

Machine Intelligence and Pattern Recognition 11

MA16G

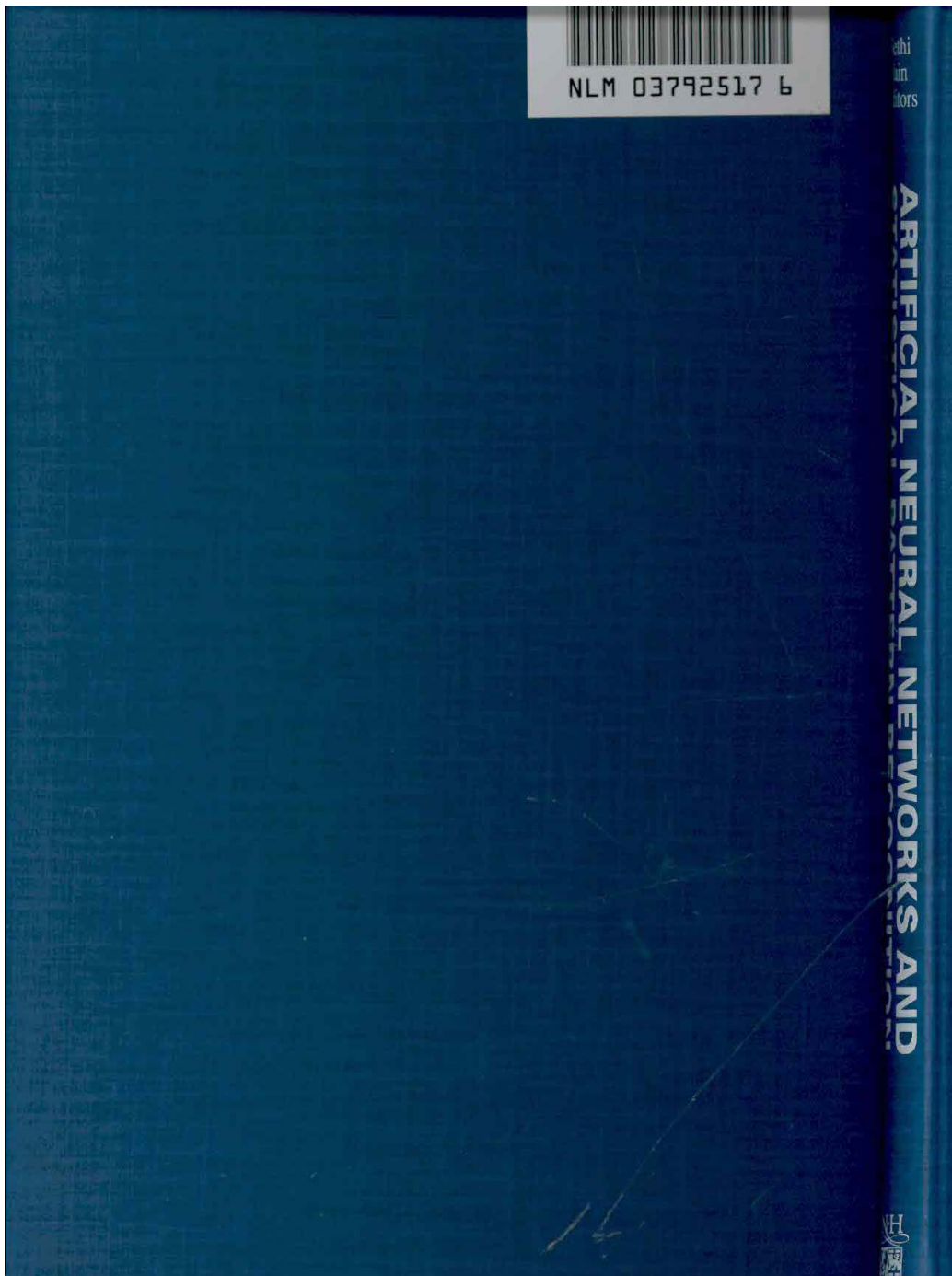
v.11

1991

ARTIFICIAL NEURAL NETWORKS AND STATISTICAL PATTERN RECOGNITION

Edited by
I.K. Sethi
A.K. Jain

North-Holland



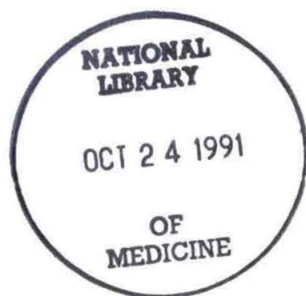
Artificial Neural Networks and Statistical Pattern Recognition

Old and New Connections

Edited by

Ishwar K. SETHI
*Wayne State University
Detroit, Michigan, U.S.A.*

Anil K. JAIN
*Michigan State University
East Lansing, Michigan, U.S.A.*



NORTH-HOLLAND
AMSTERDAM • LONDON • NEW YORK • TOKYO

ELSEVIER SCIENCE PUBLISHERS B.V.
Sara Burgerhartstraat 25
P.O. Box 211, 1000 AE Amsterdam, The Netherlands

Distributors for the United States and Canada:
ELSEVIER SCIENCE PUBLISHING COMPANY, INC.
655 Avenue of the Americas
New York, N.Y. 10010, U.S.A.

Library of Congress Cataloging-in-Publication Data

Artificial neural networks and statistical pattern recognition : old
and new connections / edited by Ishwar K. Sethi, Anil K. Jain.
p. cm. -- (Machine intelligence and pattern recognition ; v.
11)
Includes bibliographical references and index.
ISBN 0-444-88740-7. -- ISBN 0-444-88741-5 (pbk.)
1. Pattern recognition systems. 2. Neural networks (Computer
science) I. Sethi, Ishwar K., 1948- II. Jain, Anil K., 1948-
III. Series.
TK7882.P3A7 1991
006.3--dc20

91-24621
CIP

ISBN: 0 444 88740 7 (hardbound)
0 444 88741 5 (paperback)

© 1991 ELSEVIER SCIENCE PUBLISHERS B.V. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of the publisher, Elsevier Science Publishers B.V., Permissions Department, P.O. Box 521, 1000 AM Amsterdam, The Netherlands.

