites and, if their resultant signal is above its threshold, a pulse is produced that propagates down the axon to other neurons.

The Axon

An axon may be as short as 0.1 millimeter, or it can exceed 1 meter in length, extending to an entirely different part of the body. Near its end, the axon has multiple branches, each terminating in a synapse, where the signal is transmitted to another neuron through a dendrite or, in some cases, directly to a cell body. In this way, a single neuron can generate a pulse that will activate or inhibit hundreds or thousands of other neurons, each of which can in turn (through its dendrites) be acted upon by hundreds or thousands of other neurons. Thus, it is this high degree of connectivity rather than the functional complexity of the neuron itself that gives the neuron its computational power.

Dendrites extend from the cell body to other neurons where they receive signals at a synapse. On the receiving side of the synapse, these inputs are conducted to

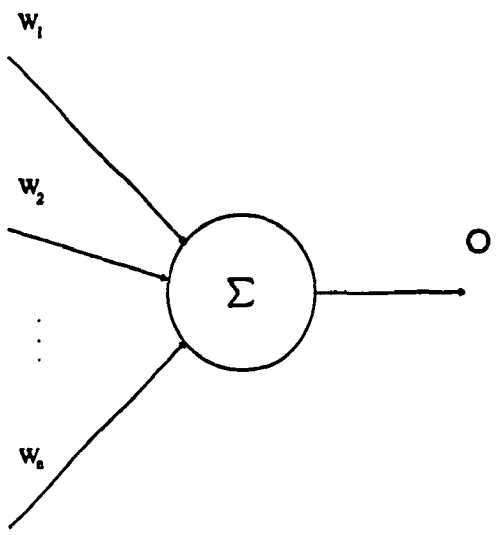
the cell body. There they are summed, some inputs tending to excite the cell, others tending to inhibit its firing. When the cumulative excitation in the cell body exceeds a threshold, the cell fires, sending a signal down the axon to other neurons.

The basic functional outline is very complex and has many exceptions; nevertheless, most artificial neural networks model only these simple characteristics.

3.1.2 The Artificial Neuron

The artificial neuron was designed to mimic the first-order characteristics of the biological neuron. In essence, a set of inputs are applied, each representing the output of another neuron. Each input is multiplied by a corresponding weight, analogous to a synaptic strength, and all of the weighted inputs are then summed to determine the activation level of the neuron. Figure 3.2 shows a model that implements this idea. Despite the diversity of the network paradigms, nearly all are based upon this configuration. Here, a set of inputs labeled $x_1, x_2, ..., x_n$ is applied to the artificial neuron. These inputs, collectively referred to as the vector $\mathbf{x}$, correspond to the signals which go into the synapse of a biological neuron. Each signal is multiplied by an associated weights $w_1, w_2, ..., w_n$, before it is applied to the summation block, labeled $\sum$. Each corresponds to the "strength" of a single biological synaptic connection. The set of weights is referred to collectively as the vector $\mathbf{w}$. The summation block, which corresponds roughly to the biological cell body, adds all of the weight inputs algebraically, producing an output that we call *net*. This may be compactly stated in vector notation as follows:

$$net = \mathbf{wx}$$



$$net = x_1 w_1 + x_2 w_2 + x_3 w_3 + ... + x_n w_n$$

Figure 3.2: Artificial neuron.

3.1.3 Activation Functions

The *net* signal is usually further processed by an activation function f to produce the neuron's output signal, o. This may be a simple linear function,

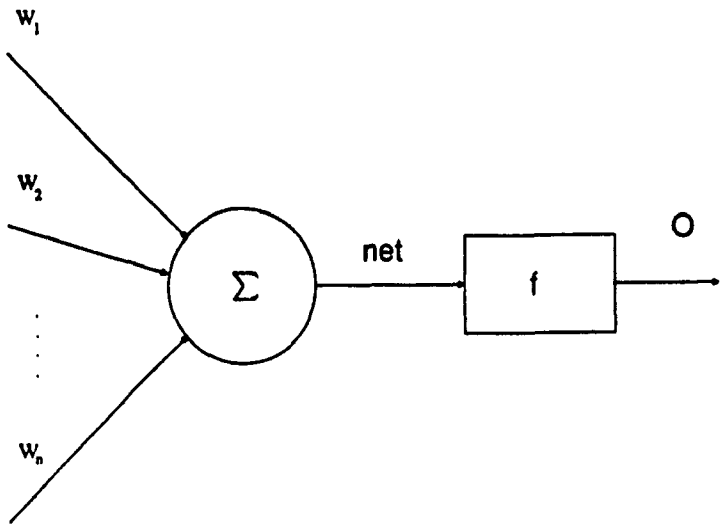
$$o = f(net)$$

where f is a threshold function, and

$$o = 1 \quad if \qquad net > T$$

$$o = 0 \quad otherwise$$

where T is a constant threshold value, or a function that more accurately simulates the nonlinear transfer characteristic of the biological neuron and permits more general network functions.



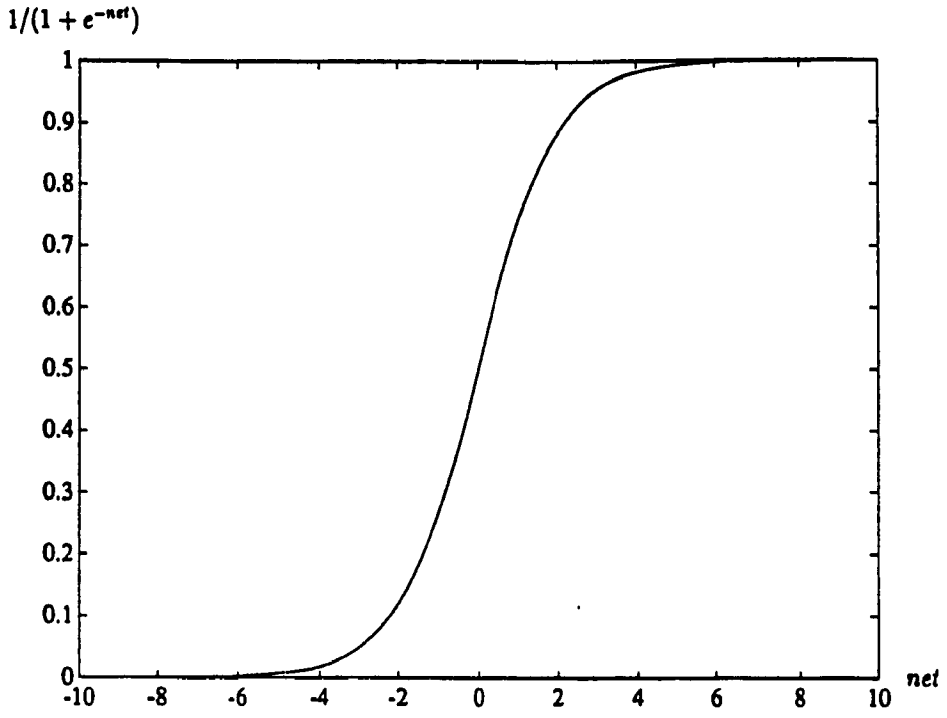
$$o = f(net)$$

Figure 3.3: Artificial neuron with activation function.

In Figure 3.3 the block labeled f accepts the *net* output and produces the signal labeled o. Normally the f processing block compresses the range of *net*, so that o never exceeds a certain preset lower limit regardless of the value of *net*. f is called a *squashing function*. The squashing function is often chosen to be the logistic function or "sigmoid" (meaning S-shaped) as shown in Figure 3.4. This function is expressed mathematically as $f(x) = 1/(1 + e^{-net})$. Thus,

$$o = 1/(1 + e^{-net})$$

By analogy to analog electronic systems, we may think of the activation function as the one which defines a nonlinear gain for the artificial neuron. This gain is calculated by finding the ratio of the change in o to a small change in *net*. Thus, gain is the slope of the curve at a specific excitation level. It varies from a low value at large negative excitation, and it drops back as excitation becomes very large



$$f(x) = 1/(1+e^{-net})$$

Figure 3.4: Sigmoid function.

and positive. Grossberg(1973) found that this nonlinear gain characteristic solves the noise-saturation dilemma that he posed; that is, how can the same network handle both small and large signals? Small input signals require high gain through the network if they are to produce a usable output; however, a large number of cascaded high-gain stages can saturate the output with the amplified noise(random variations) that is present in any realizable network. Also, large input signals will saturate high-gain stages, again eliminating any unusable output. The central high-gain region of logistic function solves the problem of processing small signals, while its regions of decreasing gain at positive and negative extremes are appropriate for large excitations. In this way, a neuron performs with appropriate gain over a wide range of input levels.

This simple model of the artificial neuron ignores many of the characteristics of

its biological counterpart. For example, it does not take into account time delays that affect the dynamics of the system; inputs produce an immediate output. More importa..t, it does not include the effects of synchronism or the frequency modulation function of the biological neuron's characteristics.

Despite these limitations, networks formed from these neurons exhibit attributes that are strongly reminiscent of the biological system. Perhaps enough of the essential nature of the biological neuron has been captured to produce responses like the biological system, or perhaps the similarity is coincidental.

3.2 Development of Neural Network

McCulloch and Pitts (1943) showed how neural-like networks could compute, the main problem then facing researchers in this area was to understand how such networks could learn.

The first idea came from Donald Hebb's [13]. Hebb proposed that a reasonable and biologically plausible change would be to strengthen the connections between elements of the network only when both the presynaptic and postsynaptic units were active simultaneously. The essence of Hebb's ideas still persists today in many learning paradigms. The details of the rules for changing weight may be different, but the essential notion that the strength of connections between the units must change in response to some function of the correlated activity if the connected units still dominates learning models.

