Statistical Science
1994, Vol. 9, No, 1, 2–54

Neural Networks: A Review from a Statistical Perspective

Bing Cheng and D. M. Titterington

Abstract. This paper informs a statistical readership about Artificial Neural Networks (ANNs), points out some of the links with statistical methodology and encourages cross-disciplinary research in the directions most likely to bear fruit. The areas of statistical interest are briefly outlined, and a series of examples indicates the flavor of ANN models. We then treat various topics in more depth. In each case, we describe the neural network architectures and training rules and provide a statistical commentary. The topics treated in this way are perceptrons (from single-unit to multilayer versions), Hopfield-type recurrent networks (including probabilistic versions strongly related to statistical physics and Gibbs distributions) and associative memory networks trained by so-called unsuperviszd learning rules. Perceptrons are shown to have strong associations with discriminant analysis and regression, and unsupervized networks with cluster analysis. The paper concludes with some thoughts on the future of the interface between neural networks and statistics.

Key words and phrases: Artificial neural networks, artificial intelligence, statistical pattern recognition, discriminant analysis, nonparametric regression, cluster analysis, incomplete data, Gibbs distributions.

1. INTRODUCTION

Given an appropriate notational convention, Figure 1 gives a diagrammatic representation of a multiple linear regression model in which the expected response, y, is related to the values $x = (x_1, \ldots, x_p)$ of covariates according to

$$y = w_0 + \sum_{j=1}^{p} w_j x_j.$$

The notational convention is that the circle represents a computational unit, into which the x_j's are fed and multiplied by the respective w_j's. The resulting products are added and then a further w_0 is added to provide the eventual output. In this way, we create a neural network representation of a very familiar statistical construct, because Figure 1 is a version of a standard neural network called the *simple* or *single-unit perceptron*.

In general, neural networks are (the mathematical models represented by) a collection of simple computational units interlinked by a system of connections. The number of units can be very large and the connections intricate.

Neural networks are used for many applications of pattern classification and pattern recognition:

- Speech recognition and speech generation
- Prediction of financial indices such as currency exchange rates
- Location of radar point sources
- Optimization of chemical processes
- Target recognition and mine detection
- Identification of cancerous cells
- Recognition of chromosomal abnormalities
- Detection of ventricular fibrillation
- Prediction of re-entry trajectories of spacecraft
- Automatic recognition of handwritten characters
- Sexing of faces
- Recognition of coins of different denominations
- Solution of optimal routing problems such as the Traveling Salesman Problem
- Discrimination of chaos from noise in the prediction of time series

Bing Cheng is Lecturer in Statistics, Institute of Mathematics and Statistics, University of Kent, Canterbury, Kent CT2 7NF, England. D. M. Titterington is Professor, Department of Statistics, University of Glasgow, Glasgow G12 8QQ, Scotland.

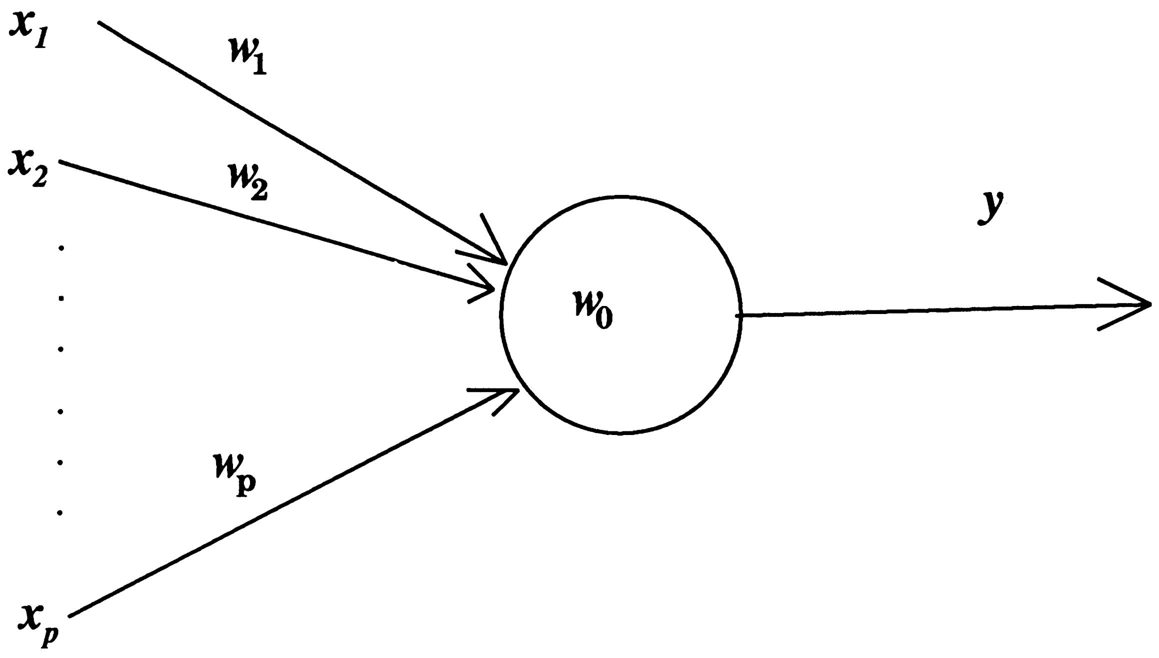

FIG. 1. *A simple (single-unit) perceptron.*

In addition, we use neural networks in robotics and in computer vision, as in the creation of a network that responds to certain visual stimuli in a way similar to the brain. Such neurological-type examples are, as yet, less common than the more prosaic applications listed in the previous paragraph. This is in spite of the fact that the initial stimulus for the development of ANN models was an effort to understand more deeply how the brain works and to construct a mechanism that would function in the same way.

The aim of artificial intelligence and neuroscience was to require the construction of a system that could compute, learn, remember and optimize in the same way as a human brain! It would not be sufficient to have a black box that came up with the right answers; rather, the answers had to be achieved by "human" mechanisms. It is generally accepted that this holy grail is still distant, and the pursuit continues. The explosive growth of activity in neural networks has, however, occurred because the frameworks that seemed reasonable prototypes for neurological modeling have been adopted and further developed as computational tools for many other fields. In particular, there are certain areas of this topic that are worthy of close attention from statisticians.

This paper is structured as follows: Section 2 gives some general reasons why statisticians should be interested in at least some of the neural-network research and, conversely, why neural-network specialists should be aware of certain statistical research. Section 3 provides the flavor of the topic through a series of examples. Sections 4 through 6 look at three broad areas in more depth. In each area, we outline the basic neural-network methodology, in terms of network architectures and training algorithms, and then present a commentary on the most important statistical points of contact. In particular, the commentary sections, while giving broad indications of the interface, include fairly detailed references to the relevant neural-network literature and, to a lesser extent, to the corresponding statistical literature. Section 4 looks at the feed-forward networks known as perceptrons, which are usually trained by a so-called supervized-learning procedure and which are used in contexts strongly related to discriminant analysis, regression and time-series analysis. Section 5 considers Hopfield-type recurrent networks: probabilistic versions, such as the Boltzmann machines, have many points of contact with statistical physics and Markov random fields, through their association with Gibbs distributions. Section 6 discusses networks trained by unsupervized learning, emphasizing their relationship with cluster analysis. Section 7 discusses the future of common interests in neural-network and statistical research. Important areas include methods for designing network architecture (model choice), methods for assessing performance, methods for parameter estimation and the identification of problem areas in which the neural-network approach is necessary.

The literature on ANNs is vast and is expanding rapidly. We found the texts by Muller and Reinhardt (1990) and Hertz, Krogh and Palmer (1991) and the review by Hinton (1989) of particular interest. Johnson and Brown (1988) provide an informal and readable account of the history, personalities and possible future directions of the field. Important, more specialized monographs include those of Minsky and Papert (1969, 1988), Rumelhart, McClelland, and the PDP Research Group (1986) and Amit (1989). In addition, there are increasingly many compilations, usually representing published conference proceedings. These include Anderson and Rosenfeld (1988), Aleksander (1989), Antognetti and Milutinovic (1991), Eckmiller (1990), Eckmiller and Von Der Malsburg (1988), Eckmiller, Hartmann and Hauske (1990), Kohonen et al., (1991) and Gelenbe (1991b). One such volume (Sethi and Jain, 1991) makes a specific claim to try to draw together the fields of ANN research and statistical pattern recognition, and Hunt et al. (1992) alerts the control engineering community to the relevance of neural networks to their subject.

Several research journals are dedicated to the field, but the total coverage includes dozens of other journals in the literatures of engineering, theoretical biology, pattern recognition, artificial intelligence, computer science, theoretical physics, applied mathematics and, embryonically, statistics.

2. WHY SHOULD STATISTICIANS BE INTERESTED?

Statisticians should become aware of, and involved in, research related to neural networks on several grounds.

2.1 Neural Networks Provide a Representational Framework for Familiar Statistical Constructs

Many ideas and activities familiar to the statistician can be expressed in neural-network notation. Our paper started with one simple case (which we will discuss further in Section 4.3), but they include regression models from simple linear regression to projection pursuit regression, nonparametric regression (Specht, 1991), generalized additive models and others (see Section 4.3.2). Also included are many approaches to discriminant analysis such as logistic regression, Fisher's linear discriminant function (LDF) and classification trees, as well as methods for density estimation of both parametric and nonparametric types: the former is exemplified by finite mixture models (Tråvén, 1991), and the latter is exemplified by kernel-based density estimation (Specht, 1990). Finally, we can include graphical interaction models. We refer to the statistical literature on these topics during the text.

In most of these cases, the statistician may react to the fact that familiar entities can be given a (usually pictorial) representation by adopting neural-network notation with a "so what?" attitude. However, the relationship is clearly introducing the neural-network community to certain statistical ideas, and the points of contact in certain areas, nonlinear regression, in particular, are leading to important research under some of the following headings.

2.2 Many Common Problems of Modeling and Inference Have Both Statistical and Neural-Network Treatments

Even for the small list of applications in Section 1, statisticians will feel that they should have some technique in their own armory to carry out a suitable analysis. Given a pattern classification problem and a training set of previously classified items, the statistician would probably try to construct an appropriate discriminant function to classify future items. The simplest version of this for the 2-class problem is Fisher's LDF (Fisher, 1936; Hand, 1981), in which the classification decision depends on the sign of

$$w^T x + w_0, \tag{1}$$

where x is the vector of indicants or feature variables corresponding to the new item and w and w_0 are, respectively, a vector of coefficients and a scalar. Fisher's LDF corresponds to a particular formula for w and w_0 expressed in terms of the training data. In the neural-network literature, linear discriminant functions such as (1) are also proposed, representing the *single-unit perceptron* alluded to in Section 1. The practical difference between this device and the statistical version lies in the way the training data are used to dictate the values used for w and w_0. They will almost never correspond to Fisher's LDF, and it is natural to enquire about the extent to which the two methods differ; see Section 4.1 for further details.