Special regulations for readers in the U.S.A. - This publication has been registered with the Copyright Clearance Center Inc. (CCC), Salem, Massachusetts. Information can be obtained from the CCC about conditions under which photocopies of parts of this publication may be made in the U.S.A. All other copyright questions, including photocopying outside of the U.S.A., should be referred to the copyright owner, Elsevier Science Publishers B.V., unless otherwise specified.

No responsibility is assumed by the publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.

pp. 1-10, 11-32: copyright not transferred.

Printed in The Netherlands

TABLE OF CONTENTS

Foreword	v
Preface	ix
 ANN AND SPR RELATIONSHIP	
Evaluation of a Class of Pattern-Recognition Networks L. Kanal	1
Links between Artificial Neural Networks (ANN) and Statistical Pattern Recognition P.J. Werbos	11
Small Sample Size Problems in Designing Artificial Neural Networks S. Raudys and A.K. Jain	33
On Tree Structured Classifiers S.B. Gelfand and E.J. Delp	51
Decision Tree Performance Enhancement Using an Artificial Neural Network Implementation I.K. Sethi	71
 APPLICATIONS	
Bayesian and Neural Network Pattern Recognition: A Theoretical Connection and Empirical Results with Handwritten Characters D.-S. Lee, S.N. Srihari and R. Gaborski	89
Shape and Texture Recognition by a Neural Network A. Khotanzad and J.-H. Lu	109
Neural Networks for Textured Image Processing J. Ghosh and A.C. Bovik	133
Markov Random Fields and Neural Networks with Applications to Early Vision Problems A. Rangarajan, R. Chellappa and B.S. Manjunath	155
Connectionist Models and their Application to Automatic Speech Recognition Y. Bengio and R. De Mori	175

IMPLEMENTATION ASPECTS

Dynamic Associative Memories M.H. Hassoun	195
Optical Associative Memories B.V.K. Vijaya Kumar and P.K. Wong	219
Artificial Neural Nets in MOS Silicon F.M.A. Salam, M.-R. Choi and Y. Wang	243
Author Index	271

Bayesian and neural network pattern recognition : a theoretical connection and empirical results with handwritten characters

Dar-Shyang Lee^a, Sargur N. Srihari^a and Roger Gaborski^b

^aDepartment of Computer Science, State University of New York at Buffalo, New York 14260

^bEastman Kodak Company, Rochester, New York 14650

Abstract

Statistical Pattern Recognition and Artificial Neural Networks provide alternative methodologies to the classification of patterns represented as feature vectors. This paper provides a theoretical relationship and an empirical comparison between the Bayes decision rule and the backpropagation model. It is shown that backpropagation performs least mean square approximation to the Bayes discriminant function. While a three-layer backpropagation network (one hidden layer) with a sufficient number of hidden units is known to possess universal mapping ability, gradient-descent based backpropagation learning does not guarantee finding the minimum probability of error solution. Experimental results with handwritten character recognition (digits and letters extracted from handwritten addresses) are presented. The experiments are with two different representations of characters : binary pixel arrays and structural features represented as binary vectors. With pixel arrays, the backpropagation model performs better than the first-order Bayes discriminant that assumes statistical independence between pixels. With structural features, the first-order Bayes and backpropagation have similar performance. However, training of a backpropagation network is much more involved. Inherent difficulties with both classifiers are discussed.

1 Introduction

Pattern recognition has been an active field of research for over thirty years inspiring many theoretical and experimental results. Well-developed statistical approaches to pattern recognition exist [Fuk72, DH73, TG74]. Recently much attention has been diverted to pattern recognition using the artificial neural network approach, with many successful experiments reported [Fuk88, Bur87, RMS89, DJG⁺89]. Weiss and Kapouleas [WK89]

have done a thorough empirical comparison of pattern recognition, neural network, and machine learning classification methods.

This paper focuses on the statistically optimal Bayes classifier and a popular neural network architecture, the backpropagation network. In particular, our interest is in experimental evaluation for the case of binary feature vectors arising from the problem of handwritten character recognition. In analyzing these two classifiers from both theoretical and empirical aspects, we hope to gain some insight into their relationships.

The organization of the paper is as follows. Section 2 introduces the Bayes classifier and its approximation for the case of binary feature vectors. Section 3 introduces artificial neural networks and the backpropagation concept. Section 4 provides a theoretical analysis of the generalized delta rule used in a backpropagation network and relates it to the error rate achieved by the Bayes classifier. It is shown that backpropagation performs least mean square error approximation to the Bayes discriminant function. Section 5 describes experiments with both classifiers trained and tested on a database of more than 20,000 handwritten characters. Comparisons of these classifiers and practical problems in training them are discussed in section 6.

2 Bayes Classifier

The Bayes decision rule is a well-studied statistical classification method which is defined to be optimal with known *a priori* and *class-conditional* probabilities [DH73]. It minimizes conditional risks for any loss function.