Probably the first such attempt occurred in 1951 when Dean Edmonds and Marvin Minsky built their learning machine. It had three hundred tubes and a lot of motors. It needed some automatic electric clutches. The memory of the machine was stored

in the positions of its control knobs, 40 of them, and when the machine was learning, it used the clutches to adjust its own knobs. Although Minsky was perhaps the first to build the network, the real beginnings of meaningful neural-like network learning can probably be traced to the work of Frank Rosenblatt. Rosenblatt invented a class of simple neuron-like learning networks which he called *perceptrons*. In [25] he gives a clear description of what he though he was doing: Rosenblatt pioneered two techniques of fundamental importance to the study of learning in neural-like networks: digital computer simulation and formal mathematical analysis, although he was not the first to simulate neural networks that could learn on digital computers.

3.3 Perceptron and Representation

Perceptron, a two layer neural network, involves only input and output units in which a set of input patterns arriving at the input layer are mapped directly to a set of output patterns at the output layer. As shown in Figure 3.5, there is no *internal* representation. These networks have proved useful in a wide variety of applications. This is what allows these networks to make reasonable generalizations and performs reasonably on patterns that have never before been presented. The similarity of patterns in a system is determined by their overlap. The overlap in such networks is determined outside the learning system itself - by whatever produces the patterns.

The proof of the perceptron learning theorem[31] (Rosenblatt 1962) demonstrated that a perceptron could learn anything it could represent. It is important to distinguish between representation and learning. Representation refers to the ability of a perceptron (or other network) to simulate a specified function. Learning requires the existence of a systematic procedure for adjusting the network weights to produce

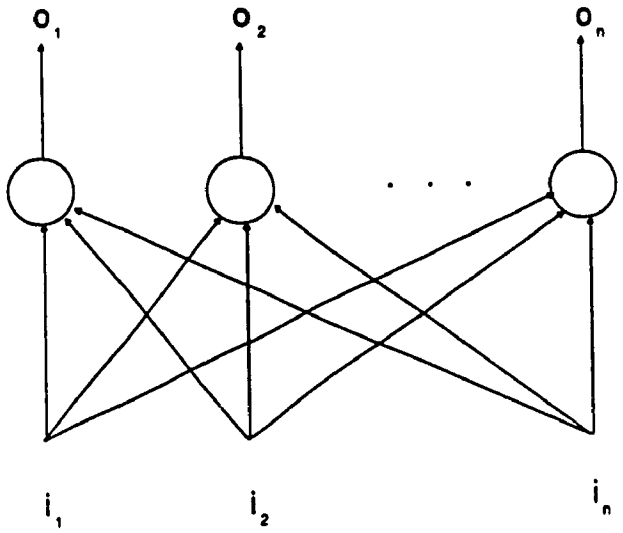


Figure 3.5: Two-layer artificial neural network

that function.

The two-layer network is seriously limited in its representational ability; there are many simple machines that the perceptron cannot represent no matter how the weights are adjusted. Minsky and Papert [25] (1969) have provided a very careful analysis of conditions under which such systems are capable of carrying out the required mappings. They show that in a large number of interesting cases, networks of this kind are incapable of solving the problems.

3.4 Multilayered Neural Network Model - Back-propagation Model

Figure 3.4 shows the neural network model consisting of multiple layers, which was originally proposed by Bryson and Ho [2] and independently rediscovered by Werbos in 1974 [41], by Parker in 1985 [27] and D. E. Rumelhart et al in 1985 [33, 32].

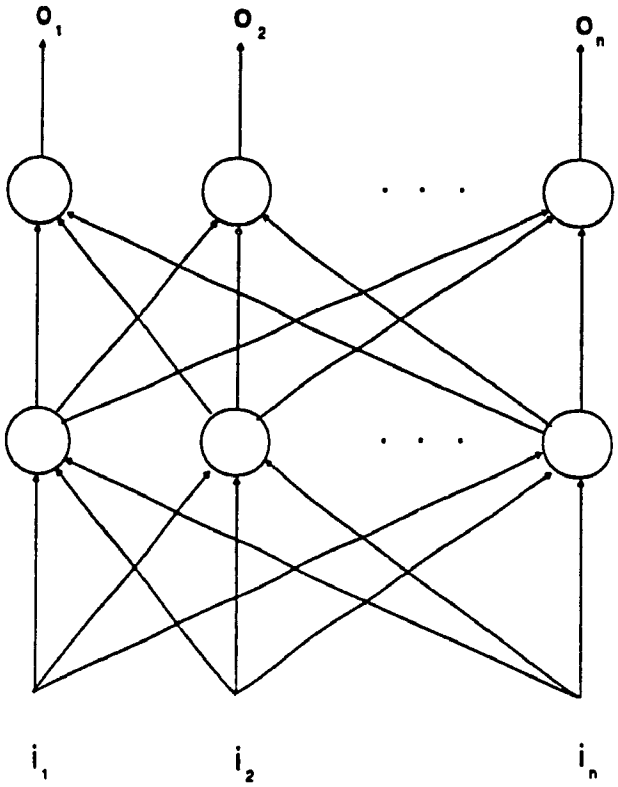


Figure 3.6: Multilayered Neural Network Model

The bottom layer of the network is called the input layer, the upper layer is called the output layer, and the intermediate layers are called the hidden layers. Each layer consists of a number of cells called units. Each element of the input vector corresponds to a simple unit in the input layer. The output signal from the output layer is equivalent to the output of a discriminant function. Each unit is connected to all units in the layers above its own. Each connection has an unbounded positive or negative weight associated with it. The output of the network is a function of the inputs and the weights.

Consider the single neural unit j with n-input, as shown in Figure 3.4. The output o_j of unit j is a function (called the activation function) of the total input fed to that unit,

$$o_j = f(net_j).$$

The inputs to the unit are the weighted outputs from all previous levels. Thus, net_j

is defined as

$$net_j = \sum_i w_{ji} o_i + \theta_j,$$

where w_{ji} is the weight of the connection from the ith unit to the jth unit, o_i is the output of the ith unit, and the summation is over all units feeding into unit j where θ_j is a variable bias with function similar to a threshold. Note that o_j can be defined recursively in terms of its inputs. For the input layer $net_m = i_m$, m is an input unit and i_m is the mth input value.

We use sigmoid function,

$$o_j = f(net_j) = \frac{1}{1 + e^{-net_j}}, \tag{3.1}$$

as an activation function. This activation function has the feature of being non-decreasing and differentiable and its range is $0 \leq o_j \leq 1$. As a result we have a nonlinear classification problem. The training of the network is obtained by the back propagation algorithm described in the next section.

3.4.1 Representation of the Three Layer Neural Network

A number of people have studied the capabilities of multilayered perceptron like neural network model. Nilsson [26] showed that a finite number of points can be divided into two arbitrary sets using three-layered perceptrons. Lippmann [22] conceptually explained that three–layered perceptrons can form convex decision regions. In more recent studies, Huang and Lippmann [15] reported that three-layered perceptrons can also form non-convex decision regions. Irie [16] proved that an arbitrary function can be represented by three–layered perceptrons with an infinite number of computing units if the input–output characteristics of them satisfy some conditions.

3.4.2 Multilayered Neural Network Learning Algorithm - Backpropagation Learning Algorithm

To train the network, we use an extension from the one layer gradient algorithm to a multilayer algorithm, called the back propagation algorithm. The name refers to an iterative training process in which the output error is propagated back through the network and is used to modify the weight values. The error signal E is defined by

$$E = \sum_p E_p = \frac{1}{2} \sum_p \sum_j (t_{pj} - o_{pj})^2, \tag{3.2}$$

where summation is performed over all output nodes j, and o_{pj} is the desired or target value of output t_{pj} for a given input pattern p vector. Once the outputs from the internal units and output units have been calculated for each input pattern p, the direction of steepest descent in parameter space is determined by the partial derivatives of E with respect to the weights and bias in the network,

$$\begin{aligned} -\frac{\partial E_p}{\partial w_{ji}} &= -\frac{\partial E_p}{\partial net_{pj}} \frac{\partial net_{pj}}{\partial w_{ji}} \\ &= -\frac{\partial E_p}{\partial net_{pj}} \frac{\partial}{\partial w_{ji}} (\sum_k w_{jk} o_{pk} + \theta_j) \\ &= -\frac{\partial E_p}{\partial net_{pj}} o_{pi} \\ &= \delta_{pj} o_{pi} \end{aligned} \tag{3.3}$$

and

$$\begin{aligned} -\frac{\partial E_p}{\partial \theta_j} &= -\frac{\partial E_p}{\partial net_{pj}} \frac{\partial net_{pj}}{\partial \theta_j} \\ &= -\frac{\partial E_p}{\partial net_{pj}} \frac{\partial}{\partial \theta_j} (\sum_k w_{jk} o_{pk} + \theta_j) \\ &= \delta_{pj} \end{aligned} \tag{3.4}$$

where we define

$$\delta_{pj} = -\frac{\partial E_p}{\partial net_{pj}}. \tag{3.5}$$

To compute δ_{pj}

$$\delta_{pj} = -\frac{\partial E_p}{\partial net_{pj}} = -\frac{\partial E_p}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial net_{pj}} \tag{3.6}$$

where

$$\frac{\partial o_{pj}}{\partial net_{pj}} = (1 - o_{pj}) o_{pj}. \tag{3.7}$$

When node j is an output node,

$$\frac{\partial E_p}{\partial o_{pj}} = \frac{\partial}{\partial o_{pj}} \frac{1}{2} \sum_j (t_{pj} - o_{pj})^2 = -(t_{pj} - o_{pj}), \tag{3.8}$$

and when j is an internal node,

$$\begin{aligned} \frac{\partial E_p}{\partial o_{pj}} &= \sum_k \frac{\partial E_p}{\partial net_{pk}} \frac{\partial net_{pk}}{\partial o_{pj}} \\ &= \sum_k \frac{\partial E_p}{\partial net_{pk}} \frac{\partial}{\partial o_{pj}} (\sum_i w_{ki} o_{pi} + \theta_k) \\ &= \sum_k \frac{\partial E_p}{\partial net_{pk}} w_{kj} \\ &= -\sum_k \delta_{pk} w_{kj}. \end{aligned} \tag{3.9}$$

Therefore, quantities δ_{pj} can be calculated in parallel for all output units j as

$$\delta_{pj} = (t_{pj} - o_{pj})(1 - o_{pj}) o_{pj}, \tag{3.10}$$

where j refers to a unit in one of the intermediate layers, δ_{pj} can be calculated using (3.7) and (3.9) in (3.6) as

$$\delta_{pj} = (1 - o_{pj}) o_{pj} \sum_k \delta_{pk} w_{kj}, \tag{3.11}$$

where summation is over all units k, which receive signals from unit j. Equation (3.11) shows how the analysis proceeds from the output layer to the previous layers.