Discriminant analysis can be thought of as a special type of regression or prediction problem with an indicator variable or vector as the response. Many of the practical problems dealt with using neural networks concern regression or prediction in a more general sense. It turns out that there are two main aspects to the treatment of any given practical problem:

(i) specifying the architecture of a suitable network; and
(ii) training the network to perform well with reference to a training set.

When, as in the context of discriminant analysis, the training set consists of *previously classified* items, (ii) is called a *supervized learning* procedure.

To the statistician, this is equivalent to

(i) specifying a regression model; and
(ii) estimating the parameters of the model given a set of data.

The differences between the two approaches lie in the ways in which (i) and (ii) are handled. The neural-network specialist will resolve (i) by constructing a network of nodes and links from which a regression function can be written down, whereas the statistician usually extracts the regression function as the mean of a conditional probability model for the response, given the covariates. Whichever approach is taken, (i) clearly poses questions of model choice. As far as (ii) is concerned in the neural-network literature, the network is adjusted to predict the responses of the training data as well as possible. The statistician, however, will typically resort to some general technique, such as maximum likelihood estimation, Bayesian inference or some nonparametric approach. In some cases, the neural-network recipe turns out to be equivalent to maximum likelihood analysis if a familiar error structure is assumed. However, the traditional neural-network approach proposes an optimality criterion without any mention of 'random' errors and probability models.

The most common neural-network approach to regression-type problems is *multilayer perceptrons* and generalizations of *single-layer perceptrons*. They are discussed in more detail in Section 4.2 and are compared with statistical competitors. These competitors are virtually all representable as multilayer perceptrons; however, they are typically comparatively simple in form, in contrast to some of the very intricate networks that have been constructed, after considerable time and effort, to treat specific applications. For an example, see the discussion of the recognition of hand-written Zip-code characters in Example 3.2. It is important to investigate to what extent 'standard' prescriptions can compete with custom-built networks, to look critically at approaches to network design (model choice) and to compare the different approaches to the (usually) heavy numerical optimization exercise required to train the networks in stage (ii) above.

As with discriminant analysis and regression, another activity common to the two research communities is what statisticians refer to as *cluster analysis*: a set of multivariate observations have to be organized or, in a sense, organize themselves into a number of mutually disparate, but internally compact, groups or clusters. The number of clusters may or may not be prescribed.

One way to think of cluster analysis is as a discriminant analysis but without the knowledge of the true class identifiers for the training set. In the terminology used in the neural-network literature, this represents *unsupervized learning*, and we shall discuss a few networks that self-organize using unsupervized learning rules to recognize certain types of pattern.

2.3 Statistical Techniques Are Sometimes Implementable Using Neural-Network Technology

We remarked in Section 2.2 that Fisher's LDF provided one linear rule for 2-class discriminant analysis. The neural-network community have their own ways of constructing linear rules, but they also have a particular method for computing the Fisher's LDF itself (Kuhnel and Travan, 1991). In addition, there are neural-network procedures for computing quadratic discriminant rules (Lim, Alder and Hadingham, 1992) for calculating principal components (Oja, 1982; Sanger, 1989) for approximating Bayesian probabilities (Richard and Lippmann, 1992) and even for approximating the rejection region for the elementary likelihood-ratio test between two simple hypotheses (Bas and Marks, 1991). The statistical community might express surprise that there is any need for a new approach to these familiar procedures in applied matrix algebra, in view of the existence of well-tried packages for eigenanalysis. However, standard packages impose a limit on the size of matrix that can be treated, and some neural-network applications involve data of very high dimension.

2.4 Some Neural Networks Have Probabilistic Elements

In most applications of neural networks that generate regression-like output, there is no explicit mention of randomness. Instead, the aim is function approximation. Although the optimality criterion used to choose the approximant may be a least-squares criterion or a cross-entropy function, there is no thought that this criterion should be interpreted as a log-likelihood function.

However, some networks do have explicit probabilistic components in their definition. Of particular interest are probabilistic versions of Hopfield networks, and developments thereof, such as Boltzmann machines. We will discuss these in Section 5.2. It is often possible to identify such networks with certain exponential family distributions (Gibbs/Boltzmann distributions). There is relevant material in the statistical physics literature as well as in the modern statistical literature related to applications of simulated annealing, Gibbs sampling and generalizations thereof and the information geometry associated with S. Amari and others.

2.5 An Increasing Effort to Embed Neural Networks in General Statistical Frameworks

There is an accelerating trend in neural-network literature to apply general statistical methodology. In some cases, the discussion is specific to the example: in speech recognition, for instance, there is current activity in comparing and blending multilayer perceptrons and hidden Markov (chain) models (Bourlard, 1990; Bourlard and Morgan, 1991; Bridle, 1992; Bengio et al., 1992). However, more general work exists, particularly in applying Bayesian formulations and methodology in the modeling of neural networks. Representative references are Buntine and Weigend (1991) and MacKay (1992a, b). See Section 4.3.5 for a more detailed discussion.

3. ELEMENTAL ASPECTS OF ARTIFICIAL NEURAL NETWORKS

3.1 The Neurological Origins of ANN Research

It is a mere half-century since the publication of arguably the first paper on ANN modeling by McCulloch and Pitts (1943). The early motivation was in artificial intelligence. It sought to discover why the human brain, although comparatively inadequate in terms of speed of serial computation, was spectacularly superior to any conceivable von Neumann computer in performing many thought processes or cognitive tasks. Modern microchips carry out, in nanoseconds, elementary operations for which the human brain requires milliseconds; yet the brain has little difficulty in correctly and immediately recognizing familiar objects from unfamiliar angles, an operation that would severely tax conventional computers. The crucial difference, therefore, lies not in the essential speed of processing but in the organization of the processing.

A key is the notion of *parallelism* or *connectionism*. The processing tasks in the brain are distributed among about $10^{11} - 10^{12}$ elementary nerve cells called *neurons*. Each neuron is *connected* to many others, can be *activated* by inputs from elsewhere and can likewise stimulate other neurons. The brain very quickly achieves complex tasks because of the vast number of neurons, the complex interneuron connections and the *massively parallel* way in which many simple operations are carried out simultaneously.

Other important characteristics of neurological activity are those of *adaptability* and *self-organization*. As we broaden our experience, our brain has to adapt in order to assimilate new knowledge and perspectives, and aspects of the neural structure have to organize themselves accordingly.

Research in artificial intelligence aims to discover and emulate the precise structure and mode of operation of the neural network in a real brain. This will involve expertise in psychology, neuroscience and computer science. Here we exploit, in nonneurological contexts, the structure of a large number of simple computational units interlinked in an appropriate way and with a well-defined mechanism for learning and adapting itself from experience, that is, from data.

3.2 The Structure of ANN Models

A basic feature of ANN models is a representation of a single *neuron*. Figure 2a provides a schematic diagram of a real neuron: its main features are the *nucleus* within the *cell body* (or *soma*), the *axon* and the *nerve fibres* (*dendrites*) leading from the soma. The axon sprouts root-like strands, each one terminating at a *synapse* on a dendrite or cell body of another neuron. A typical axon generates up to 10^3 synaptic connections with other neurons, and it is clear that the global system of connections in a brain is vastly complicated.

Figure 2b contains an even more crude representation of the neuron. This reveals the neural system as a directed graph involving *nodes* (the neuron cell bodies), sometimes called *units*, and internodal *connections* or *links* (the *synaptic links*). Signals are transmitted within pairs of units; sets of nodal outputs are created on the basis of inputs from other units; and the whole system evolves through time. A seminal step, taken by McCulloch and Pitts (1943), was to conceive a simple artificial neuron with the following structure (Figure 2c).

The McCulloch-Pitts neuron receives inputs from each of a set of other units that provide binary (±1) inputs $x = (x_1, \ldots, x_p)$ and output

$$y = \operatorname{sgn}\left(\sum_{j=1}^{p} w_j x_j + w_0\right). \tag{2}$$

The McCulloch-Pitts neuron is just a "binary" version of the regression net in Figure 1. In (2), the $\{w_j, j = 1, \ldots, p\}$ are called *connection weights*, *connection strengths* or *connectivities*; w_0 is a *bias* term and $\operatorname{sgn}(\cdot)$ denotes the sign function. In the trivial regression net of Figure 1, the connection weights are regression slope parameters and the bias is the intercept. In neurological terminology, the neuron fires ($y = +1$) or fails to fire ($y = -1$) accordingly as

$$\sum_{j=1}^{p} w_j x_j + w_0 > 0 \ (\leq 0).$$

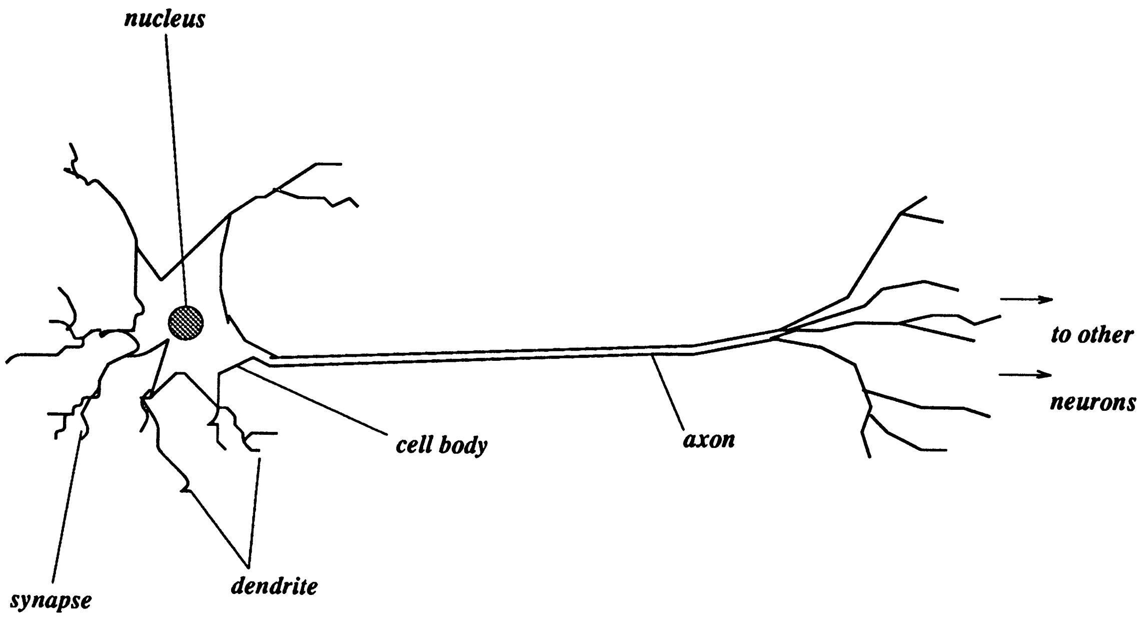

FIG. 2a. *Schematic diagram of real neuron.*

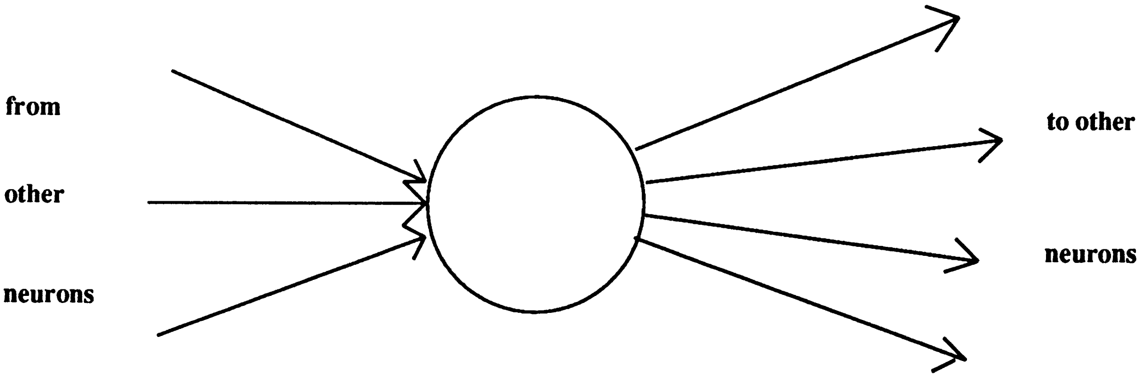

FIG. 2b. *Elemental version of* 2a.