Let the input $\mathbf{x}$ be a vector of d random variables, and let $C_1, C_2, \dots, C_n$ be the n classes to which all possible input vectors may belong. Bayes rule states that

$$P(C_j|\mathbf{x}) = \frac{P(\mathbf{x}|C_j)P(C_j)}{P(\mathbf{x})} \quad (1)$$

where

$$P(\mathbf{x}) = \sum_{i=1}^n P(\mathbf{x}|C_i)P(C_i). \quad (2)$$

By assigning $\mathbf{x}$ to the class with maximum *a posteriori* probability, $P(C_j|\mathbf{x})$, the probability of error is minimized. Defining the discriminant function $g_j(\mathbf{x})$ for each class C_j as the *a posteriori* probability for class j

$$g_j(\mathbf{x}) = P(C_j|\mathbf{x}) \quad (3)$$

the minimum probability of error can be achieved by selecting the class whose discriminant function has the highest value, or

$$g_j(\mathbf{x}) = \frac{P(\mathbf{x}|C_j)P(C_j)}{P(\mathbf{x})}. \quad (4)$$

Since $P(\mathbf{x})$ is independent of the classes, it can be eliminated without affecting the decisions. We can further reduce this equation by applying a monotonically increasing function, such as log, on all discriminant functions. The result is

$$g_j(\mathbf{x}) = \log P(\mathbf{x}|C_j) + \log P(C_j). \quad (5)$$

In order to use this discriminant function, we need to know the probability density function of $\mathbf{x}$ with respect to each class C_j , and the *prior* probability of each class C_j . In practical pattern recognition problems, complete knowledge of either is rare. More often we are provided with a finite number of samples, and are forced to estimate the *a priori* and *class-conditional* probabilities based on the training set.

2.1 Binary Case

We will now consider the case where the components of $\mathbf{x}$ are binary. In general, the *a priori* probability, $P(C_j)$ can be estimated quite accurately if the training set is large. However, approximation of class-conditional probabilities is a challenging task. Since there are 2^d possible d -dimensional binary vectors, 2^d probabilities need to be calculated for each class. The large dimensionality often encountered in pattern recognition prohibits us from making such approximations.

2.1.1 First-Order Approximation

If the components of $\mathbf{x}$ are all statistically independent of each other, the class-conditional probability can be expressed as the product of conditional probabilities of each component, as

$$P(\mathbf{x}|C_j) = \prod_{i=1}^d P(x_i|C_j), \quad (6)$$

where x_i is the i th component of $\mathbf{x}$. Therefore, we need to estimate d probabilities for each class.

If we let

$$P_{ij} = P(x_i = 1|C_j) \quad (7)$$

and

$$1 - P_{ij} = P(x_i = 0|C_j) \quad (8)$$

then

$$P(\mathbf{x}|C_j) = \prod_{i=1}^d P_{ij}^{x_i} (1 - P_{ij})^{1-x_i} \quad (9)$$

Substituting Equation 9 into Equation 5, it is simple to derive the following.

$$g_j(\mathbf{x}) = \sum_{i=1}^d x_i \log \frac{P_{ij}}{1 - P_{ij}} + \sum_{i=1}^d \log(1 - P_{ij}) + \log P(C_j). \quad (10)$$

Since this equation is linear in x_i , we can rewrite Equation 10 as

$$g_j(\mathbf{x}) = \sum_{i=1}^d x_i w_{ij} + b_j \quad (11)$$

where

$$w_{ij} = \log \frac{P_{ij}}{1 - P_{ij}} \quad (12)$$

and

$$b_j = \sum_{i=1}^d \log(1 - P_{ij}) + \log P(C_j). \quad (13)$$

Therefore, the discriminant function for each class can be efficiently characterized by a bias factor and a d -dimensional weight vector.

2.1.2 Estimating Parameters

Since the input vector is binary, and its components are independent under our assumption, the *class-conditional* probabilities can be estimated by building a $n \times d$ histogram. Then P_{ij} can be approximated by the sample mean k_{ij}/s_j , where s_j is the number of samples for class j , and k_{ij} is the number of occurrences of 1's in the i th component among all s_j samples. One common problem encountered using this approach is to determine a suitable value for P_{ij} when $k_{ij} = 0$, $s_j = 0$ or $k_{ij} = s_j$. Any one of those situations leads to representing either infinity or negative infinity. A useful technique that avoids this problem is to estimate P_{ij} by $(k_{ij} + 1)/(s_j + 2)$. In fact, it has been shown this is the best estimation in case of a mean square error loss function. Good results have also been observed by assigning P_{ij} the value of $1/3s_j$ when P_{ij} is 0.

With a given set of feature vectors, with complete knowledge of *a priori* and *conditional* probabilities, the Bayes decision rule will provide the minimum error rate. The discriminant function in Equation 10 would be appropriate if the components in the

feature vectors are indeed statistically independent and all conditional probabilities are known. In some pattern recognition problems, neither of those assumption can be justified. Consequently, working under the above assumptions, we run the risk of obtaining suboptimal performance.

2.1.3 Higher-Order Approximation

There are techniques which allow us to compromise between working under the assumption of strict statistical independence among components and having to estimate 2^d probabilities. One technique is to use a generalized decision function which has the form

$$\sum_{i=0}^m c_i \phi_i(\mathbf{x}), \quad (14)$$

where $\phi_i(\mathbf{x})$ are orthogonal functions and c_i are their coefficients.

An example of a generalized decision function is one based on Bahadur-Lazarsfeld polynomials [DH73]. The *class-conditional* probability can be calculated by

$$P(\mathbf{x}|C_j) = P_I(\mathbf{x}|C_j) \sum_{i=0}^{2^d-1} a_{ji} \psi_{ji}(\mathbf{x}), \quad (15)$$

where

$$a_{ji} = \mathcal{E}[\psi_{ji}(\mathbf{x})] \quad (16)$$

is the expected value of $\psi_{ji}(\mathbf{x})$, and

$$\psi_{ji}(\mathbf{x}) = \begin{cases} 1 & i = 0 \\ y_1 & i = 1 \\ \vdots & \\ y_d & i = d \\ y_1 y_2 & i = d + 1 \\ \vdots & \\ y_{d-1} y_d & i = d + 1 + d(d-1)/2 \\ y_1 y_2 y_3 & i = d + 2 + d(d-1)/2 \\ \vdots & \\ y_1 y_2 \cdots y_d & i = 2^d - 1 \end{cases} \quad (17)$$

and y_i is the normalized variable,

$$y_i = \frac{x_i - P_{ij}}{\sqrt{P_{ij}(1 - P_{ij})}}.$$

The weighting function $P_I(\mathbf{x})$ is precisely the conditional probability under the independence assumption,

$$P_I(\mathbf{x}) = \prod_{i=1}^d p_i^{x_i} (1 - p_i)^{1-x_i}. \quad (18)$$

From the definition of ψ , it is clear that $a_{j0} = 1$ and $a_{j1} = a_{j2} \cdots = a_{jd} = 0$. Therefore, the discriminant function in Equation 10 is simply the first order approximation by Bahadur-Lazarsfeld expansion. The probability density function can be estimated more accurately by going into higher order terms. For our database, empirical results discourage us from such expansion as little is gained at a cost of many more coefficients. Nevertheless, in general this trade off between accuracy and efficiency will depend on the nature of the problem at hand.