Using (3.10) and (3.11) in (3.3) and (3.4), we obtain - the steepest descent direction from a current weight-bias configuration. The weights w_{ji} and biases θ_j are

changed according to

$$\Delta w_{ji}(n) = \eta \sum_{p} \delta_{pj} o_{pi} + \alpha \Delta w_{ji}(n-1) \tag{3.12}$$

$$\Delta \theta_j(n) = \eta \sum_{p} \delta_{pj} + \alpha \Delta \theta_j(n-1), \tag{3.13}$$

where n is equal to the number of epochs (in one epoch all the input training patterns are presented to the network), η is the learning rate, and α is a constant which determines the effect of past weight changes on the current direction of movement in the weight space.

3.4.3 Implementation of Backpropagation Learning Algorithm

The following algorithm describes the learning process.

1. Initialize Weights and Offsets

   Set all weights and node offsets to small random values of magnitude less than 1.

2. Present Input and Desired Outputs

   Present a continuous valued input vector $i_0, i_1, ..., i_{n-1}$ and specify the target outputs $t_0, t_1, ..., t_{m-1}$. If the network is used as a pattern classifier, then all target outputs are set to zero except for the one corresponding to the class of the input. That target output is set to one. On each trial a new input might be present or samples from a training set could be presented cyclically until weights stabilize.

3. Calculate Actual Outputs

Use the sigmoid nonlinearity from above to calculate outputs o_i in the output layer.

4. Adapt Weights

Use a recursive algorithm starting at the output nodes and working back to the first hidden layer. Adjust weights and bias terms by

$$\begin{aligned} w_{ji}(n) &= w_{ji}(n-1) + \Delta w_{ji}(n) \\ &= w_{ji}(n-1) + \eta \sum_p \delta_{pj} o_{pi} + \alpha \Delta w_{ji}(n-1) \\ \theta_j(n) &= \theta_j(n-1) + \Delta \theta_j(n) \\ &= \theta_j(n-1) + \eta \sum_p \delta_{pj} + \alpha \Delta \theta_j(n-1) \end{aligned}$$

In this equation $w_{ji}(n)$ is the weight from hidden node i or from an input to node j at epoch n, o_{pi} is either the output of node i or is an input, η is a gain term, and δ_{pj} is an error term for node j. If node j is an output node, then

$$\delta_{pj} = o_{pj}(1 - o_{pj})(t_{pj} - o_{pj}),$$

where t_{pj} is the desired output of node j and o_{pj} is the actual output for the pattern p. If node j is an internal hidden node, then

$$\delta_{pj} = o_{pj}(1 - o_{pj}) \sum_k \delta_{pk} w_{jk},$$

where k is over all nodes in the layers above node j.

5. Repeat by going to step 2 until the total error E in Eq.3.2 is less than a certain value which depends on the application. In this work, that value is set to 0.4.

3.4.4 Problems with the Backpropagation

Multi-layer networks are quite versatile and can be used in many applications [3, 42]. The strength of neural networks lies in their ability to generate any decision regions as long as the network consists of at least three layers [22]. However, because of digital computer numerical round-off error, the learning process is often impeded, the learning algorithm falls into a false solution, in which all the derivatives in equations (3.3) and (3.4) of standard backpropagation are numerical zero but the error is not small. This stops the iteration learning process.

As pointed out in [1], the problem is that the output units $\delta_{pj} = (t_{pj} - o_{pj})(1 - o_{pj})o_{pj}$ can be numerically zero not only when $o_{pj} \rightarrow t_{pj}$ but also when $o_{pj} \rightarrow 0$ or $o_{pj} \rightarrow 1$. This leads to $\delta_{pj} = 0$ for internal units as well, resulting in all derivatives equal to zero, which means that the network will lose its learning ability.

One way to deal with the problem is to restart the process at randomly chosen starting points [33]. This method is impractical for a larger training data base. An alternative solution is proposed by introducing a new energy function.

3.5 Modification of Backpropagation Learning Algorithm

When neural network is used as a classifier, we expect only one of the outputs to be high corresponding to the input class. The remaining outputs should be low, so that the target value is either 1 or 0 [22]. For given output o_j and the target value $t_{pj} = 0$, the corresponding output unit can be evaluated by

$$-(1 - t_{pj})ln(1 - o_{pj}),$$

(see Figure 3.7). If the target value is $t_{pj} = 1$, then the output unit can be evaluated

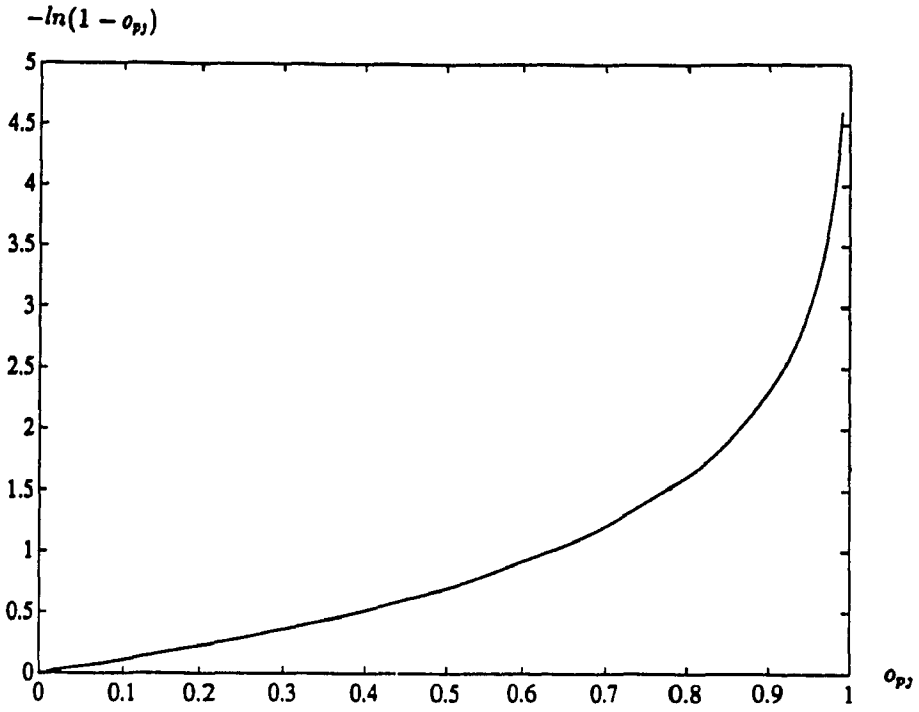


Figure 3.7: Evaluation function for an output unit having corresponding to $t_{pj} = 0$.

by

$$-t_{pj} \ln o_{pj},$$

(see Figure 3.8), where $0 < o_{pj} < 1$. So that, a new energy function could be defined for each pattern by:

$$E_p = -\sum_j ((1 - t_{pj}) ln(1 - o_{pj}) + t_{pj} \ln o_{pj}) \tag{3.14}$$

The above function can be used to evaluate each of the outputs, as well as the entire set of patterns. By substituting this newly defined energy function, Eq.(3.14) into Equation (3.8), we obtain

$$\begin{aligned} \frac{\partial E_p}{\partial o_{pj}} &= \frac{\partial}{\partial o_{pj}} \sum_j -(t_{pj} \ln o_{pj} + (1 - t_{pj}) \ln(1 - o_{pj})) \\ &= -t_{pj} \frac{1}{o_{pj}} + (1 - t_{pj}) \frac{1}{1 - o_{pj}} \end{aligned} \tag{3.15}$$

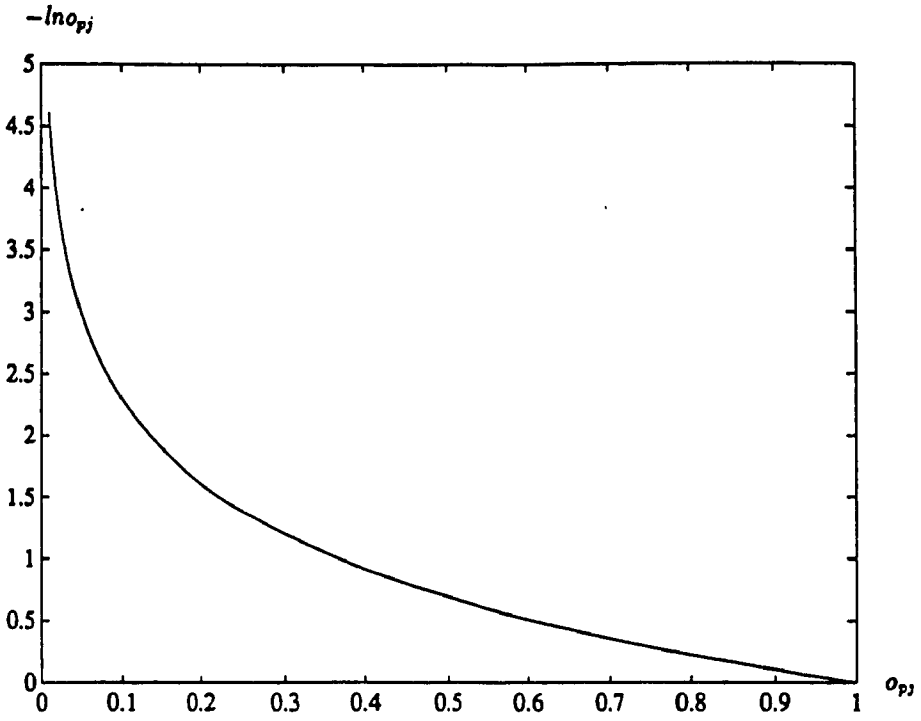


Figure 3.8: Evaluation function for an output unit having corresponding to $t_{pj} = 1$.