In general, the input-output relationship at a neuron takes the form

$$y = f(\phi(x, w)), \tag{3}$$

where f and ϕ are prescribed functional forms, x represents the inputs (not necessarily binary) and w are the connection weights associated with connections leading *into* the unit. The function f is called the *activation function.*

Although there seems to be a redundancy in (3) in using both f and ϕ, it is helpful to use this notation. Usually, ϕ is linear as in (2), and f is chosen from a small selection of functions, including the following:

- $f(u) = \text{sgn}(u) = f_h(u)$, the *hard limiter nonlinearity,* produces binary (± 1) output.
- $f(u) = \{\text{sgn}(u) + 1\}/2$ produces binary (0/1) output.
- $f(u) = (1+e^{-u})^{-1} = f_s(u)$, the *sigmoidal* (*logistic*) *nonlinearity*, produces output between 0 and 1.
- $f(u) = \tanh(u)$ produces output between -1 and 1.
- $f(u) = (u)_+$ produces a non-negative output.
- $f(u) = +1$ with probability $f_s(u)$ and $f(u) = -1$ with probability $1 - f_s(u)$ provides random binary (± 1) output via logistic regression (Bridle, 1990).
- $f(u) = u$ is of course linear, as in our very first example in Section 1.

In practice, the units will usually have more than one output strand. The art of network construction in ANNs is to use simple individual units but to link together enough of them and in a suitable manner to solve a particular problem.

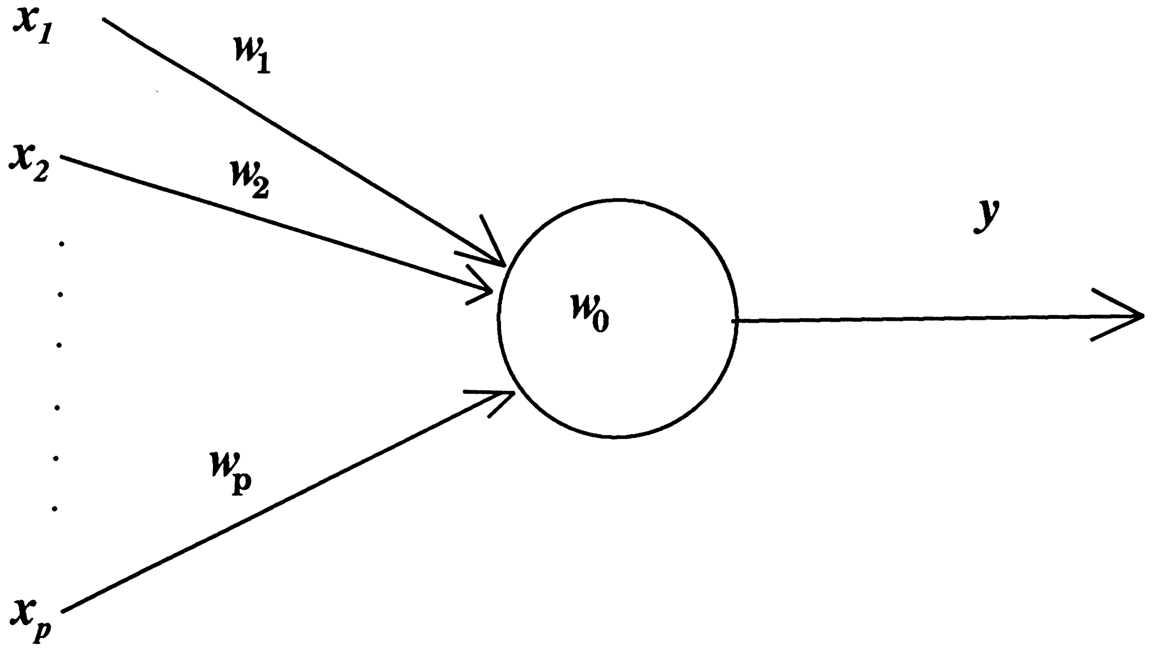

FIG. 2c. *The McCulloch-Pitts neuron.*

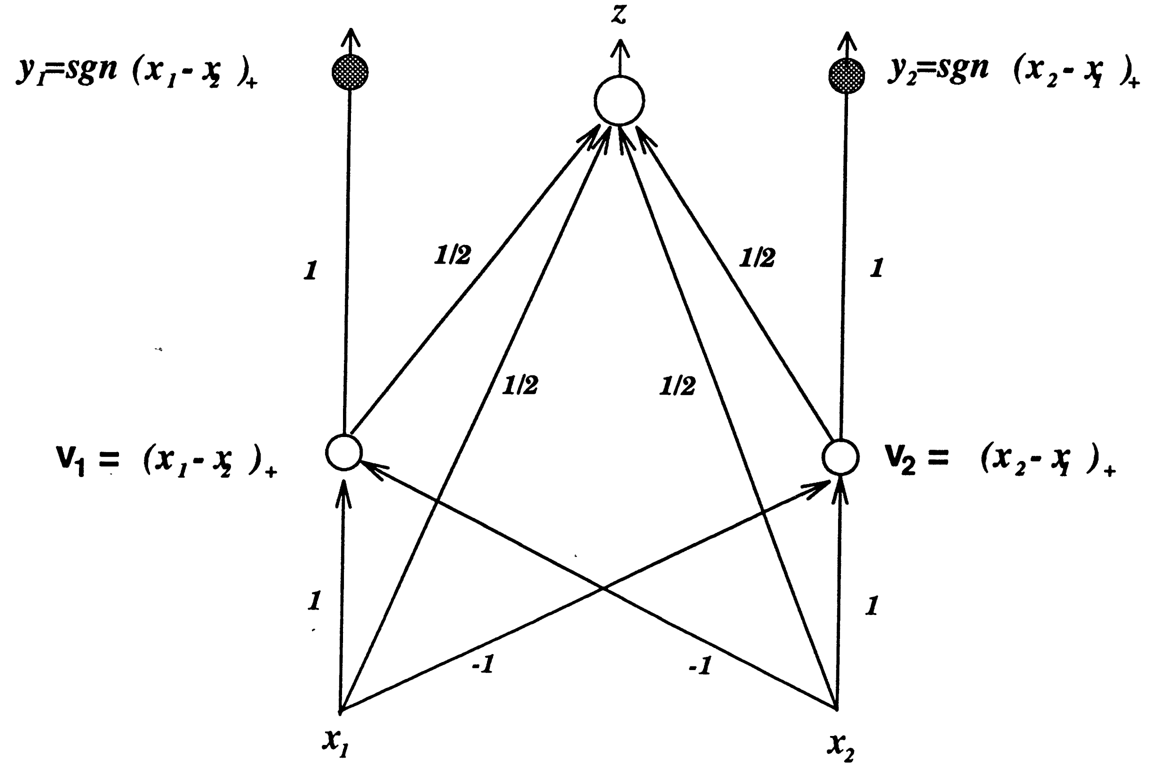

FIG. 3. *A network for finding the larger of two positive numbers, given eventually by* $z = \{\frac{1}{2}(x_1-x_2)_+ + \frac{1}{2}(x_1+x_2) + \frac{1}{2}(x_2-x_1)_+\}_+ = \max(x_1, x_2)$.

3.3 Some Illustrative Examples

Example 3.1. This first example, taken from the helpful review by Lippmann (1987), is trivial and non-statistical; but it helps to reinforce the notation. The network in Figure 3 identifies which of two nonnegative numbers is the larger, as well as displaying the number itself. The "inputs" are the two numbers (x_1, x_2), and there are three "output" nodes at the top: one fires ($y_1 = 1$) if $x_1 > x_2$, the second fires ($y_2 = 1$) if $x_2 > x_1$ and the third displays $z = \max(x_1, x_2)$. In the middle, there are two more nodes, called *hidden* nodes with outputs (v_1, v_2), that contribute towards the calculation. Figure 3 shows the network architecture, suitable values for the connection weights, the activation functions required at the different nodes and the progression of the calculation through the network. Solid discs correspond to units with hard-limiter nonlinearities ($f(u) = \text{sgn}(u) = f_h(u)$) and open circles to units with nonlinearities $f(u) = (u)_+$. Hidden units, which have no direct physical meaning and are, therefore, somewhat analogous to latent variables, are a feature of most practical ANN models.

In simple problems like this, we can both construct a network and assign activation functions and weights that do the required job perfectly. In most applications, however, this is not feasible, and the network is used only as an approximation in the same spirit as statistical modeling. This naturally complicates the issues of designing the architecture and activation functions and choosing the associated parameters (the connection weights and biases).

Example 3.2. *A network for Zip-code recognition.* As an example of a much larger network, we consider the one developed by Le Cun et al. (1989) for recognizing hand-written Zip-codes. The training data consisted of 7291 hand-written Zip-code digits preprocessed to fit a 16 × 16 pixel image with grey levels in the range -1 to $+1$. In this case, the dimensionality of each input, x, is $p = 256$.

The network architecture, depicted in Figure 4, consists of an input layer of 256 units (laid out, in view of the context, as a 16 × 16 array) leading up through three layers of hidden units to an output layer of 10 units that corresponds to the desired digits $\{0, 1, \ldots, 9\}$. The essence of the construction of the three hidden layers $\{H_1, H_2, H_3\}$ and the inter-layer connections is as follows (for more detail, see Le Cun et al., 1989):

1. *Layer* H_1. This layer contains 768 units arranged in 12 8 × 8 squares. Each unit in each of the 8×8 squares receives inputs from a 5×5 square receptive field within the input image. The receptive fields leading to adjacent units in the H_1-layer are two pixels apart so that the input image is undersampled and some information about position is lost. All units in a given 8×8 H_1-square use the same connection weight but have different biases. Thus, the H_1-layer acts as an array of feature detectors picking up features without regard to position. The number of parameters involved in the (input $\rightarrow H_1$) connections is clearly $(25 \times 12) + 768 = 1068$.