3 Artificial Neural Networks and Back Propagation

Artificial neural networks have been used in pattern recognition problems as early as 1957 when Rosenblatt introduced the perceptron [Ros57]. It is well-known that they are incapable of discriminating between non-linearly separable classes [MP69].

One major contribution in the revival of neural networks is the generalized delta rule of Rumelhart et.al. [MRG86], which was discovered independently by Werbos [Wer74] and Parker [Par85]. This rule provides a learning scheme for a multi-layer perceptron-like network with non-linear activation functions, thus allowing generation of any type of decision surface. Of course, other neural network architectures have been attempted for pattern recognition and other applications. However, we will not give a comprehensive overview of all neural network models here. A good introduction to various models can be found in [MRG86, Lip87].

Although the generalized delta rule can be applied to any network, we will concentrate on a particularly popular *layered feedforward* network, the backward error propagation (BP) model. The network consists of three types of layers, each of which is composed of various number of units. Units in adjacent layers are connected through links whose associated weights determine the contribution of units on one end to the overall activation of units on the other end. There are generally three types of layers. Units in the *input* layer bear much resemblance to the sensory units in a classical perceptron. Each of them is connected to a component in the input vector. The *output* units, analogous to the response units in a perceptron, represent different classes of patterns. Arbitrarily many *hidden* layers may be used depending on the desired complexity. Each unit in the hidden layer is connected to every unit in the layer immediately above and below.

Data flow in a BP can be either in forward or backward direction. In the forward direction, input data is transmitted from input to output layer, layer by layer, using the propagation rule.

$$o_j = f(\sum_i w_{ij} o_i + \theta_j) \quad (19)$$

where the output of unit j , o_j , is determined by the weighted sum of outputs of all units in the preceding layer, $\sum_i w_{ij} o_i$, and a bias term, θ_j , applied on an activation function f . An example of f suggested in [MRG86] is the sigmoid function,

$$f = \frac{1}{1 + e^{-x}}. \quad (20)$$

However, any choice of continuous, non-decreasing, differential function will be appropriate. The decision of the system is measured at the output layer. For the purpose of pattern recognition, it is common to assign an input pattern to the class whose corresponding output unit has the highest value among all output units.

The decision surface for a BP is formed during training phase by a series of weight adjustments. These weight adjustments are determined by the error signals transmitted in the backward direction. As each labeled pattern is fed in from the input layer and propagated to the output layer, values of the output units are compared to the desired output responses. The amount of error attributed by each unit, δ_j , is calculated, layer by layer, from output to input layer. For output units,

$$\delta_j = (t_j - o_j) o_j (1 - o_j), \quad (21)$$

where t_j is the ideal output response. For hidden units,

$$\delta_j = o_j (1 - o_j) \sum_k w_{jk} \delta_k, \quad (22)$$

where δ_k is the error contributed by each unit in the layer immediately above. After δ is calculated for each unit, every weight is adjusted by

$$\Delta w_{ij}(t) = \eta \delta_j o_i + \alpha \Delta w_{ij}(t-1), \quad (23)$$

where $\Delta w_{ij}(t)$ is the weight change at iteration t ; η , the learning rate and, α , the momentum, are scaling factors. The choice of these two parameters has great effect on the convergence rate of the system. In experiments described in Section 5, we used a learning rate of 0.75 and a momentum of 0.5. It has also been observed that BP performance is influenced by the number of hidden units available [KIH88, GWG89]. For digit recognition we chose 15 hidden units, and for alphabet recognition we used 70 hidden units.

4 Relationship

We will first derive a theoretical relationship between the Bayes decision rule and backward error propagation.

The Bayes decision rule performance is optimal for a given set of features in the sense that it minimizes the probability of error and the conditional risk. As stated, this requires complete knowledge of the underlying probability density function for each class. However, in practice, finite training samples and high dimensionality compound this simple decision theory. The difficulty in actuating the class-conditional probability often makes the performance less than optimal.

Unlike the empirical approach in the 60's, the resurgence of neural networks has been led off by a series of theoretical analysis on their capabilities. Many important works have demonstrated the universal mapping ability of a backpropagation model under various constraints [HN89, Ara89, IM88, SW89, Shv90].

In order to understand the theory behind backpropagation, we need to study the derivation of generalized delta rule, which originates from minimizing the squared error sum between network and desired output responses over all patterns.

$$E = \sum_{\mathbf{x}} E_{\mathbf{x}} = \sum_{\mathbf{x}} \frac{1}{2} \sum_j (t_{\mathbf{x}j} - o_{\mathbf{x}j})^2 \quad (24)$$

where $t_{\mathbf{x}j}$ and $o_{\mathbf{x}j}$ are, respectively, the desired and actual response of j th output unit to pattern $\mathbf{x}$. To minimize E with respect to each weight w_{ij} , it is necessary to find the root for its partial derivative,

$$\frac{\partial E}{\partial w_{ij}} = \sum_{\mathbf{x}} \frac{\partial E_{\mathbf{x}}}{\partial w_{ij}}. \quad (25)$$