For the output node, Equation (3.10) can now be rewritten as:

$$\begin{aligned}
\delta_{pj} &= -\frac{\partial E_p}{\partial net_{pj}} \\
&= -\frac{\partial E_p}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial net_{pj}} \\
&= -(-t_{pj}\frac{1}{o_{pj}} + (1 - t_{pj})\frac{1}{1 - o_{pj}})(1 - o_{pj})o_{pj} \\
&= t_{pj} - o_{pj}
\end{aligned} \tag{3.16}$$

For internal node, δ_{pj} remains unchanged (see Equation (3.11)).

Comparing equation (3.16) with (3.10), one can see that the term $(1 - o_{pj})o_{pj}$has been eliminated. Therefore, the problem of false solution in the original learning algorithm is solved through the introduction of the new energy function, Eq.(3.14).

Chapter 4

Simulation Results and Discussions

In the previous chapters, the use of Fourier descriptors as feature vectors and the neural network as pattern classifier (both standard and modified backpropagation model) have been discussed extensively. In this chapter, these features and the classifier will be used in classification experiments with handwritten numerals. In section 4.1, the collection of the sample data used in the training and testing and its preprocessing will be discussed. In section 4.2, the rejection criterion for the handwritten numeral recognition is defined. Following a brief introduction of the experimental procedure in section 4.3, the simulation results are presented in the following order. First, in section 4.4, a small set of samples are used to train and test different configurations of the neural networks. Second, in section 4.5, a large set of samples are used to train and test the recognition system developed in this work. Finally, in section 4.6, a discussion on the classification results is given.

4.1 Data Collection and Preprocessing

4.1.1 Data Collection

As Le Cun et al [5] pointed out, the performance of the numeral recognition system is highly test-set dependent. A system may successfully recognize 99% of test data consisting of well-formed numerals but score only 80% when confronted with the poorly formed digits that are both routinely produced and easily recognized by people. To avoid such situations, the zip code images collected by the U. S. Postal Service from the dead letter envelopes at different locations in the United States were used as the test data set. Preprocessing was carried out from the original zip codes images to produce 6,000 binary images of isolated digits. Most of the images were fairly clean; however, a significant fraction was distorted. The defects are quite common among "5's", in which the top horizontal stroke is often missing. More discussion of difficult characters in this data base can be found in Mai [23].

4.1.2 Data Preprocessing

The zip code is first segmented into smaller images each of which contains only one numeral. The segmented image is then binarized. A filter is applied to eliminate certain small spots in the image, which could be mistreated as holes during the feature extraction stage.

Each image is further processed by the border following algorithm presented in the following section.

4.1.3 Border Following Algorithm

In the literature on digital picture processing, a number of algorithms can be found for "following" the border of a simply-connected object (e.g. [7, 39, 24]). However,

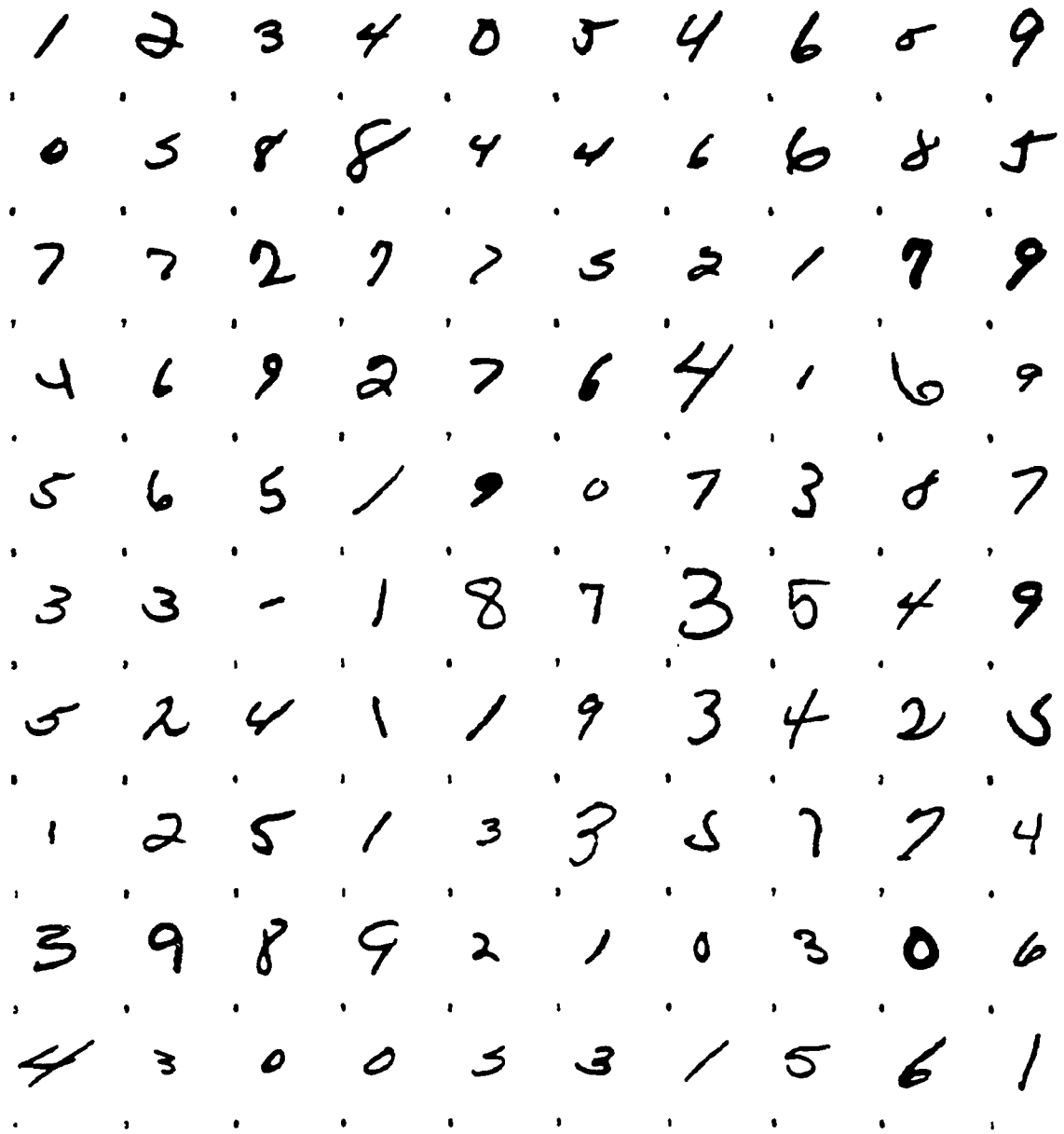


Figure 4.1: Samples of collected numerals from the U.S. post office.

some of these algorithms fail to visit every border element in succession for some simply-connected object.

In this section, a border following algorithm from [43] is presented and used to extract the border of a numeral.

Definitions and Notations

The image of the numeral is assumed to be a rectangular ordered array, with M rows and N columns. Elements of an image with values 0 and 1 are called 0-elements and 1-element, respectively. Let $x_1, x_2, ..., x_8$ denote eight elements in the neighborhood of an arbitrary element x_0 in the image as shown in Table 4.1, and let S, S_1, and S_2 denote three sets of integers $(1, 2, ..., 8)$, $(1, 3, 5, 7)$, and $(2, 4, 6, 8)$ respectively. It is assumed that the first row, the Mth row, the first column, and the Nth column are all filled with 0-elements. These rows and columns are called the frame of the image.

DEFINITION 1 (neighborhood). A set of elements $x_k : k \in S_1$ is called the 4-neighborhood of the element x_0. Similarly a set of elements $x_k : k \in S$ is called the 8-neighborhood of the element x_0.

DEFINITION 2 (connectivity). Two elements a_1 and a_2 with a common value are said to be 4-connected (8-connected), if a sequence of elements $y_0(= a_1), y_1, ..., y_n(= a_2)$ exists, such that each y_i is in the 4-neighborhood (8-neighborhood) of $y_{i-1}(1 \leq i \leq n)$ and all y_i have the same values as a_1 and a_2.

DEFINITION 3 (connected component). Each equivalence class of elements defined by 4-connectedness is called a 4-connected component. An 8-connected component is defined in a similar way. A connected component of 0-elements is called a 0-component, and that of 1-elements is called a 1-component.

x_4	x_3	x_2
x_5	x_0	x_1
x_6	x_7	x_8

Table 4.1: 8-neighborhood Pixel x_0 .