2. *Layer* H_2. This layer contains 12 4 × 4 squares of units. The connections from H_1 to H_2 are similar in character to those from the input layer to H_1, and the H_2-squares are also designed to detect features. Each H_2-unit combines information from 5 × 5 squares, identically located in 8 of the 12 squares in H_1. Thus, 200 H_1-units contribute to the input of each H_2-unit. As before, the sets of weights (but not the biases) for all 16 units in a given 4×4 square in H_2 are constrained to be the same. Thus, associated with the 192 H_2-units, there are 12×200 connection weights and 192 biases: a total of 2592 free parameters.

3. *Layer* H_3. Layer H_3 is straightforward, consisting of 30 units. The scheme of connections is straightforward too, all H_2 units being linked with all H_3 units. (The two layers are *fully connected.*) This results in $(30\times192)+30 = 5790$ parameters. Layer H_3 is, in turn, fully connected to the output layer, requiring $(10\times30)+10 = 310$ parameters.

Altogether, therefore, the network involves 1256 units, 63,660 connections and 9760 independent parameters! The part of the network above layer H_2 enables a flexible discriminant rule to be created based on what are presumed to be useful classification features created in the H_2-units.

This network represents a very highly parameterized model, but the training data set was also large, of the form $\{(x^{(r)}, z^{(r)}), r = 1, \ldots, 7291\}$, in which each $x^{(r)}$ represents 256 pixels and each $z^{(r)}$ is a 10-dimensional indicator of the true digit. The network belongs to the class of *multilayer perceptrons*, mentioned earlier in Section 2.2 and discussed in more detail in Section 4.2, where the issue of training is also described. When Le Cun et al. (1989) applied the resulting discriminant rule to the training set, only 10 (0.14%) of the 7291 images were misclassified. As usual, this is an unrealistically low error rate so far as predicting future performance is concerned. When the rule was applied to a test set of 2007 further digits, 102 (5.0%) mistakes were made.

The level of performance of an ANN on the universe of possible data (not just on the training data) is called its *generalization ability*; empirical assessment normally requires a large test set of typical cases. Generalization ability is impaired if the ANN is overfitted to the training data, usually by allowing too many free parameters. For the Zip-code problem Le Cun et al. (1990) further reduced the number of free parameters by a factor of about four and achieved a substantial improvement in performance on the test set.

Example 3.3. *NETtalk* (*Sejnowski and Rosenberg,* 1987). Figure 5 displays the architecture of

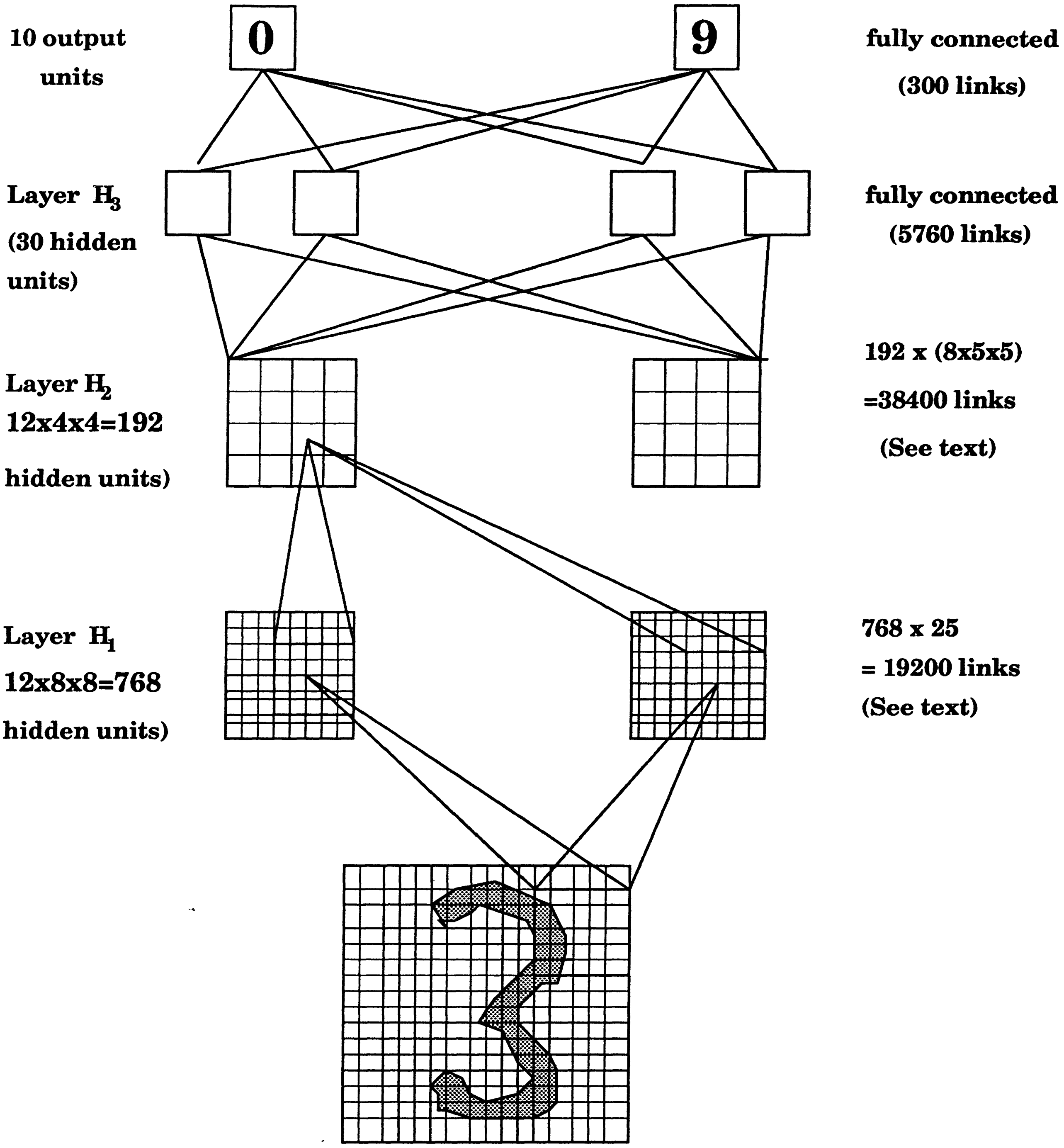
10 output units
0
9
fully connected (300 links)
Layer H_3 (30 hidden units)
fully connected (5760 links)
Layer H_2 12x4x4=192 hidden units)
192 x (8x5x5) =38400 links (See text)
Layer H_1 12x8x8=768 hidden units)
768 x 25 = 19200 links (See text)

FIG. 4. *The network developed by Le Cun et al.* (1989) *for Zip-code recognition.*

the NETtalk network designed to learn to speak English. The network scans English text and, at any instant, seven consecutive characters make up the input. The corresponding output is a phoneme code, subsequently transmitted to a speech generator, that represents the symbol at the middle of the input window. There were 7 × 29 input units representing indicators of the presence/absence in each of the seven positions of members of the alphabet of 26 letters and 3 punctuation characters. There were 80 units in the single hidden layer and 26 output units. As in Example 3.2, it is envisaged that the hidden units create useful discriminant features that are merged into a powerful classification rule at the output layer. A training set of 1024 words and their associated phoneme codings led to the creation of intelligible speech after 10 iterations of the learning rule and to 95% accuracy after 50 itera-

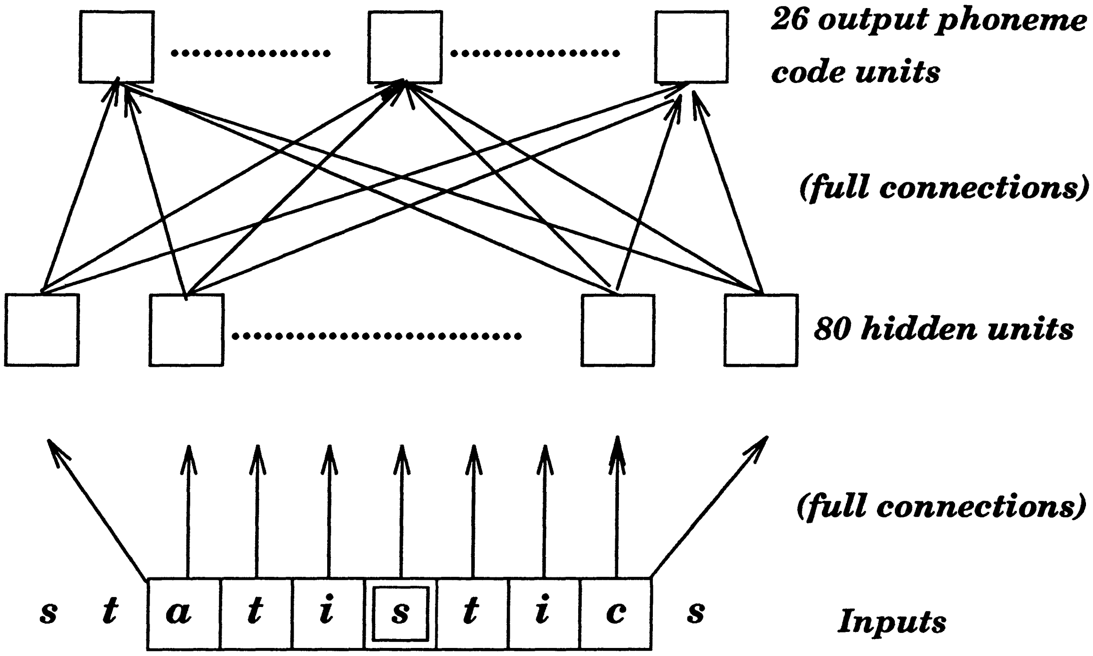

FIG. 5. *Structure of network used in NETtalk.*

tions. The learning behavior resembled a child's early speech in that the first features apprehended were the points of separation between words. Some of the hidden units could be given interpretations, for instance, as discriminators between vowels and consonants. Again, note the analogy with latent variables in statistics.

Generalization ability was assessed using a test set, and 78% accuracy was achieved, representing quite intelligible speech. If the network was "damaged" by removing some hidden units, performance was degraded a little but recovered after retraining (i.e., reestimating the remaining parameters). Resistance to partial damage is an important property of neural networks in contrast to serial computing, in which a single small change or error can have catastrophic consequences. Sophisticated rule-based speech generators often out-perform machines such as NETtalk, but the latter does well in view of its simplicity of construction and training.