Thus it is sufficient to minimize $E_{\mathbf{x}}$. Gradient descent is the standard technique for solving such problems. An approximation of gradient descent is achieved by making weight changes proportional to $E_{\mathbf{x}}$ after each presentation, i.e.,

$$\Delta_{\mathbf{x}} w_{ij} \propto - \frac{\partial E_{\mathbf{x}}}{\partial w_{ij}},$$

The detailed derivation can be found in [MRG86]. For our purpose, it suffices to state the result:

$$\Delta_{\mathbf{x}} w_{ji} = \eta \delta_{\mathbf{x}j} o_{\mathbf{x}i}, \quad (26)$$

where

$$\delta_{\mathbf{x}j} = (t_{\mathbf{x}j} - o_{\mathbf{x}j}) f'(\sum_i w_{ij} o_{\mathbf{x}i} + \theta_j) \quad (27)$$

for output units and

$$\delta_{\mathbf{x}j} = f'(\sum_i w_{ij}o_{\mathbf{x}i} + \theta_j) \sum_k \delta_{\mathbf{x}k}w_{kj} \quad (28)$$

for hidden units. In Equation 28 and 29, f' denotes the first derivative of f . When f is the sigmoid function in Equation 20, it can be shown that

$$f'(x) = f(x)(1 - f(x)) \quad (29)$$

and we obtain the formula in Equation 21 and 22.

We should point out that backpropagation performs gradient descent in E . However, by adjusting the weights after each pattern is presented, we deviate from true gradient descent. Nevertheless, Rumelhart noted that by selecting sufficiently small learning rate, η , a good approximation of gradient descent can be obtained through sequences of small movements [MRG86].

The statistical significance of minimizing squared error can be shown in the following analysis. First, we rewrite the Bayes decision rule for notational convenience.

$$P(C_j|\mathbf{x}) = \frac{P(\mathbf{x}|C_j)P(C_j)}{\sum_{i=1}^s P(\mathbf{x}|C_{i \neq j})P(C_{i \neq j})} = \frac{P(\mathbf{x}, C_j)}{P(\mathbf{x})}$$

Consider a single output unit $f_j(\mathbf{x}, \mathbf{w})$ and random classification variable t_j , which is 1 if $\mathbf{x} \in C_j$ and 0 otherwise.

$$t_j(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in C_j \\ 0 & \text{if } \mathbf{x} \notin C_j \end{cases} \quad (30)$$

Then the criterion function J_s is defined.

$$\begin{aligned} J_s(\mathbf{w}) &= \sum_{\mathbf{x}} [f_j(\mathbf{x}, \mathbf{w}) - t_j]^2 \\ &= \sum_{\mathbf{x} \in C_j} [f_j(\mathbf{x}, \mathbf{w}) - 1]^2 + \sum_{\mathbf{x} \notin C_j} [f_j(\mathbf{x}, \mathbf{w}) - 0]^2 \\ &= n \left[\frac{n_j}{n} \frac{1}{n_j} \sum_{\mathbf{x} \in C_j} [f_j(\mathbf{x}, \mathbf{w}) - 1]^2 + \frac{n - n_j}{n} \frac{1}{n - n_j} \sum_{\mathbf{x} \notin C_j} f_j(\mathbf{x}, \mathbf{w})^2 \right] \end{aligned}$$

where n is the total number of training samples, and n_j is the number of samples for class C_j .

Applying the law of large numbers and Bayes rule, we obtain the following result.

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{1}{n} J_s(\mathbf{w}) &= J(\mathbf{x}, \mathbf{w}) \\ &= P(C_j) \int [f_j(\mathbf{x}, \mathbf{w}) - 1]^2 P(\mathbf{x}|C_j) d\mathbf{x} + P(C_{i \neq j}) \int f_j(\mathbf{x}, \mathbf{w})^2 P(\mathbf{x}|C_{i \neq j}) d\mathbf{x} \\ &= \int [f_j(\mathbf{x}, \mathbf{w}) - 1]^2 P(\mathbf{x}, C_j) d\mathbf{x} + \int f_j(\mathbf{x}, \mathbf{w})^2 P(\mathbf{x}, C_{i \neq j}) d\mathbf{x} \end{aligned}$$

$$\begin{aligned}
&= \int f_j(\mathbf{x}, \mathbf{w})^2 P(\mathbf{x}) d\mathbf{x} - 2 \int f_j(\mathbf{x}, \mathbf{w}) P(\mathbf{x}, C_j) d\mathbf{x} + \int P(\mathbf{x}, C_j) d\mathbf{x} \\
&= \int f_j(\mathbf{x}, \mathbf{w})^2 P(\mathbf{x}) d\mathbf{x} - 2 \int f_j(\mathbf{x}, \mathbf{w}) P(C_j | \mathbf{x}) P(\mathbf{x}) d\mathbf{x} + \int P(C_j | \mathbf{x}) P(\mathbf{x}) d\mathbf{x} \\
&= \int [f_j(\mathbf{x}, \mathbf{w}) - P(C_j | \mathbf{x})]^2 P(\mathbf{x}) d\mathbf{x} + \int P(C_j | \mathbf{x}) P(C_{i \neq j} | \mathbf{x}) P(\mathbf{x}) d\mathbf{x}
\end{aligned}$$

In the last equation, since the second term on the right hand side is independent of $\mathbf{w}$, whatever minimizes J_s/n also minimizes the first term. Therefore, backpropagation learning seeks least mean square approximation of the posterior density function in weight space. This is an extension of the analysis of two class linear classifiers given in [DH73].

Knowing that the output of a backpropagation network approximates the posterior density function, it is reasonable to ask whether a minimum probability of error solution exists. In [HH90] it has been shown that when the probability density function is Gaussian, a perceptron with a sigmoid transfer function approximates the a posteriori probabilities. We will extend this result to multiclass problems without restricting the form of underlying density functions.