DEFINITION 4 (hole). Any connected component of 0-elements which does not contain the frame of an image is called a hole. A connected component of 1-elements with no hole is said to be simply connected, and otherwise, multiply connected.

The border following algorithm [43] can then be stated as follows,

> Find the starting point of the pattern's outer boundary and the direction of pixel movement by scanning the pattern from left to right and from top to bottom. The first non zero pixel is the starting point y_0 and z_0 is an arbitrary element in the 8-(4-) neighborhood of y_0. Elements are searched for clockwise starting from z_0, in the 4- (8-) neighborhood of y_0 and the first 1-element found is named y_1. Then the search for 1-elements in this neighborhood of y_1 is resumed clockwise from y_0, and y_2 is then chosen in the same way as y_1. This procedure is repeated, and the sequence of 1-elements $y_3, y_4, ...$is determined successively until $y_m = y_0$ and $y_{m+1} = y_1$ hold for the first time. The sequence of 1-elements $y_0, y_1, y_2, ..., y_{m-2}, y_{m-1}$ is called a border line and each of $y_0, y_1, ..., y_{m-1}$ an element of the border line.

Applying the above stated border following algorithm, we construct a border line, consisting a sequence of coordinates of the border elements of a handwritten numeral.

4.2 Rejection Criterion for Numeral Recognition

In practical applications, the user is less interested in the raw error rate than in the number of rejections necessary to reach a given level of accuracy. The appropriate ratio of rejections to substitutions will vary with the application, but, in general, the number of rejects should exceed the number of substitutions [40]. In the course of this project, we measured the percentage of test set that must be rejected in order to get 1% error rate on the remaining test set.

The rejection of patterns is incorporated into the neural network with two rules based on the output values at the output layer. The first rule is to reject a pattern when the highest activation level among the outputs does not exceed the threshold (t_1). The second rule is to reject a pattern when the difference between the highest and the second highest value of the activation is less than (t_2). Otherwise, the pattern will be classified to the class associated with the highest activation level in the output layer.

The rejection thresholds (t_1 and t_2) were obtained from the test set. The performance of the classifier was measured by setting the output unit activation criteria, which must be attained in order to accept a classification. For activations below this level the pattern is rejected as unclassifiable. It has been found that in order to obtain a misclassification rate no higher than 1%, 12.6% of the pattern must be rejected, see Table 4.5. It is expected that a patient human could achieve the same error rate by rejecting about 5% of the patterns [5].

4.3 Experimental Procedure

All the experimental tests were carried out in two stages: Firstly, feature vectors, including both FD's and topological features, were extracted from all the input patterns. All such feature descriptors were stored in a pattern file, for later use. Secondly, a neural simulation program carried out to train and test. The simulation program was developed by the PDP group and modified by myself due to the fact that the original program has the difficulty to incorporate negative values in pattern descriptor as an input pattern. The simulation program was also modified to incorporate the program developed in section 3.5.

Feature extraction was carried out on the SUN 386i and training was performed on the MIPS M120/5 machine which is 4.1 times faster than the SUN. The training of 4000 patterns took about 4 seconds computer time for each sweep. The number of sweeps need to train the network variant according to network structure and training patterns. The network was trained until it stabilized, that means the total error fell below 0.4, a 100% recognition rate was obtained for the training patterns.

4.4 Results on the Standard Model

Results presented in this section were obtained from the simulation on a small set of samples (200 patterns for training and another 200 patterns for testing) and a relatively large set of samples (1000 patterns for training and another 1000 patterns for testing), extracted from our data base (also see [6]). The standard backpropagation model was used as a pattern classifier in this simulation.

There were two goals for this experiment.

- To determine the size of the neural network, i.e. the number of internal units.

- To determine the number of Fourier descriptors in feature vectors.

The tests ware carried out by first extracting the outer boundary of each numeral, then for each numeral a set of FD's were computed. Each pattern was represented by FD's and or topological features. The resulting neural network thus had a variable number of inputs, 10 outputs and a variable number of hidden units. The training of the network was stopped, when the total output error dropped below 0.4.

4.4.1 Number of Hidden Units

Figure 4.2 presents the test results with the recognition rate plotted versus the number of the hidden units. 15 harmonics of the Granlund's FD were used as inputs with the total of 30 inputs for the neural network. A total of 200 patterns (20 patterns from each class of the numerals). were used for the training and additional 200 patterns were used for testing. Figure 4.2 shows the best recognition rate of 75%, which corresponds to 90 hidden neural units.

As discussed in section 3.4.1, three layered neural networks can classify all kinds of mappings when the number of hidden neuron goes to infinity. However, this is not a very practical result. For a given application a finite number of neurons is needed in order to make the mapping possible. Test results in Figure 4.2 show that the minimum number of hidden neurons required was about 50. When the number of hidden neurons was less than 50, the total error of the training patterns never dropped below 0.4.

4.4.2 Number of FD's

Figure 4.3 presents the testing results on the recognition rate in the case of varying number of Granlund FD's used. Again the same number of training and testing

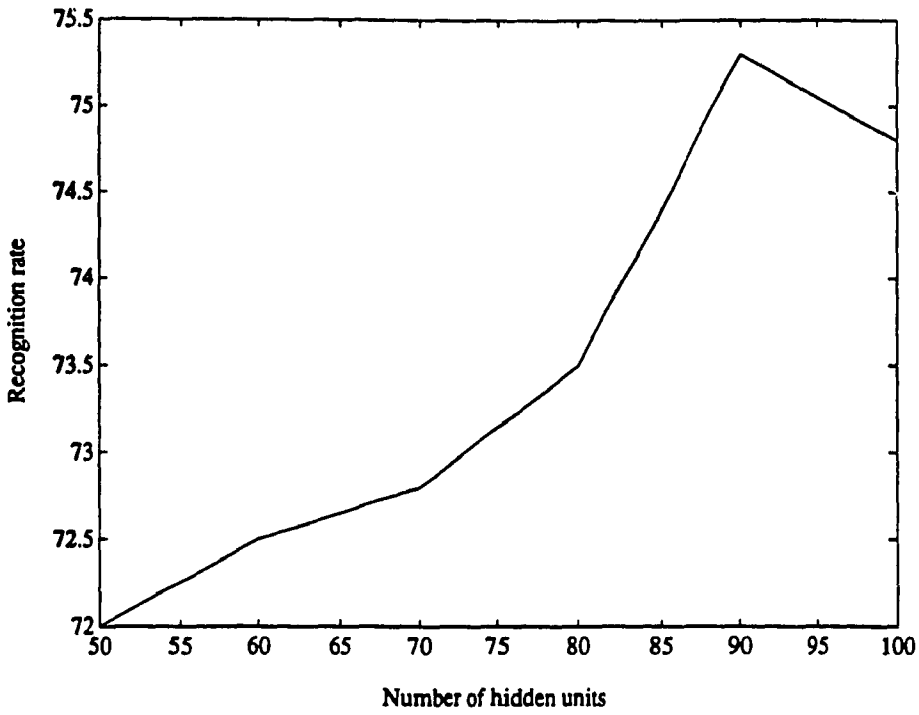


Figure 4.2: Recognition rate vs. number of hidden neurons for Granlund FD's alone for 200 training and 200 testing patterns.

patterns were used as in Figure 4.2. Figure 4.3 shows that the optimal number of FD's is about 20. Figure 4.3 also shows the effect of the learning rate η on the recognition rates.

Pairs which are often misidentified are (6,9) and (1,0). It can be concluded that the FD's alone are not enough to classifier these patterns. The recognition rate can be improved by adding topological features.

Figure 4.4 shows the effect of the number of hidden neurons on the recognition of the feature vectors composed of Granlund FD's and topological features. The same number of training and testing samples were used. The recognition rates are higher than those shown in Figure 4.2 in which only FD's are used. It can be seen, by comparing Figures. 4.4 and 4.2, that the change in the number of hidden neurons has less effect on the recognition rate than that change inclusion or exclusion of

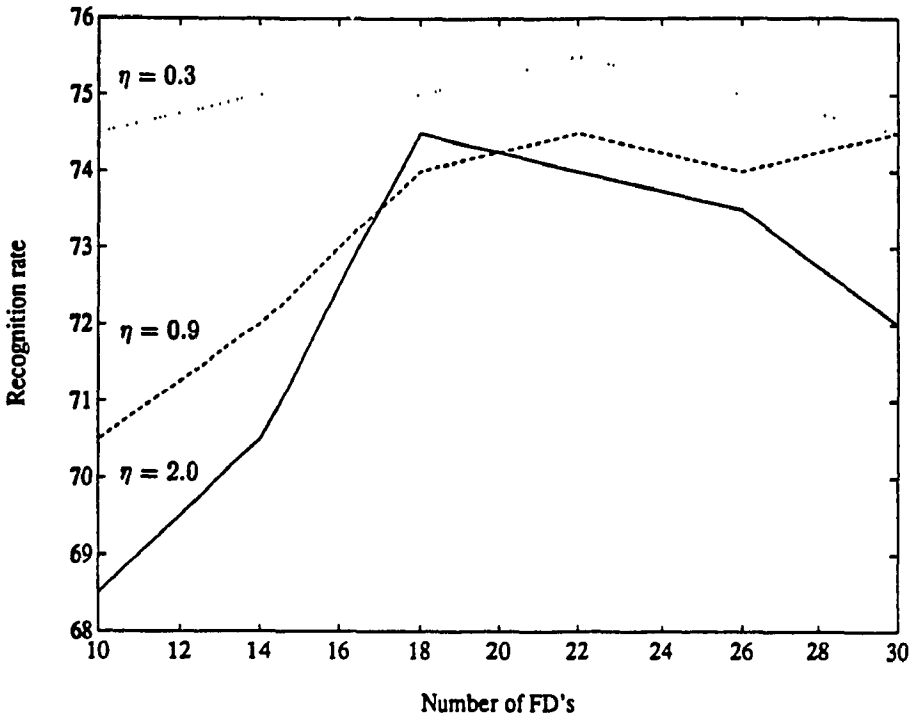


Figure 4.3: Recognition rate vs. number of FD's for Granlund FD's for 200 training and 200 testing samples.

topological features. Further tests using Zahn and Roskies's FD's reveal that the recognition rate with their descriptor is substantially lower than with Granlund's FD and is in the range of 30%. The convergence did not occur for fewer than 50 hidden neural units even after 4000 learning sweeps. The number of iterations for convergence decreased generally as the number of hidden neural units increased. This concludes that the approach of Granlund's FD is superior to that of Zahn and Roskies's FD.