Examples 3.2 and 3.3 are both examples of multilayer perceptrons of which there is a multitude of further applications including medical prognosis (Lowe and Webb, 1990). In fact, they are so common that the phrase "Artificial Neural Networks" is often taken to be synonymous with "multilayer perceptrons." However, there are other types of network architecture with important applications, and we give a taste of these next.

Example 3.4. *An associative* (*Hopfield*) *network for digit recognition.* The training set in Example 3.2 contained many cases from each of the ten underlying classes, corresponding to the digits $\{0, 1, \dots, 9\}$. In *associative memories*, each class is represented by an exemplar. When an observed pattern, usually a partial or noisy version of an exemplar, is presented, the memory should identify the correct uncorrupted exemplar. The concept underlying such ANN models is to mimic the capacity of the human brain to store a library of patterns and to be able to associate one of them with a newly observed pattern. The term *content-addressable* is also used in that the observed pattern is identified (correctly, one hopes) on the basis of its content.

Figures 6a and 6b display the results of the application of a basic, deterministic, Hopfield network (Hopfield, 1982) to digit recognition. The digits are presented as 9×7 binary images; thus each pattern x is p-dimensional, where $p = 63$. The learning process (i.e., the method of storing the exemplars in the memory) and the rule for processing observed patterns are described in Section 5.1. Here we merely report some results.

Figure 6a shows the exemplars and the result of presenting the pure exemplars to the trained network. The digits $\{4, 6, 7\}$ are correctly recognized and 0 almost is, but the rest are not! Table 1 gives the distances, in terms of the numbers of pixels on which they disagree, between the final states and the desired exemplars. It also shows how many iterations were required.

Figure 6b explores the robustness of the memory when the pure 4 and 7 are distorted by error. The colour of each pixel was changed, with probability

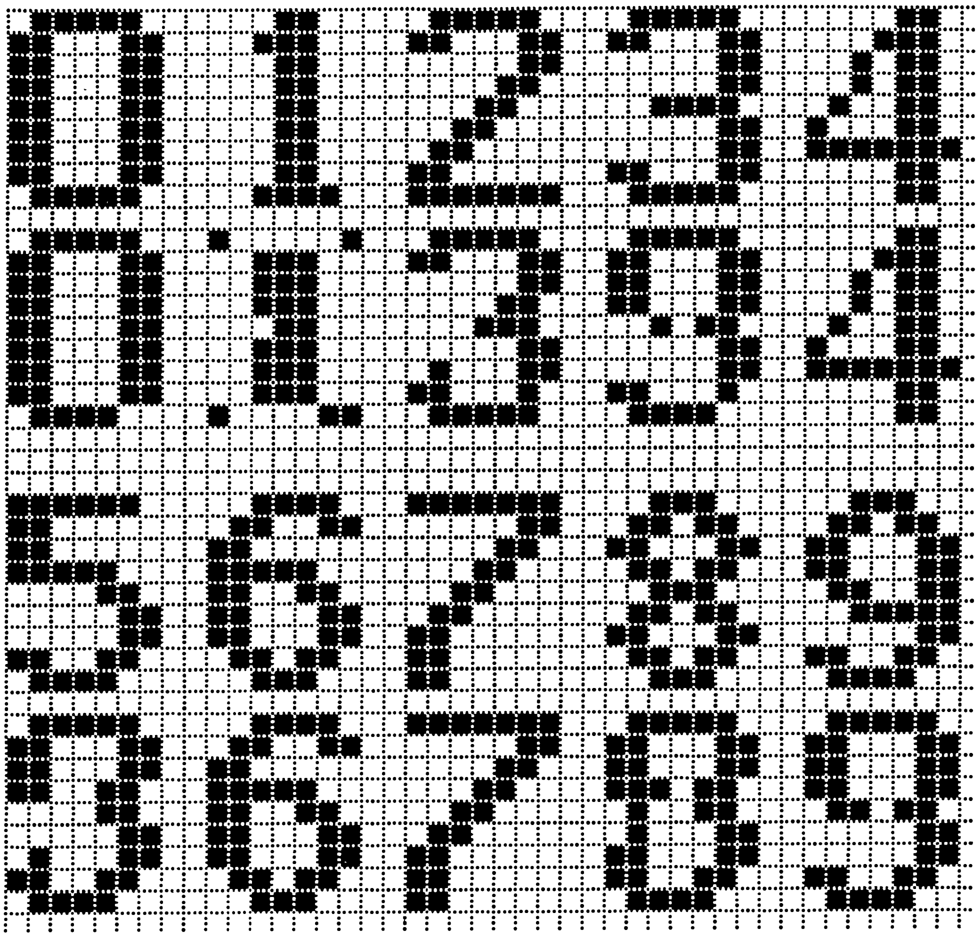

FIG. 6a. *Performance of Hopfield network on pure exemplars.*

TABLE 1
Some quantitative indices on the performance for Figure 6a: (-) denotes number of pixels different from exemplar; [-] denotes number of iterations needed for convergence

Pure exemplar	0	1	2	3	4	5	6	7	8	9
Limit point	(1) [1]	(14) [2]	(11) [4]	(7) [2]	(0) [0]	(9) [2]	(0) [0]	(0) [0]	(17) [4]	(11) [3]

$\pi \in \{0(0.05)0.25\}$, independently of the other pixels. Table 2 provides quantitative results as in Table 1. For more discussion of this example see Cheng and Titterington (1994).

In Section 5, we will look at Hopfield networks in more detail. In particular, we will reveal the relationship between probabilistic versions and such topics as spin-glass models, Gibbs distributions, Markov chain Monte Carlo and the EM algorithm.

Example 3.5. *Cluster analysis by adaptive resonance theory* (*ART*). In cluster analysis, it is uncommon for the number of clusters, let alone their locations, to be specified beforehand. Instead, the analysis uses a training set of (unclassified) items, according to some unsupervized learning algorithm, and allows the number of clusters to be determined by the data. In adaptive resonance theory (ART) (Carpenter and Grossberg, 1988) cluster centers are created and are modified, and the associated clusters grow as items in the training set are sequentially incorporated. A new item is either assigned to an existing cluster and the cluster center adapted accordingly, or it becomes the center of a new cluster if implausibly far from (that is, if it does not "resonate" with) any existing cluster center.

Example 3.6. *Representation of distributions using feature maps.* Figure 7a shows a single-layer network typical of simple versions of Kohonen's self-

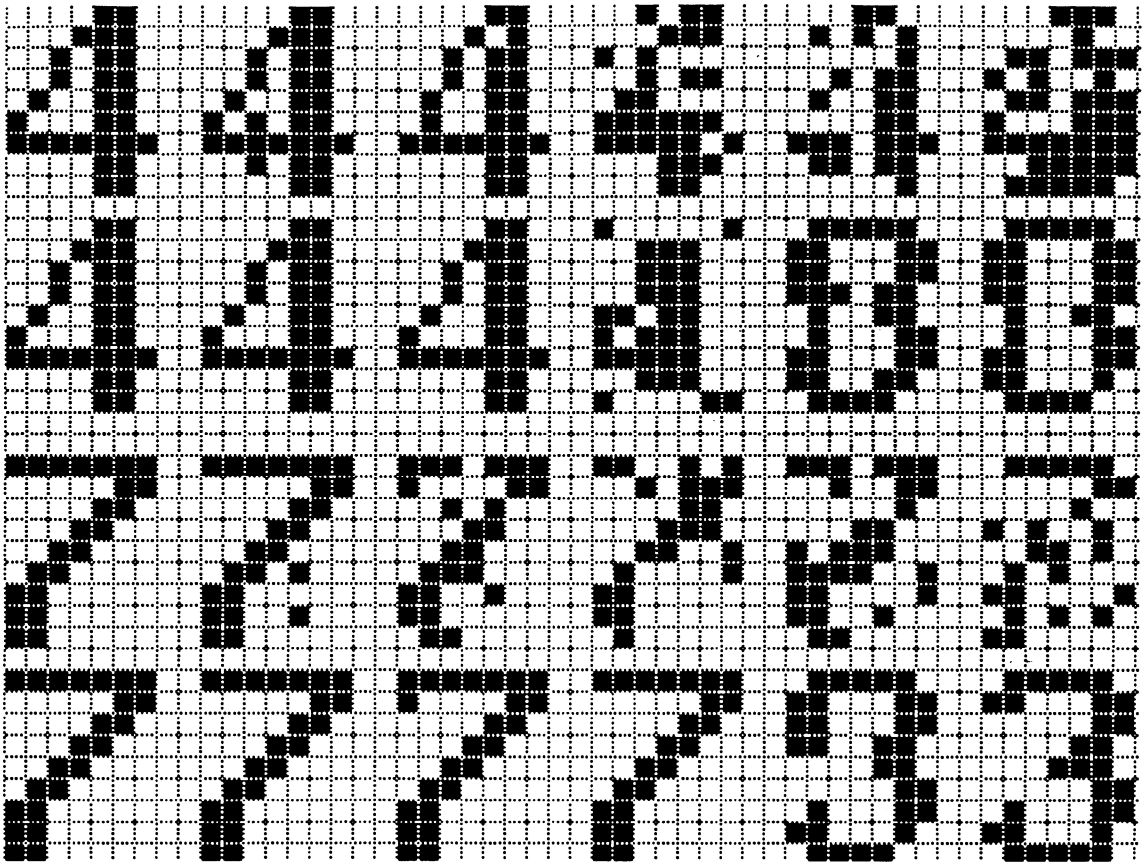

FIG. 6b. *Performance of Hopfield network on error-corrupted input.*

TABLE 2

Some quantitative indices on the performance for Figure 6b: (-) *denotes number of pixels different from exemplar;* [-] *denotes number of iterations needed for convergence*

π	0	0.05	0.1	0.15	0.2	0.25
Initial input: 4	(0)	(2)	(2)	(14)	(11)	(16)
Limit point	(0)	(0) [1]	(0) [1]	(25) [3]	(28) [3]	(32) [5]
Initial input: 7	(0)	(2)	(8)	(10)	(12)	(15)
Limit point	(0)	(0) [1]	(1) [1]	(0) [1]	(28) [3]	(24) [5]

organizing feature maps. The inputs here are of dimension $p = 2$, and there are full connections to the output units. The aim is to display the main features of the (frequency) distribution of input vectors. A particular learning rule (see Section 6.1) updates the weight vectors between the inputs and the output units as input vectors are presented. Any given input vector causes a particular output node to fire, leading to changes in the weights along the corresponding links and also, but usually to a lesser degree, to changes in weights along links to neighboring output nodes. There may also be lateral connections between pairs of output nodes: excitatory (positive weights) if the nodes are close, inhibitory (negative weights) between somewhat more distant nodes and, ultimately, as internodal distance increases, of zero strength.