Using the criterion function in our previous analysis, we can minimize the total error by finding the root to the first partial derivative of $J(\mathbf{w}, \mathbf{x})$ with respect to $\mathbf{w}$. Therefore,

$$\frac{\partial J}{\partial \mathbf{w}} = 2P(C_j) \int_{\mathbf{x}} (o_j - 1) \frac{\partial f_j(\mathbf{x}, \mathbf{w})}{\partial \mathbf{w}} P(\mathbf{x} | C_j) d\mathbf{x} + 2(1 - P(C_j)) \int_{\mathbf{x}} o_j \frac{\partial f_j(\mathbf{x}, \mathbf{w})}{\partial \mathbf{w}} P(\mathbf{x} | C_{i \neq j}) d\mathbf{x} = 0 \quad (31)$$

All possible $\mathbf{w}$ which achieves local minimum must satisfy this condition. However, any solution which satisfy this requirement may not be a minimum in E space. In fact, it may well be a local maximum or saddle point. To ensure the local minimum property, we need to exam the sign of the second derivative. The solution is a minimum if and only if

$$\frac{\partial^2 J}{\partial \mathbf{w}^2} > 0$$

Nevertheless, we will proceed to analyze its most obvious solution. This condition can be satisfied when

$$P(C_j)(1 - f_j(\mathbf{x}, \mathbf{w})) \frac{\partial f_j(\mathbf{x}, \mathbf{w})}{\partial \mathbf{w}} P(\mathbf{x} | C_j) = (1 - P(C_j)) f_j(\mathbf{x}, \mathbf{w}) \frac{\partial f_j(\mathbf{x}, \mathbf{w})}{\partial \mathbf{w}} P(\mathbf{x} | C_{i \neq j})$$

for all $\mathbf{x}$. Under the assumption that $\partial f_j(\mathbf{x}, \mathbf{w}) / \partial \mathbf{w}$ is integrable and nonzero everywhere, the above equation can be simplified to

$$(1 - f_j(\mathbf{x}, \mathbf{w})) \frac{P(C_j) P(\mathbf{x} | C_j)}{P(\mathbf{x})} = f_j(\mathbf{x}, \mathbf{w}) \frac{P(C_{i \neq j}) P(\mathbf{x} | C_{i \neq j})}{P(\mathbf{x})}$$

The multiplicative constant $P(\mathbf{x})$, of course, has to be nonzero. By applying Bayes rule (Equation 1), we obtain

$$(1 - f_j(\mathbf{x}, \mathbf{w}))P(C_j|\mathbf{x}) = f_j(\mathbf{x}, \mathbf{w})P(C_{i \neq j}|\mathbf{x})$$

Thus,

$$P(C_j|\mathbf{x}) = f_j(\mathbf{x}, \mathbf{w})P(C_{i \neq j}|\mathbf{x}) + f_j(\mathbf{x}, \mathbf{w})P(C_j|\mathbf{x})$$

Since

$$P(C_j|\mathbf{x}) + P(C_{i \neq j}|\mathbf{x}) = 1$$

the following conclusion is reached :

$$f_j(\mathbf{x}, \mathbf{w}) = P(C_j|\mathbf{x}). \quad (32)$$

Not surprisingly, one solution is the posterior density function. The implicit assumption being made in the above analysis is that f is functionally capable of representing $P(C_j|\mathbf{x})$. A three layer BP model with infinite number of hidden units is necessary and sufficient to approximate any posterior density function to any degree of accuracy [HN89, Ara89, SW89]. The diagnosis of a more realistic feedforward network with bounded fan in can be found in [Shv90]. Without any prior knowledge of forms of the density functions, using a two layer network (no hidden layers) or a linear activation function will severely limit the representation power of our classifier.

From the above analysis, we make the following conclusion. Backpropagation performs least mean square approximation to the Bayes decision rule. A minimum probability of error solution indeed exists if the network is computationally capable of representing the a posteriori probability function exactly. A three layer backpropagation model (1 hidden layer) with sufficient number of hidden units has been shown to possess universal mapping ability and, therefore, can approximate any (a posteriori density) function to any degree of accuracy [HN89, SW89, Ara89].

However, the gradient descent technique used in backpropagation learning rule does not guarantee convergence to this solution. Networks can get trapped in a local minimum. Without an exact representation of the posterior function, this mean-square-error approximation may not give us the optimal classifier.

In this section we have shown that a multilayer feed forward network is functionally equivalent to approximating the a posteriori probability using gradient descent.

5 Experimental Results

Several experiments were conducted on handwritten digits and alphabets using the first order Bayes approximation and backward error propagation. Original images were di-

Table I:

95% confidence interval for digit recognition correct rate with 212 PPI images (manual segmentation)

	low	estimation	high
First-order Bayes (pixel)	91.4	92.2	92.9
Backprop (pixel)	94.9	95.5	96.0
First-order Bayes (feature)	97.5	97.9	98.3
Backprop (feature)	98.1	98.5	98.8

rectly obtained from handwritten addresses on mail samples. The specification of data and size of training and testing sets will be described in each section.

Input data for those pattern recognition algorithms primarily consisted of two formats. The first type of input, which we will refer to as *pixel* input, was binarized, size normalized images. Binarization was achieved by global thresholding, and moment normalization [Cas70] was used to scale all images to size 16x16.

The second type, which we will call *feature* input, was uniform length feature vectors of 636 binary components obtained by various feature extractors described in [LLS90]. Three types of feature extractors were used in constructing the feature vector: structural, contour analysis and feature templates. The structural approach was based on the work of [Pav86]. Several parameters and thresholds were adjusted to adapt the variations in character styles. The contour analysis based method used chaincodes to extract features such as holes and concavities. The third extractor implemented feature templates [BGJH88]. Detected features are represented in form of a fixed lengthed binary vector.