4.4.3 Results from Training Relatively Large Training Samples

In this experiment, 100 patterns for each of numerals 0-9 were used to train the neural networks. Another 1000 patterns were used to test the resulting network. The recognition rates were 84% for the Zahn-Roskies approach with topological features

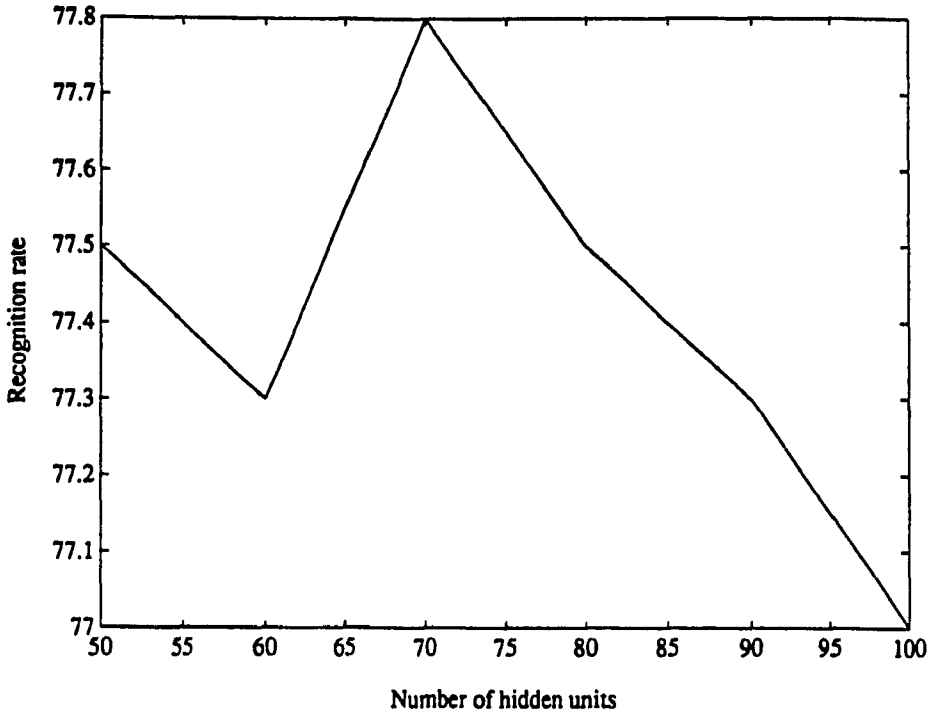


Figure 4.4: Recognition rate vs. number of hidden neurons for Granlund FD's with topological features for 200 training and 200 testing samples.

and 92% for Granlund approach with topological features. The results are shown in Tables 4.2 and 4.3 respectively.

Comparing the results obtained with smaller testing sets and relatively larger sets, it can be seen that a larger training set will result in a higher recognition rate. However, further tests show that when the training set gets larger than 2000 patterns in total, it is not easier to keep the total training error below 0.4. This means that the network may not be able to classify correctly even some of training samples.

4.5 Results from Modified Backpropagation Model

Results presented in this section were obtained, except specified, from the simulation on the modified backpropagation model [19, 17, 18]. The network configuration is based on the 34 inputs (Fourier descriptors and 4 topological features) and 10 outputs

In \ Out	0	1	2	3	4	5	6	7	8	9	Reject
0	88	0	1	1	0	0	7	1	0	2	0
1	0	98	0	0	0	1	0	1	0	0	0
2	0	1	71	0	6	7	0	2	5	7	0
3	0	0	1	85	2	5	0	2	2	3	0
4	1	0	2	2	83	4	1	2	0	5	0
5	0	0	8	0	0	82	0	0	3	7	0
6	1	0	1	0	2	0	85	1	8	2	0
7	0	1	0	9	4	2	0	78	0	6	0
8	0	2	10	0	1	1	1	0	84	1	0
9	0	0	2	5	4	6	1	4	5	73	0

Overall results
The number of numerals correctly classified = 827 (82.7%).
The number of numerals misclassified = 173 (17.3%).
The number of rejected numerals = 0 (0.00%).

Table 4.2: Confusion table for Zahn-Roskies' approach with topological features

In \ Out	0	1	2	3	4	5	6	7	8	9	Reject
0	96	0	0	2	0	0	0	1	1	0	0
1	0	100	0	0	0	0	0	0	0	0	0
2	2	0	83	0	4	0	2	5	2	2	0
3	0	0	2	92	0	2	1	1	1	1	0
4	0	0	0	1	91	1	1	2	2	2	0
5	0	0	2	0	1	91	0	0	1	5	0
6	1	0	1	0	0	0	94	0	2	2	0
7	0	1	2	0	0	1	1	94	0	1	0
8	2	1	1	1	0	0	1	0	92	2	0
9	4	0	0	0	2	2	0	1	0	91	0

Overall results The number of numerals correctly classified = 924 (92.4%).
The number of numerals misclassified = 76 (7.6%).
The number of rejected numerals = 0 (0.00%).

Table 4.3: Confusion table for Granlund's approach with topological features.

In \ Out	0	1	2	3	4	5	6	7	8	9	Reject
0	189	0	0	1	1	0	1	2	3	3	0
1	0	199	0	0	0	0	0	1	0	0	0
2	0	1	189	0	2	2	2	1	1	2	0
3	1	0	3	190	0	3	3	0	0	0	0
4	2	0	2	0	183	0	1	0	1	11	0
5	1	0	1	2	1	189	3	0	0	3	0
6	1	0	1	2	0	2	189	3	2	0	0
7	0	0	3	2	3	2	1	189	0	0	0
8	1	0	2	1	0	1	0	0	195	0	0
9	0	0	2	0	6	2	2	0	3	185	0

Overall results
The number of numerals correctly classified = 1897 (94.85%).
The number of numerals misclassified = 103 (5.15%).
The number of rejected numerals = 0 (0.00%).

Table 4.4: Confusion table for Granlund FD's without rejection.

and a variable number of hidden neurons. The network was trained on 4000 patterns that is for 400 patterns for each class of numerals and a total of 2000 patterns were tested.

4.5.1 False Solution

As mentioned in 3.4.4 section, the false solution problem occurred when a training set got larger. Figure 4.5 clearly shows that the old training algorithm gets stuck in the "local minimum". The total error for the whole training set never goes to zero. In fact, the output values are totally different from the target ones, and 7 of the 4000 numerals were misclassified. Keeping the same configuration as used in Figure 4.6, while using modified BP algorithm instead of the original one, the total error for all training set could be reduced to less than 0.4. For comparison purpose, the total error presented in Figure 4.6 was calculated the same way as the total error

In \ Out	0	1	2	3	4	5	6	7	8	9	Reject
0	179	0	0	0	0	0	1	0	2	0	18
1	0	195	0	0	0	0	0	0	0	0	5
2	0	0	165	0	1	1	1	0	0	0	32
3	0	0	1	175	0	0	1	0	0	0	23
4	0	0	0	0	158	0	0	0	0	2	40
5	1	0	0	0	0	176	0	0	0	1	22
6	0	0	0	1	0	0	173	0	0	0	26
7	0	0	1	0	1	1	0	171	0	0	26
8	0	0	0	0	0	0	0	0	170	0	27
9	0	0	1	0	1	0	0	0	0	165	33

Overall results
The number of numerals correctly classified = 1728 (86.40%).
The number of numerals misclassified = 20 (1.00%).
The number of rejected numerals = 252 (12.60%).

Table 4.5: Confusion table for Granlund FD's with rejection.

In \ Out	0	1	2	3	4	5	6	7	8	9	Reject
0	180	0	2	5	0	0	4	2	3	4	0
1	0	199	1	0	0	0	0	0	0	0	0
2	2	0	158	13	5	7	1	4	6	4	0
3	0	0	11	184	1	1	0	2	1	0	0
4	1	0	2	0	187	5	0	0	2	3	0
5	2	2	12	1	0	172	0	4	3	4	0
6	2	0	1	0	3	0	187	0	7	0	0
7	0	1	4	5	3	3	0	181	0	3	0
8	0	0	5	1	1	0	10	0	182	1	0
9	0	0	5	0	8	5	0	6	2	174	0

Overall results
The number of numerals correctly classified = 1804 (90.20%).
The number of numerals misclassified = 196 (9.80%).
The number of rejected numerals = 0 (0.00%).

Table 4.6: Confusion table for Zahn - Roskies FD's without rejection.