After the training phase, a plot can be drawn of the weight pairs (one from each input link) associated with the output nodes. Figure 7b, analogous to Figure 9.12 of Hertz, Krogh and Palmer (1991), schematically shows the result of training an 8 × 8 output layer based on a very long sequence of bivariate observations uniformly distributed on, respectively, a disc, a triangle and an L-shape. The distribution of the input-to-output weight pairs, represented as the 64 mesh points in the plots, reflects the uniformity.

Kohonen (1990) lists many applications of

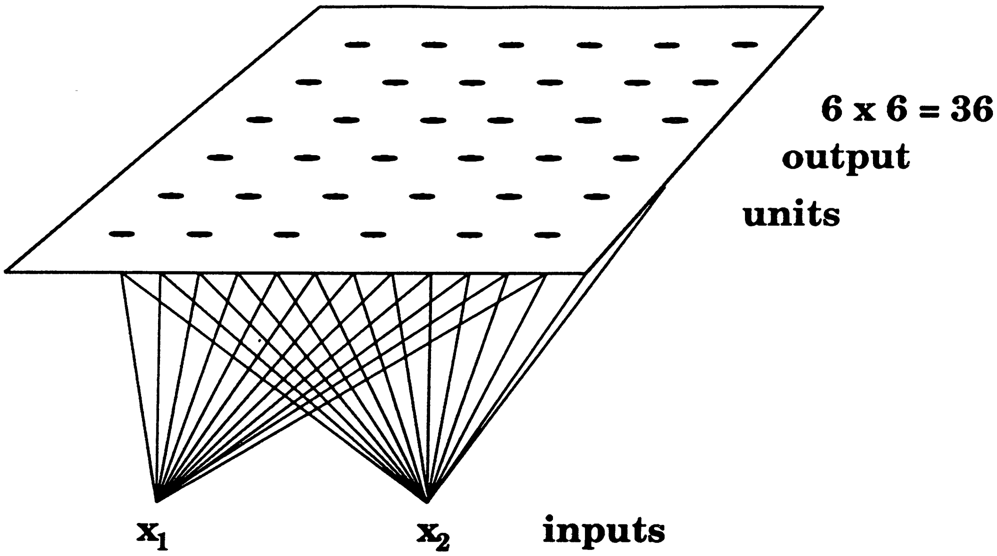

FIG. 7a. *Kohonen network.*

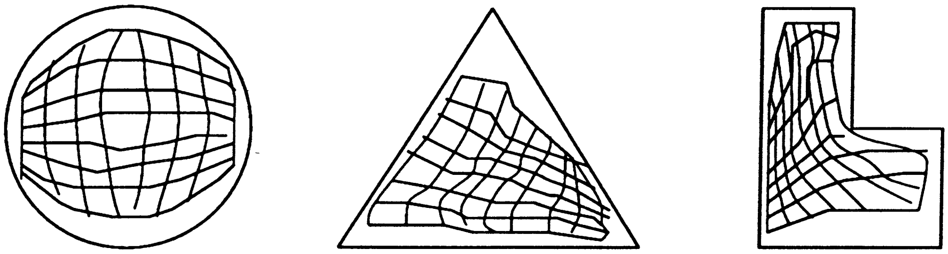

FIG. 7b. *Schematic performance on uniform data on various spaces.*

this method, including representation of speech phonemes and colours and imitation of both speech (the Finnish phonetic typewriter) and handwriting.

In the following sections we describe more formally some of the main types of network, the associated learning rules and the important areas of common research interest with statistics.

4. MULTILAYER PERCEPTRONS

Networks are used in practice to process a set of items, such as speech patterns or digits requiring recognition or patients requiring diagnosis. Each item is associated with a p-vector, x, of measurable features and a target, z, which represents, for instance, the indicator of the true speech pattern, digit or disease category or a more general response. The target, z, is often a vector. The network receives the vector x as inputs and creates a (set of) outputs, y, as a predictor of the unknown z. The "formula" for y is a function of the network architecture, the set of activation functions and all the parameters.

4.1 The Simple (single-unit) Perceptron

4.1.1 Architecture

The architecture of the single-unit perceptron is that of Figure 1 or Figure 2c. A set of p input variables, x (now not necessarily binary), generate a binary output variable, y, through the formula

$$y = f_h\left(\sum_{j=1}^{p} w_j x_j + w_0\right).$$

A neater version is obtained by creating the dummy variable $x_0 \equiv 1$, so that

$$y = f_h\left(\sum_{j=0}^{p} w_j x_j\right) = f_h(w^T x),$$

where w and x are now $(p+1)$-dimensional. The training data are denoted by $D = \{(x^{(r)}, z^{(r)}), r = 1, \ldots N\}$, where $\{z^{(r)}\}$ are the class indicators (± 1) and $x^{(r)} = \{x_j^{(r)} : j = 0, \ldots, p\}$ is the feature vector corresponding to the rth observation.

4.1.2 Training

The *perceptron learning rule* is a recursive algorithm in which the weights are modified as the training data are processed. Suppose that an observation (x, z) from the training set is to be incorporated and that $y = y(w)$ denotes the (binary) prediction for z, given x, on the basis of the current values w for the weights and bias. Then, for $j = 0, \ldots, p, w_j$ changes according to

$$(4) \qquad w_j \rightarrow w_j + \Delta w_j,$$

where

$$(5) \qquad \Delta w_j = \eta(z - y)x_j = \eta\delta x_j.$$

In (5) δ is the error incurred by applying the current rule to the new observation; since z and y are both binary, w changes if, and only if, the current rule misclassifies the new observation. The parameter $\eta(> 0)$ is called the *learning rate*; and the learning rule, called the *delta rule,* has the flavor of a gradient descent method for optimization as now indicated.

Suppose we wish to minimize a function $E(w)$. Then the iterative step of the gradient descent algorithm takes w_j to $w_j + \Delta w_j$, where

$$\Delta w_j = -\eta \frac{\partial E(w)}{\partial w_j},$$

for some step-size $\eta > 0$. Suppose we now take

$$E(w) = \frac{1}{2} \sum_{r=1}^{N} (z^{(r)} - x^{(r)T} w)^2 = \sum_{r=1}^{N} e_r(z^{(r)}, x^{(r)T} w),$$

and consider a recursive version of steepest descent in which

$$\Delta w_j = -\eta \frac{\partial e_r(w)}{\partial w_j} = \eta(z^{(r)} - x^{(r)T} w) x_j^{(r)},$$

Then Δw_j matches (5) with $(z, y) = (z^{(r)}, x^{(r)T} w)$ so that (5) would be recursive-steepest-descent were y given by $x^T w$. See Widrow and Hoff (1960).

The *single-unit perceptron convergence theorem* (Rosenblatt, 1962; Minsky and Papert, 1969, 1988) essentially states that, if the two training sets of feature vectors, one corresponding to each of the two classes, can be separated in R^p by a hyperplane, then the delta rule converges to give one such hyperplane in a finite number of steps. In practice, this involves processing each member of the training set a number of times. Hyperplanes can also be constructed that separate the training sets in a prescribed optimal sense (Rujan, 1991; Wendemuth, 1993). Efron (1964) studied the working of the perceptron when the training sets are not linearly separable.

The case of m (> 2) classes involves $(m - 1)$ output units that are fully connected to the inputs. Output units are usually depicted in a layer, and the resulting network is called the *single-layer perceptron.* (The input layer is typically not counted.)

Publication of Rosenblatt (1962) led to a surge of activity in view of the apparent power of single perceptrons to learn, as established in the perceptron convergence theorem. This was substantially deflated by Minsky and Papert (1969), who pointed out that the scope for perceptrons was very limited. It was easy to identify elementary logical problems that single-layer perceptrons cannot solve. The most famous of these is the XOR ("exclusive-or") problem of discovering whether or not two binary variables are equal. Networks could be devized to solve such problems, but there seemed to be no obvious learning rule until the work of Rumelhart, Hinton and Williams (1986a, b) that we will discuss in Section 4.2.2.

4.2 Multilayer Perceptrons (MLP)

4.2.1 Architectures

Multilayer perceptrons are far more flexible prediction mechanisms. Figure 8 shows a 2-layer version with a single output node and one layer of hidden units. Figure 5 showed another 2-layer perceptron and Figure 4 showed a 4-layer example. Other ANN architectures consist of interlinked input, output and hidden nodes, but the multilayer perceptron has the following special features.

- The hidden nodes are arranged in a series of layers.
- With the inputs at the bottom and the outputs at the top, the only permissible connections are between nodes in consecutive layers and directed upwards. In consequence, the multilayer perceptron is called a *feed-forward network*.

Weights are specified for all connections. Biases and activation functions are proposed for each of the hidden and output nodes. The outputs need not be binary.

Suppose the output v_k from the kth of the M hidden units in Figure 8 is given by

$$(6) \qquad v_k = g_k(\psi_k(x, \nu_k)), \qquad k = 1, \ldots, M,$$

and that the single output y is

$$(7) \qquad y = f(\phi(v, w)).$$

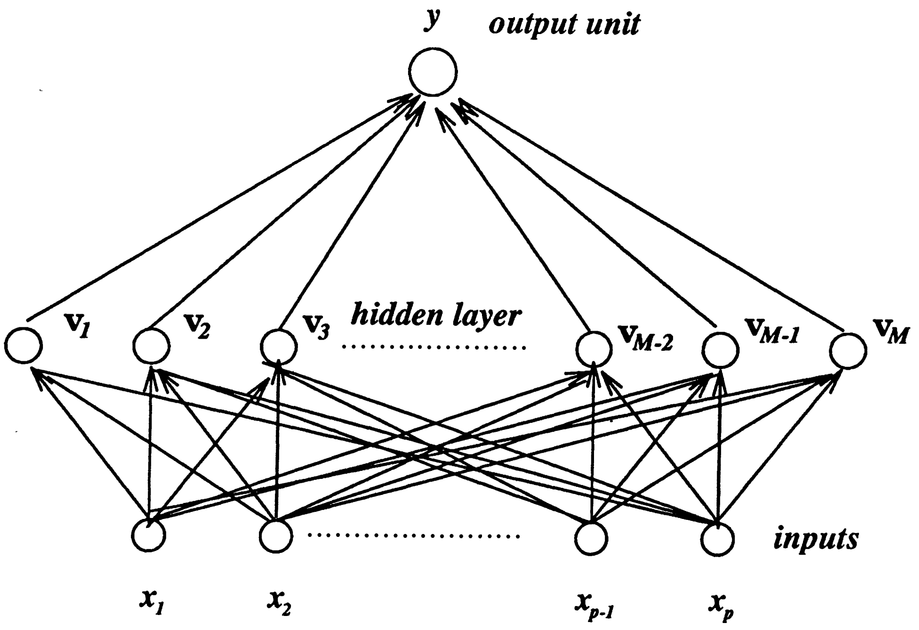

FIG. 8. *A single-output 2-layer perceptron.*

Then the expression of y as a function of x is a complicated nonlinear regression function with, as parameters, the $M+1$ sets of weights $\nu_1, \ldots, \nu_M, w$.