5.1 Digit Recognition

Two different experiments were done on handwritten digit recognition. Data from the first experiment consists of binary images that were scanned by OCR machines at 212 ppi (pixel per inch). These digits were acquired by manually locating the zipcodes on an envelope, and then segmenting them by a program. Binarization and moment normalization were applied on the images to obtain 16x16 bitmaps. The data has been cleaned up to eliminate any badly segmented images. The training set used to estimate parameters (section 2.1.2) contained 10,000 images, and the test set was a disjoint set of 5,000 images. Table I shows the 95% confidence interval for the correct rate [Hig62, CP34].

The images used for the second experiment were scanned at 300 ppi with 256 different gray scale value, and then thresholded to binary and moment normalized. Segmentation of digits were performed by a program. The training set, which was composed of 18,650 images, was cleaned up. Whereas the 2711 images in the novel testing set contains an estimated 3% of improperly segmented images. The performance is shown in Table II. Not surprisingly, the performance is not as good as that in the first experiment. Most of the errors were caused by poor segmentations.

5.2 Alphabet Recognition

The alphabet recognition experiment involves classification of all handprinted upper and lower case alphabets. However, since each individual character is presented without the baseline information, the following classes are combined : C-c, K-k, O-o, P-p, S-s, U-u, V-v, W-w, X-x, Z-z. After preliminary experiments, it was observed that much of the substitution error was caused by badly formed characters in class I-i-l, Y-y. Therefore, those classes are also combined. Thus we have a total of 41 classes.

All images in the training and testing set were scanned at 300ppi with 256 gray scale values and then thresholded and moment normalized to 16x16 binary images. Segmentation was performed on-line manually. The training set consists of 8000 images, unevenly distributed among classes. The test set contained 2865 previously unseen images. Experimental results are summarized in Table III.

Samples of images used in both the digit and alphabet recognition problems are given in the appendix. These images were extracted from addresses on envelopes.

6 Discussion

From the analysis in section 4, we have seen for a specific target function, outputs of a backpropagation network approximate the a posteriori probabilities. Therefore, by assigning the input pattern to the class whose corresponding output unit has the highest value, we effectively maximize the (estimated) posterior probability, and minimize the probability of error.

In practice, however, the comparison between Bayesian classifier and backpropagation is complicated by various factors. In most real-world problems, high-dimensionality and finite training samples inhibit us from making accurate calculation of probability density functions. Working reluctantly under an assumption of statistical independence or with low order approximations, we expectedly obtain suboptimal performance.

While backpropagation provides an automated solution to such problems, it also has its pitfalls. Using a gradient descent technique in a most likely non-parabolic error space, it is susceptible to local minima. In fact, Hecht-Nielsen [HN89] has shown the existence of

Table II:

95% confidence interval for digit recognition correct rate with 300 PPI images (automatic segmentation)

	low	estimation	high
First-order Bayes (pixel)	84.9	86.2	87.4
Backprop (pixel)	89.1	90.3	91.4
First-order Bayes (feature)	92.7	93.7	94.6
Backprop (feature)	91.9	92.9	93.8

Table III:

95% confidence interval for alphabetic character recognition correct rate with 300 PPI images (manual segmentation)

	low	estimation	high
First-order Bayes (pixel)	68.9	70.6	72.2
Backprop (pixel)	72.0	73.7	75.3
First-order Bayes (feature)	89.0	90.1	91.1
Backprop (feature)	87.4	88.6	89.7

local minima in backpropagation error surface. Although optimization techniques have been proposed to find global minimum [Was88, Bab90], their effectiveness in complex problems remains to be seen. Furthermore, the classical backpropagation is vulnerable to other problems such as slow convergence and generalization. Many researchers have proposed methods to improve the speed of convergence [Fah88, IP90, ST90, Jac88] However, there is still some dispute in the definition of *convergence* itself. It has been observed that an *absolute convergence* may not provide the best generalization result. The effect of additional hidden units on generalization has not been thoroughly explored either. Under such conditions, its performance may also be degraded.

From the tables in section 5, we can make the following observations. The accuracies of Bayesian classifier and BP are comparable in most cases. BP perform better than the first order Bayes approximation with *pixel* input. This is not surprising considering the second classifier, being a first order approximation, is capable of achieving only linear decision surfaces. Contrarily, the activation function in a backpropagation network allows it to capture higher order relations.

In the case of *feature vector* input, their relative performance is not clear. The results in digit and character recognition experiments showed similar accuracy rates for both classifiers. Although the first-order Bayesian approximation has out performed backpropagation network in two experiments, their confidence intervals overlap considerably. If the feature components are indeed statistically independent, the first-order Bayesian classifier provides optimal decision, and backpropagation network is able to find good approximation to such decision. However, it is more likely that the features are not independent, and better local minima would have been found had the parameter space been explored more thoroughly.

We should also take into consideration that while the implementation of first order Bayes discriminant classifier is straight forward, the optimal configuration for a BP network can only be found with more research. Therefore, backpropagation has much more potential for improvement.

7 Conclusion

We have shown for a specific case that the outputs of a backpropagation network are direct estimations of a posteriori probabilities. Therefore, this neural network model has the same computational power as the Bayes decision rule. Empirical results obtained in real world handprinted characters recognition experiments showed that these two classifiers have comparable performances. A finer comparison would require a detailed analysis of the nature of input data. It is expected that if the input components are statistically independent, the Bayes classifier will be optimal. However, if any dependence exists among input components, backpropagation is likely to out-perform first order Bayesian classifier. Normally the limitation in computing time and space hinders higher order approximation of probability density functions. On the other hand, many techniques have been developed to mitigate the difficulties encountered in training a backpropagation. Under the circumstance, the neural network approach is a very promising alternative to Bayesian classification.

8 Acknowledgements

This work was supported in part by the United States Postal Service and by the Eastman Kodak Company.