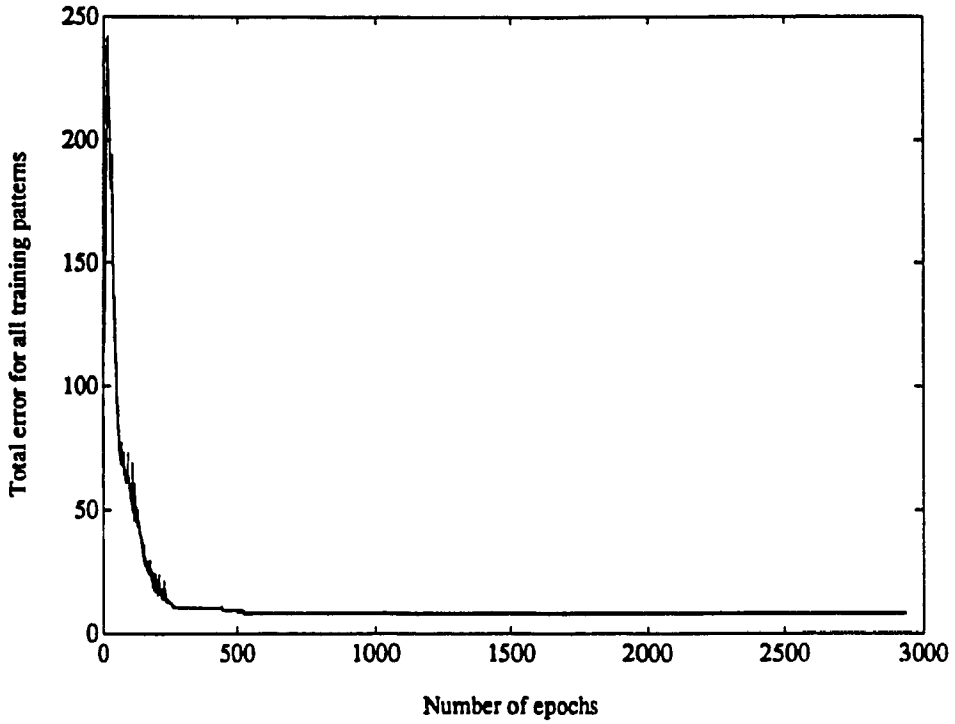


Figure 4.5: The false solution for original learning algorithm.

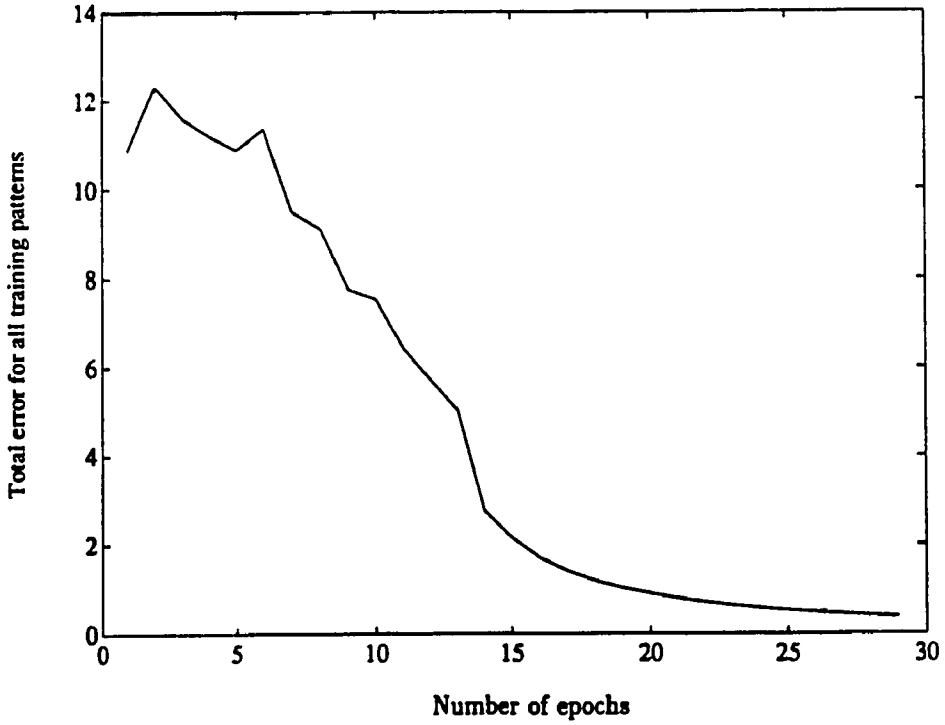


Figure 4.6: Improved convergence ability obtained with the modified learning algorithm.

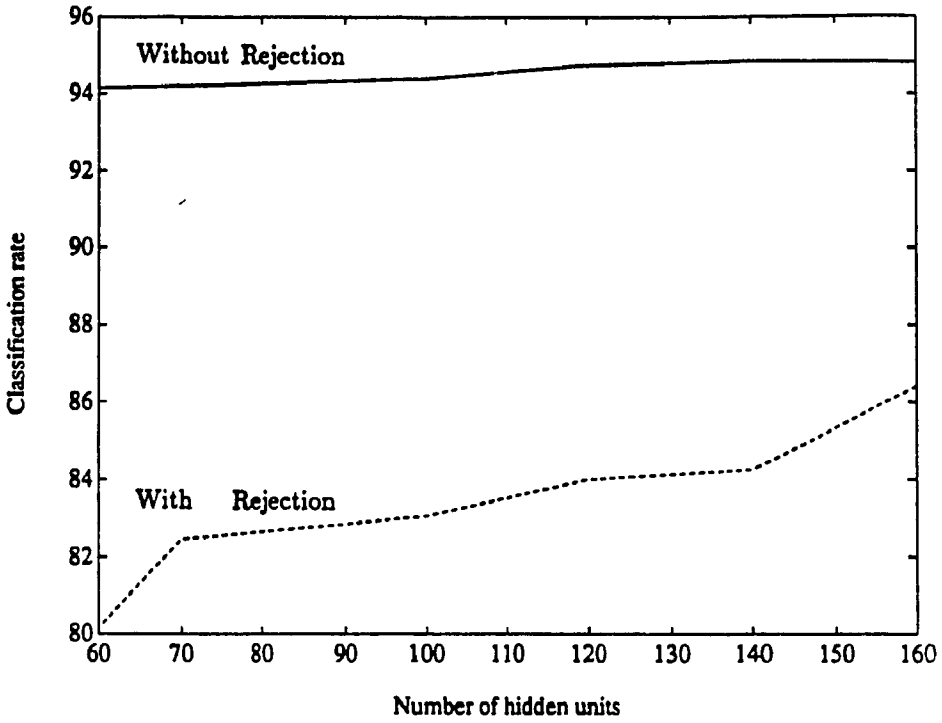


Figure 4.7: Improved classification result from large training set obtained with the modified learning algorithm.

in Figure 4.5.

4.5.2 Classification Results

The recognition results are obtained from the neural network classifier with 160 hidden units and are presented in following tables. Tables 4.5 and 4.4 present results using Granlund's FD's with and without rejection. Table 4.6 is the result of using Zahn-Roskies's descriptors without rejection. Comparing these tables with the previous tables, the classification rates are better than those with the smaller training set.

Figure 4.7 shows the recognition rate as a function of the number of hidden units. It can be seen that the classification rate is higher when the number of hidden units is increased. In the case with rejection, the number of hidden units has more effect on the classification rate than in the case without rejection.

In our experiments we obtained better recognition rates using Granlund Fourier descriptors than using Zahn and Roskie's FD's.

Chapter 5

Conclusions

The basic objectives of this work were to develop and validate a pattern recognition system for handwritten numerals. This system has been implemented by using the Fourier descriptors together with the topological features of the numerals as features and the neural network as the pattern classifier.

Simulation results have shown that the use of FD's as features for numeral recognition was a reliable choice. The results also proved once again that the performance of the Granlund's FD's is better than that of the Zahn and Roskies's FD's [12]. In addition, the use of the topological features of the numerals reduced the confusion rate.

The modified learning algorithm for the neural network is not only two times faster than the original learning model in [33], but it also eliminates the false solution problem which exists in the original learning algorithm during the training stage.

With FD's topological feature vectors and the modified learning model to the neural network, 95% correct recognition rate has been achieved on the testing set of 2000 samples based on 4000 training patterns. A 100% classification rate was achieved on the training data set which is very difficult to achieve with the conventional classifier.

Future Research Directions

Although the use of the backpropagation model for pattern recognition is very successful, there still are a number of unsolved problems related to this model.

- There is no guarantee that the network can be trained in a finite amount of time.
- It is very difficult to predict the optimal number of hidden neurons required for a reliable classification rate.

Bibliography

[1] G. K. Atkin, J. E. Bowcock, and N. M. Queen. Solution of a distributed deterministic parallel network using simulated annealing. *Pattern Recognition*, 22(4):461–466, 1989.

[2] A. E. Bryson and Y. C. Ho. *Applied Optimal Control.* Blaisdell, New York, 1969.

[3] D. J. Burr. Experiments on neural net recognition of spoken and written text. *IEEE Transaction on Acoustics, Speech, and Signal Processing*, 36(7):1162–1168, July 1988.

[4] G. W. Cottrell, P. Munro, and D. Zipser. Image compression by backpropagation: An example of extensional programming. ICS Report 8702, University of California, San Diego, 1987.

[5] Y. Le Cun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard, and W. Hubbard. Handwritten digit recognition : Applications of neural network chips and automatic learning. *IEEE Communications Magazine*, pages 41–46, November 1989.

[6] W. Dai, A. Krzyzak, and C. Y. Suen. On the recognition of handwritten characters using neural network. *Proc. Vision Interface' 90*, pages 84-93, 1990.

[7] R. O. Duda and J. H. Munson. Graphical data processing research study and experimental investigation. *AD657670*, pages 4–8, June 1967.