Various special cases exist.

Example 4.1. Suppose $f(\phi(v,w)) \equiv v^T 1 + w_0$ and, for each k,

$$v_k = \psi_k(x^T \nu_k).$$

Then

$$y = w_0 + \sum_{k=1}^{M} \psi_k(x^T \nu_k),$$

defining a class of additive/linear models equivalent to projection pursuit models (Friedman and Stuetzle, 1981).

Example 4.2. *Generalized additive models* (*Hastie and Tibshirani,* 1990). Suppose $M = p, v_{0k} = 0$ and $v_{ik} = \delta_{ik}$ (the Kronecker δ), and f is as in Example 4.1. Then

$$y = w_0 + \sum_{k=1}^{p} \psi_k(x_k),$$

defining a generalized additive model.

Example 4.3. Here $f(\phi(v,w)) = w_0 + \Sigma_k v_k w_k$ and, for each k,

$$v_k = g(x^T \nu_k + \nu_{0k}).$$

Thus

$$(8) \qquad y = w_0 + \sum_{k=1}^{M} w_k g\left(\sum_{i=1}^{p} x_i \nu_{ik} + \nu_{0k}\right).$$

The case where g is a sigmoidal nonlinearity corresponds to the model discussed by Barron (1991) and used by Nychka et al. (1992).

Example 4.4. *Radial basis function approach* (*Broomhead and Lowe,* 1988; *Moody and Darken,* 1989). Here

$$y = w_0 + \sum_{k=1}^{M} w_k \tau^{-p} \phi(\|x - c_k\|/\tau),$$

where ϕ is called a radial basis function, the $\{c_k\}$ are points in R^p and τ is a scale parameter. The function $\phi(\cdot)$ corresponds to a spherically symmetric function such as the p-variate Gaussian density. This method has clear similarities with kernel-type nonparametric methods (Lowe, 1991) and fixed-knot spline regression. Variations based on regularization are discussed in Poggio and Girosi (1990), Girosi and Poggio (1990) and Poggio (1990). A similar network based on wavelets is discussed by Zhang and Benveniste (1992).

4.2.2 Training

We define the prediction error criterion

$$E = E(W) = \sum_{r=1}^{N} \Delta(z^{(r)}, y^{(r)}(W)),$$

where W denotes all the weights, Δ is a measure of disparity and $y^{(r)}(W)$ is the prediction for $z^{(r)}$, computed as a function of W and $x^{(r)}$. For the perceptron defined by (6) and (7), for instance,

$$\begin{aligned} y^{(r)} &= y^{(r)}(W, \{\nu_k\}) \\ &= f(\phi[\{g_k(\psi_k(x^{(r)}.\nu_k)), k = 1, \ldots, M\}, w]). \end{aligned}$$

If y is a vector of continuous-valued components, it is common to use the Euclidean norm

$$\Delta(z,y) = \|z - y\|_2^2 \tag{9}$$

and weights W that minimize $E(W)$ are least-squares estimates. If y is an m-dimensional set of probabilities and z is an indicator vector, as in classification problems, a natural alternative to (9) is

$$\Delta(z,y) = -\sum_{j=1}^{m} z_j \log y_j. \tag{10}$$

This is equivalent to $KL(z,y) = \Sigma_{j=1}^{m} z_j \log(z_j/y_j)$, the Kullback-Leibler directed divergence (or cross-entropy) between z and y. Both (9) and (10) have been used to assess the performance of discriminant rules, (9) giving the so-called Brier score and (10) the logarithmic score; see Titterington et al. (1981). They are also equivalent to log-likelihood functions, (9) corresponding to standard Gaussian assumptions and (10) to quantal response models.

In practice, numerical methods are required to minimize $E(W)$, and techniques such as conjugate gradients, quasi-Newton algorithms, simulated annealing and genetic algorithms have been implemented. These methods are often much faster than the so-called method of *error backpropagation* (also called the *generalized delta rule*; Bryson and Ho, 1969; Werbos, 1974; Parker, 1985). Its creation was a major element in the explosive reemergence of interest in multilayer perceptrons in the mid-1980's (Rumelhart, Hinton and Williams, 1986a, b; Rumelhart, McClelland and the PDP Research Group, 1986).

As with the delta rule of section 4.1.2, the generalized delta rule is a gradient-descent algorithm. The algorithm uses the chain rule for differentiation and requires differentiable activation functions. The sigmoidal nonlinearity clearly satisfies this requirement but the hard-limiter does not. Hinton (1992) presents a lucid sketch of the method, of which a compelling feature was the fact that the calculations in the iterative step can be laid out on a network with the same architecture as the original perceptron but with the directions reversed (hence "backpropagation of errors"). Thus, in this sense, the ANN can do its own learning, which is an essential feature if the total procedure is to be plausible as a valid manifestation of artificial intelligence. However, convergence is so slow, even with modifications designed to speed it up, that it is clear that the brain does not learn by the generalized delta rule. In spite of this, the rule remains popular in the neural-network literature; the iterative steps involve the aggregation of simple calculations, localized within the network, no matter how massive the network might be. Whatever numerical method is used, the $E(W)$-surfaces are typically complicated with many local minima.

4.3 Statistical Commentary

4.3.1 Classification and discrimination

As mentioned in Section 2.2, the statistical literature contains various discriminant rules to compete with the single-unit perceptron, including Fisher's LDF (Fisher, 1936) and linear logistic regression for quantal response. See, for instance, Duda and Hart (1973), McLachlan (1992), Hand (1981), and Cox and Snell (1989). The LDF is a likelihood ratio or Bayes rule when the training sets are random samples from two equiv-covariant p-variate Gaussian distributions, and, if the covariance matrices are unequal, there are corresponding quadratic discriminant functions. All these recipes can be depicted as networks if we include input nodes corresponding to squares and products of the components of x. In general, such networks are called *higher order*. The neural-network literature includes its own approach to the creation of quadratic discriminant rules (Lim, Alder and Hadingham, 1992; Kressel, 1991). It would be of interest to compare all the rules in terms of criteria such as error rates, well researched for Fisher's LDF, and robustness to assumptions under which the model-based (statistical) rules are "optimal."

The use of hidden layers provides flexibility in the type of discriminant rule, but the nonneural-network approaches also have more sophisticated versions that are motivated by relaxing the assumptions about the underlying probability model $p(z,x)$. Since $p(z,x) = p(z|x)p(x) = p(x|z)p(z)$, the important modeling tasks involve $p(z|x)$ itself (on which logistic regression is based) or $p(x|z)$, the class-conditional densities of x. At the nonparametric extreme, kernel-based density estimates might be used for $p(x|z)$; see, for instance, Silverman (1986). One advantage of model-based methods is that, provided the model is valid, the estimated values of $\{p(z|x)\}$ indicate the relative plausibilities of the various possible classes for an item giving data x.

Other types of classifier include classification trees (Breiman et al., 1984; Quinlan, 1983), generalized additive models for quantal response (Hastie and Tibshirani, 1990) and regression by alternating conditional expectation (ACE) (Breiman and Ihaka, 1984). In addition, nonparametric rules include *k*-nearest neighbour (*k-NN*) procedures which assign an item to the majority class among the *k* training cases that are closest, in a prescribed sense, to the unclassified item.

Ripley (1993a) describes these approaches in more detail and compares some of them, empirically, with neural-network approaches, such as multilayer perceptrons, with various architectures and using various numerical procedures for training (backpropagation, quickprop and conjugate gradients), and linear vector quantization (LVQ; see Sections 6.1 and 6.2). He found that the nearest neighbor and LVQ methods worked well but, being "nonparametric", offered little in the way of explanation of the structure. Projection-pursuit regression filled that gap without excessive computing time, and the tree-based methods were fast and gave clear interpretation. On the other hand, the multilayer perceptrons took a long time to train and, in terms of results, had little to offer over simple methods such as nearest-neighbor methods. In a further empirical study, Ripley (1994a) compared Fisher's LDF, logistic regression, nearest-neighbor methods, multilayer perceptrons, trees, projection pursuit regression and Friedman's (1991) multivariate adaptive regression splines (MARS). Also see Section 4.3.2.

Fisher's LDF is still motivating new and potentially powerful classification tools for very high dimensional problems. Hastie, Buja and Tibshirani (1992) point out that LDF's overfit if the components of x are multitudinous (p very large) and highly correlated (because they are very highly parameterized), and they underfit, obviously, if the class boundaries are nonlinear. A natural approach to the first difficulty is to regularize as in ridge regression and smoothing-spline regression; see Titterington (1985), for instance. This is taken a stage further by Hastie, Buja and Tibshirani (1992) in their penalized discriminant analysis (PDA). They choose w to minimize

$$(11) \qquad \sum_{r=1}^{N}\{\theta(z^{(r)}) - x^{(r)T}w\}^2 + \lambda w^T \Omega w,$$

where $\{\theta(z)\}$ are a set of m optimal scores, one for each of the m classes, Ω is a nonnegative definite smoothing matrix and $\lambda(> 0)$ is a smoothing parameter. They applied the method to the Zip-code data of Example 3.2; recall that Le Cun et al. (1989) achieved 5% error rate on the test data. Hastie, Buja and Tibshirani (1992) achieved an error rate of 8.2% compared to the 11% incurred by standard linear discriminant analysis (LDA), and their approach involves 256 parameters compared with over 2000 in LDA. They also show that the PDA coefficients can provide helpful interpretation. Although the error rates are not as low as those of Le Cun et al. (1989), PDA is reasonably successful and constitutes a *general* approach in contrast to the intricate custom-built network of Le Cun et al. (1989). Other penalized varieties of LDA exist; see Friedman (1989), for example.

Key factors underlying PDA are the well-known relationships between LDA and both multiple linear regression and canonical correlation analysis (Mardia, Kent and Bibby, 1979). In their development of flexible discriminant analysis (FDA), Hastie, Tibshirani and Buja (1992) adopt the regression interpretation but generalize the form of the regression function. They adopt the additive model form (Example 4.2) and use a least-squares estimation criterion that is penalized by curvature penalties similar to those used in the definition of cubic splines (Silverman, 1985). Expression of the fitted functions in terms of spline basis functions leads to a quadratic optimization criterion similar to (11). Among several examples, they apply FDA and a variety of other methods, including multilayer perceptrons, to a set of vowel-recognition data; FDA performs encouragingly well. More systematic comparisons would be informative.

4.3.2 Regression

The most obvious statistical interpretation of multilayer perceptrons (MLP) is that they provide nonlinear regression functions that are estimated by optimizing some measure of fit to the training data. If the latter are noise-free, then the exercise is one of function approximation. There are many recent systematic developments in regression, and at least three important questions must be faced:

- How do the neural-network and statistical prescriptions compare in terms of "performance"?
- How good are various architectures at approximating members of particular classes of regression functions?
- What are the most reliable and practicable numerical methods for parameter estimation?