References

- [Ara89] M. Arai. Mapping abilities of three-layer neural networks. *Proc. of International Joint Conference on Neural Networks*, 1:419-423, 1989.
- [Bab90] N. Baba. A hybrid algorithm for finding the global minimum of error function of neural networks. *Proc. of International Joint Conference on Neural Networks*, 1:585-588, 1990.
- [BGJH88] H Baird, H. P. Graf, L. D. Jackel, and W. E. Hubbard. A vlsi architecture for binary image classification. In *Proceedings of Cost 13 Workshop : From the Pixels to the Features*, Bonas, France, 1988.
- [Bur87] D. J. Burr. Experiments with a connectionist text reader. *Proc. of International Conference on Neural Networks*, 4:717-724, 1987.
- [Cas70] R. G. Casey. Moment normalization of handprinted characters. *IBM Journal of Research and Development*, Sept 1970.
- [CP34] C.S. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26, 1934.

- [DH73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, 1973.
- [DJG⁺89] J. Denker, L. Jackel, W. R. Gardner, H. P. Graf, D. Henderson, R. E. Howard, W. Hubbard, H. S. Baird, and I. Guyon. Neural network recognizer for handwritten zip code digits. In *Advances in Neural Information Processing Systems*, volume 1. Morgan Kaufmann, 1989.
- [Fah88] S. E. Fahlman. An empirical study of learning speed in back-propagation networks. Technical Report CMU-CS-88-162, CMU, 1988.
- [Fuk72] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 1972.
- [Fuk88] K. Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, 1:119-130, 1988.
- [GWG89] M. Gutierrez, J. Wang, and R. Grondin. Estimating hidden unit number for two-layer perceptrons. *Proc. of International Joint Conference on Neural Networks*, 1:677-681, 1989.
- [HH90] B. Horne and D. Hush. On the optimality of the sigmoid perceptron. *Proc. of International Joint Conference on Neural Networks*, 1:269-272, 1990.
- [Hig62] W. H. Highleyman. The design and analysis of pattern recognition experiments. *Bell System Technical Journal*, 41:723-744, March 1962.
- [HN89] R. Hecht-Nielsen. Theory of the backpropagation neural network. *Proc. of International Joint Conference on Neural Networks*, 1:593-605, 1989.
- [IM88] B. Irie and S. Miyake. Capabilities of three-layered perceptrons. *Proc. of International Conference on Neural Networks*, 1:641-648, 1988.
- [IP90] Y. Izui and A. Pentland. Speeding up back propagation. *Proc. of International Joint Conference on Neural Networks*, 1:639-642, 1990.
- [Jac88] R. A. Jacobs. Increased rates of convergence through learning rate adaption. *Neural Networks*, 1:295-307, 1988.
- [KH88] S. Y. Kung and J. N. Hwang. An algebraic projection analysis for optimal hidden units size and learning rates in backpropagation learning. *Proc. of International Joint Conference on Neural Networks*, 1:363-370, 1988.
- [Lip87] R. P. Lippman. An introduction to computing with neural nets. *IEEE ASSP*, 1987.
- [LLS90] D. S. Lee, S. W. Lam, and S. N. Srihari. A structural approach to recognize handprinted and degraded machine-printed characters. In *Proceedings of IAPR Syntactic and Structural Pattern Recognition Workshop*, Murray Hill, New Jersey, 1990.

- [MP69] M. Minsky and S. Papert. *Perceptrons*. MIT Press, 1969.
- [MRG86] J. L. McClelland, D. E. Rumelhart, and PDP Research Group. *Parallel Distributed Processing*. MIT Press, 1986.
- [Par85] D. Parker. Learning-logic. Technical Report TR-47, MIT, 1985.
- [Pav86] T. Pavlidis. A vectorizer and feature extractor for document recognition. *Computer Vision, Graphics, and Image Processing*, 35:111-127, 1986.
- [RMS89] A. Rajavelu, M. T. Musavi, and M. V. Shirvaikar. A neural network approach to character recognition. *Neural Networks*, 2:387-393, 1989.
- [Ros57] F. Rosenblatt. The perceptron : A perceiving and recognizing automaton. *Cornell Aeronaut. Lab Report*, 85-4601-1, 1957.
- [Shv90] H. Shvaytser. On the learning power of networks with a bounded fan-in layer. *Proc. of International Joint Conference on Neural Networks*, 1:313-316, 1990.
- [ST90] R. S. Scalero and N. Tepedelenlioglu. A fast training algorithm for neural networks. *Proc. of International Joint Conference on Neural Networks*, 1:715-718, 1990.
- [SW81] J. Sklansky and G. N. Wassel. *Pattern Classifiers and Trainable Machines*. Springer-Verlag, New York, 1981.
- [SW89] M. Stinchcombe and H. White. Universal approximation using feedforward networks with non-sigmoid hidden layer activation functions. *Proc. of International Joint Conference on Neural Networks*, 1:613-617, 1989.
- [TG74] J. T. Tou and R. C. Gonzalez. *Pattern Recognition Principles*. Addison-Wesley, 1974.
- [Was88] P. D. Wasserman. Combined backpropagation/cauchy machine. In *Proceedings of International Neural Network Society*, New York, 1988. Pergamon Press.
- [Wer74] P. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1974.
- [WK89] S. M. Weiss and I. Kapouleas. An empirical comparison of pattern recognition, neural nets, and machine learning classification methods. *Proc. of Eleventh International Joint Conference on Artificial Intelligence*, 1:781-787, 1989.

Appendix : Sample images used in experimentation.
Handwritten Digits (10 classes)

0000000000

1111/1111

2222222222

3333333333

4444444444

5555555555

6666666666

7777777777

8888888888

9999999999

Alphabets (52 classes)

A A A A A A A A A A

a a a a a a a a a a

B B B B B B B B B B

b b b b b b b b b b

C C C C C C C C C C

c c c c c c c c c c

D D D D D D D D D D

d d d d d d d d d d

E E E E E E E E E E

eeeeeeeeee

FFFFF F

ffff f fffff

GGGGGGGG

gggggggg

HHHHHHHH

hhhhhhhh

IIIIIIII

iiiiiiii

JJJJJJJJ