[8] H. Freeman. Boundary encodig and processing. In Lipkin and Rosenfeld, editors, *Picture Processing and Psychopictorics*, pages 241–306. Academic Press, New York, 1970.

[9] H. Freeman. Computer processing of line-drawings. *Comput. Surv.*, 6:57–97, 1974.

[10] K. Fukushima. A neural network for visual pattern recongition. *IEEE Computer*, pages 65–75, March 1988.

[11] G. H. Granlund. Fourier preprocessing for hand print character recognition. *IEEE Transactions on Computers*, pages 195–201, Febuary 1972.

[12] T. A. Grogan and O. R. Mitchell. Shape recognition and description: A comparative study. Technical Report TR-EE 83-22, School of Electrical Engineering, Purdue University, West Lafayette, Indiana, 1983.

[13] D. Hebb. *Organization of Behavior*. Science Editions, New York, 1961.

[14] W. C. Hoffman. The lie transformation group approach to visual neuropsychology. In E. L. J. Leeuwenberg and H. F. J. M. Buffart, editors, *Formal Theories of Visual Perception*. Wiley, N.Y., 1978.

[15] W. Y. Huang and R. P. Lippmann. Neural net and traditional classifiers. *Conference on Neural Information Processing Systems*, November 1987.

[16] B. Irie and S. Miyake. Capabilities of three-layered perceptrons. *Proc. 1988 IEEE International Conference on Neural Networks*, pages I-641-648, New York, 1988. IEEE Press.

[17] A. Krzyzak, W. Dai, and C. Y. Suen. Classification of large set of handwritten characters using modified back propagation model. *Proc. of the International Joint Conference on Neural Networks*, pages III 225-232, 1990.

[18] A. Krzyzak, W. Dai, and C. Y. Suen. On the recognition of handwritten characters using neural network. In R. Plamondon and H. D. Cheng, editors, *Pattern Recognition Architectures. Algorithms and Applications.* World Scientific Publishing Co., 1990.

[19] A. Krzyzak, W. Dai, and C. Y. Suen. Unconstrained handwritten character classification using modified backpropagation model. *Proceedings of the International Workshop on Frontiers in Hnadwriting Recognition*, pages 155-165, 1990.

[20] A. Krzyzak, T. Kasvand, and C. Y. Suen. *Computer Vision and Shape Recognition.* World Scientific Publishers, Singapore, 1989.

[21] D. G. Lebedev and D. S. Levedev. Quantizing the images by separation and quantization of contours. *Eng. Cybernet*, 1:77-81, 1965.

[22] Richard P. Lippmann. An introduction to computing with neural nets. *IEEE ASSP Magazine*, pages 4-22, April 1987.

[23] T. A. Mai. A knowledge based approach to recognize unconstrained handwritten numerals. Master's thesis, Concordia University, 1989.

[24] S. J. Mason and J. K. Clemens. Character recognition in an experimental reading machine for the blind. In P. A. Kolers and M. Eden, editors, *Recognizing Patterns: Studies in Living and Automatic Systems*, pages 156–167. MIT Press, Cambridge, Mass., 1968.

[25] M. Minsky and S. A. Papert. *Perceptrons.* MIT Press, Cambridge, MA, expanded edition, 1988.

[26] N. J. Nilsson. *The Mathematical Foundations of Learning Machines.* Morgan Kaufmann, N. Y., 1990.

[27] D. B. Parker. Learning-logic. Technical Report TR-47, Center for Computational Research in Economics and Management Science, MIT, April 1985.

[28] T. Pavlidis and F. Ali. Computer recognition of handwritten numerals by polygonal approximations. *IEEE Trans. Syst. Man Cybernet.* SMC-6:610–614, 1975.

[29] E. Persoon and K. S. Fu. Shape discrimination using Fourier descriptors. *IEEE Transactions on Systems, Man Cybernetics*, SMC-7(3):170–179, March 1977.

[30] W. Pitts and W. S. McCulloch. How we know universals: The perception of auditory and visual forms. *Bulletin of Mathematical Biophysics*, 9:127–147, 1947.

[31] F. Rosenblatt. *Principles of Neurodynamics.* Spartan Books, New York, 1962.

[32] D. E. Rumelhart. Learning internal representations by error propagation. ICS Report 8506, Institute for Cognitive Science, University of California, San Diego, September 1985.

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In David E. Rumelhart, James L. McClelland, and

PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol 1: Foundations*, pages 318–362. MIT Press, Cambridge, MA, 1986.

[34] T. J. Sejnowski and C. R. Rosenberg. Parallel networks that learn to pronouce English text. *Complex Systems*, 1:145–168, 1987.

[35] M. Shridhar and A. Badreldin. High accuracy character recognition algorithm using Fourier and topological descriptors. *Pattern Recognition*, 17(5):515–524, 1984.

[36] M. Shridhar and A. Badreldin. Recognition of isolated and simply connected handwritten numerals. *Pattern Recognition*, 19(1):1–12, 1986.

[37] C. Y. Suen, M. Berthod, and S. Mori. Automatic recognition of handprinted characters–the state of art. *Proceedings of the IEEE*, 68(4):469–487, April 1980.

[38] Sun Microsystems. *SunView Programmer's Guide*, March 1989.

[39] M. Symons. A new self-organising pattern recognition system. In *Proc. IEE Conf. on Pattern Recognition*, pages 11–20, London, Eng., July 1968. IEEE.

[40] W. E. Weideman, M. T. Manry, and H. C. Yau. A comparison of a nearest neighbor classifier and a neural network for numeric handprint character recognition. In *Proc. IEEE International conference on Neural Networks*, pages I–117–120, 1989.

[41] P. J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, November 1974.

[42] P. J. Werbos. Backpropagation: Past and future. In *Proc. 1989 IEEE International Conference on Neural Networks*, pages I 343–353, 1989.

[43] S. Yokoi, J. Toriwaki, and T. Fukumura. An analysis of topological properties of digitized binary pictures using local features. *Computer Graphics and Image Processing*, 4:63–73, 1975.

[44] C. T. Zahn and R. S. Roskies. Fourier descriptors for plane closed curves. *IEEE Trans. Computers*, C-21:269–281, March 1972.

Appendix A

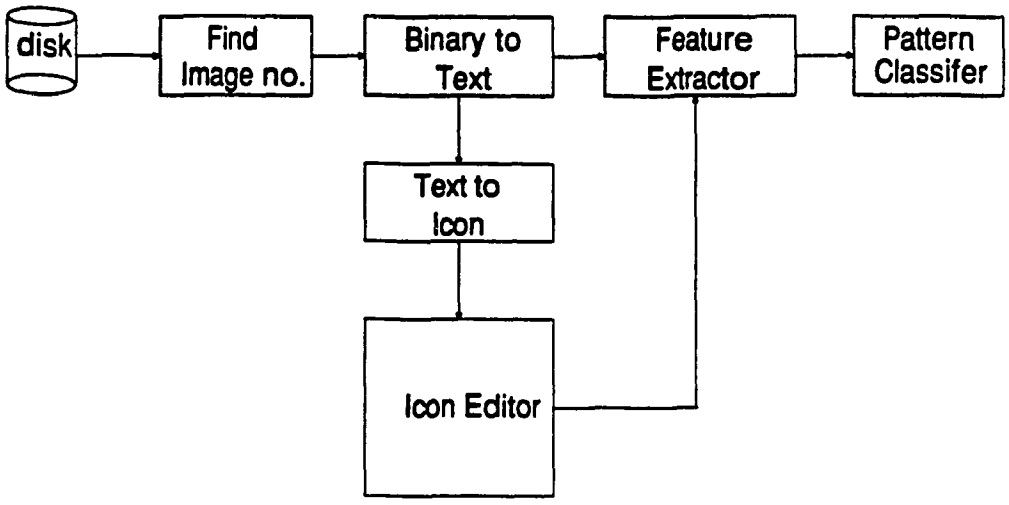
Handwritten Character Recognition Verification System

As part of this work, a handwritten character recognition verification system (CRVS) has been developed. The purpose of this system is to provide a tool to facilitate system developer in testing his/her recognition system. The CRVS provides the following utilities.

- Test the data from the image data base.
- Interactively write your own test handwritten image on the screen.
- Interactively change the image data.

A diagram of the basic system is shown in Figure A.1.

The system was developed on the SUN386i workstation. The image data base is stored in system with different classes of numerals stored according to different files. The file is organized in the sequence of image pattern. Individual image data is stored in two bytes following the image pixels information. The first two bytes are used to store the dimension of the image pattern(number of rows followed by the number of column). Each pixel will take one bit of disk memory(binary file format).



Disk A system disk which used for storing the image data.

Find Image No. A procedure for searching the pattern which requires the user to enter the image no. which they want to test the system.

Binary to Text A procedure for translating the binary file format text file format.

Text to Icon A procedure for translating the test file format to icon file format for displaying the image on the icon editor.

Feature Extractor Two kinds of feature extractor are used in this work.

Pattern Classifier A neural network will be used as a pattern classifier.

Figure A.1: Block diagram of an interactive handwritten recognition verification system

So the image stored in my system is almost 8 times smaller than the original image when one byte in used to store one pixel(text file format). It saves a lot of disk space(almost 1/8 disk space).

For image processing, a text file format is preferred. There is a routine which translates the binary file format to the text file format. There is also a file format translation from the text file format to iconedit file [38] format for displaying the image on the icon edit.

The user can test similar patterns by modifying them with the iconedit to see whether the pattern is recognizable as a check of the system's robustness. This can be easily accomplished. The user can load the image data to iconedit, then use the mouse to modify the pattern interactively, the features extracted from the newly created image then pass through the pattern classifier to verify the classification result. Using this interactive software, one can easily find the weakness of the features and improve the pattern classification results.