Examples 4.1 and 4.2 define two of the recent statistical developments: projection-pursuit regression and generalized additive models. Other innovations include Friedman's (1991) MARS (multivariate adaptive regression splines) and Tibshirani's (1992) modification of projection pursuit regression based on so-called slide functions.

In MARS, the model is

$$y = w_0 + \sum_{k=1}^{M} w_k \prod_{s=1}^{k_s} h_{sk}(x_{v(s,k)}),$$

where $v(s,k)$ is the index of the predictor used in the sth factor of the kth product. For k odd,

$$h_{sk}(x) = [x - t_{sk}]_+ ; h_{s,k+1}(x) = [t_{sk} - x]_+,$$

where the knot t_{sk} is one of the unique values of $x_{v(s,k)}$. Terms in the model are added and pruned to achieve a good fit to the training data and to ordinary stepwise regression. Barron and Xiao (1991) suggest a version of MARS (polynomial-based MAPS) that incorporates a roughness penalty in (11). Expressions like (11) also underlie standard smoothing splines (Silverman, 1985; Wahba, 1990), but high-dimensional versions of these are not practicable.

Tibshirani's (1992) variation of projection-pursuit regression is related to both (7) and (8). He suggests the model

$$y = w_0 + \sum_{k=1}^{M} w_k (x^T \nu_k - u_k)_+, \tag{12}$$

where x is p-dimensional, and he calls $(\cdot)_+$ the *slide* function. He exploits the result of Friedman and Silverman (1989) that there is an $O(N)$ algorithm for finding the knots, $\{u_k\}$. His algorithm also uses Breiman's (1993) so-called hinge-function fitting. On albeit small-scale examples, the method compared favourably with MARS and multilayer perceptron fitting.

We now consider the question of how well these models approximate arbitrary underlying regression functions. As a rule, the more hidden layers there are and/or the more nodes there are within each layer, the more flexible are the resulting fitted functions. To obtain concrete results, we have to impose smoothness constraints on the target function, but results are available that, taken at face value, seem impressive. For instance (Lorentz, 1966), every continuous function on $[0,1]^p$ can be exactly represented by a function of the form

$$y(x) = \sum_{j=1}^{2p+1} f_j \left(\sum_{k=1}^{p} \psi_{jk}(x_k) \right).$$

However, although the ψ_{jk} are independent of the true function, the f_j are not; see also Barron and Barron (1988). As a further example, Cybenko (1989), White (1990) and Hornik, Stinchcombe and White (1989) showed that continuous functions on compact subsets of R^p can be uniformly approximated by 2-layer perceptrons with sigmoidal activation functions, as defined in Example 4.3, a model involving $m(p+2)+1$ parameters. White (1989) establishes some statistical theory. Barron (1993) shows that, for true functions which satisfy a smoothness constraint given by a bound on the first moment of the magnitude distribution of the Fourier transform, this same network achieves integrated squared error $O(1/M)$ in contrast to the $O((1/M)^{2/p})$ suffered by ordinary series expansions. Thus, the perceptron of Example 4.3 offers superior parsimony of parametrization; see Barron (1989, 1992, 1994). Similar results are available in the context of sinusoidal activation functions (Jones, 1992), slide functions (Tibshirani, 1992) and wavelet networks (Zhang and Benveniste, 1992).

It is important to be aware of such practical limitations. In some situations in which it appears that a multilayer perceptron with one hidden layer is adequate, the number of nodes required can be prohibitive. In addition, results involving smoothness constraints on the underlying fitted surface may rule out functions of genuine practical interest. In general, the practical implications of these results require careful appraisal and there is a need for more constructive results; see related remarks in Section 4.3.3. As Geman, Bienenstock and Doursat (1992) point out, although the models are nominally parametric, the flexibility required of the models implies that we are effectively in a *nonparametric* regression context.

If we turn now to the question of numerical algorithms, it seems that for moderately sized problems methods such as conjugate gradients are much faster than the generalized delta rule. In very large problems, the network structure underlying the latter and its capacity for massively parallel processing may revive its attraction. In the Zip-code example, Le Cun et al. (1989) used a modified Newton algorithm in which a diagonal approximation to the Hessian matrix eliminates the most time-consuming component of the algorithm. The Gauss-Newton algorithm, ubiquitously popular in nonlinear least-squares (Seber and Wild, 1989), is also a candidate as in Tibshirani's (1992) interpretation of Breiman's (1993) method for fitting hinge functions. Gawthrop and Sbarbaro (1990) use a recursive Gauss-Newton algorithm. The simplifying feature of Gauss-Newton is its avoidance of second derivatives of the function being optimized. Webb, Lowe and Bedworth (1988) compare various methods.

4.3.3 *Time series analysis*

A central problem of nonlinear time series analysis is to construct a function, $F : R^d \to R^1$, in a dynamical system with the form

$$Z_t = F(Z_{t-1}, \dots, Z_{t-d}),$$

or possibly involving a mixture of chaos and randomness,

$$Z_t = F(Z_{t-1}, \dots, Z_{t-d}) + \varepsilon_t,$$

in which F is a chaotic map and $\{\epsilon_t\}$ denotes noise. Various types of ANN have been used to approximate the unknown F. Casdagli (1989) and Moody and Darken (1989) used a radial basis functions (RBF) network, while Sanger (1989) and Nychka et al. (1992) used multilayer perceptrons to duplicate results of Farmer and Sidorowich (1989) and Lapedes and Farber (1987). Nychka et al. (1992) illustrated the technique on the chaotic Mackey-Glass differential delay equation. Stokbro, Umberger and Hertz (1990) generalized the normalized RBF network

$$\hat{F}(x) = \sum_{k=1}^{M} F_k P_k(x),$$

where F_k are scalar parameters, and $\{P_k(.)\}$ are normalized versions of RBF to a neural network whose hidden units have localized receptive fields. Thus,

$$\hat{F}(x) = \sum_{k=1}^{M} (a_k + b_k(x - x_k)\sigma_k^{-1}) P_k(x),$$

where the x_k are pre-specified p-dimensional vector parameters, and the σ_k are scalar parameters. Given σ_k, the coefficients a_k and b_k are estimated by minimizing

$$E = \sum_{t=1}^{N} (Z_t - \hat{F}(Z_{t-1}, \dots, Z_{t-d}))^2.$$

Stockbro, Umberger and Hertz (1990) report simulations of the reconstruction of the one-dimensional logistic map,

$$Z_t = F(Z_t) = \lambda Z_t(1 - Z_t),$$

and of the Mackey-Glass equation.

It has been found that the dynamical features of the input signal affect the response of an ANN. For example, Mpitsos and Burton (1992) use error backpropagation to train a two-layer network using inputs from three sources: (i) chaotic data from the logistic map with $\lambda = 3.95$; (ii) white noise; and (iii) sinusoidal data. They find that learning is more effective when given chaotic inputs than in the random case. In general it is still not clear how an ANN responds to "irregular" input signals and this is clearly an interesting problem.

4.3.4 *Architecture design and generalization ability*

Section 4.3.2 revealed how some MLPs act as universal approximators, but care has to be taken not to fit overly intricate models. Overfitting the model fits part of the noise in the training set in addition to the underlying structure leading to a substantial difference between the abilities of the fitted model to back-predict the training data and predict future responses. The ability to perform well for items not in the training data is known as the *generalization* ability. It is important to find a suitable compromise between overfitting and underfitting: the latter results in a biased model.

It is familiar in discriminant analysis that naive error rates based on the training set over-estimate the true generalizability. Test sets provide an empirical estimate of the true error rate (Hand, 1981; Titterington et al., 1981); in the absence of a test-set, devices such as leave-one-out cross-validation can be applied (Lachenbruch and Mickey, 1968; Stone, 1974). This notion is also useful in more general regression contexts. Suppose $\hat{y}_r = \hat{y}_r(x, w^{(r)})$ represents a fitted model based on all training data apart from $(z^{(r)}, x^{(r)})$. Then the simple cross-validation average prediction loss is

$$CV(\hat{y}) = N^{-1} \sum_{r=1}^{N} \Delta(z^{(r)}, \hat{y}_r(x^{(r)}, w^{(r)})),$$

where $\hat{y}$ denotes the underlying fitted model and $\Delta(.,.)$ is some measure of loss, such as squared Euclidean distance. It is natural to choose a network to minimize $CV(\hat{y})$ or some similar quantity, such as the generalized cross-validation (GCV) criterion introduced by Craven and Wahba (1979) and used with MARS by Friedman (1991). As Barron and Xiao (1991) remark, these criteria are closely related to model-choice procedures such as Mallows' C_p (Mallows, 1973), Akaike's AIC (Akaike, 1974), Schwartz's (1978) Bayesian BIC method and the minimum description length (MDL) of Rissanen (1987), which is asymptotically equivalent to BIC. BIC typically chooses more parsimonious models than does AIC; see Barron (1991), Barron and Xiao (1991), Shibata (1981) and Li (1987) for further discussion. These criteria are now being used in the neural-networks literature; see for instance Levin, Tishby and Solla (1990), who follow the Bayesian path to the MDL criterion, and Wolpert (1992).

A further modification that combats overfitting in a Bayesian-like way is the *weight-decay* method (Hinton, 1986, 1989; Hertz, Krogh and Palmer, 1991, Section 6.6). The method involves adding a penalty term to the residual sum of squares that is proportional to the sum of the squares of all the weights.

Generalization ability measures performance averaged over the complete ensemble of possible cases. In practice, often empirical measures are only available based on a test set or cross-validated training set, but some theoretical results exist. For certain cases, Baum and Hausler (1989) bound the probability of a prescribed disparity between the empirical training-set error-rate and the ensemble error-rate. The bound is a function of the Vapnik-Chervonenkis dimension, VCdim, (Vapnik, 1982) of the space of functions representable by the network; it is an index of the capacity of the network. As Ripley (1994a) remarks, their bound implies that a two-layer network with M hidden nodes requires a training set of size N equal to about $\#(W)/\epsilon$ to guarantee a success rate of at most ϵ worse than that for the training set, where $\#(W)$ denotes the number of weights. It would be of interest to know how this result is affected if cross-validatory error rates are used for the training set. For results on average-case rather than worst-case generalization abilities, see Levin, Tishby and Solla (1990). Moody(1992) also contains interesting developments.

As Example 3.2 illustrates, some networks involve huge numbers of parameters even after considerable simplification of the parameterization. Parsimony can be also be achieved if it is appropriate to impose invariance requirements; see Bienenstock and Von der Malsburg (1987), Perantonis and Lisboa (1992) and Fukumi et al. (1992).

In Geman, Bienenstock and Doursat's (1992) lucid account of nonparametric regression in neural-network contexts, they exploit the above role for the VCdim in sketching the asymptotic theory of mean-squared error estimation of regression functions. They make the crucial remark that reassuring asymptotic results are likely to be rendered irrelevant in practice by the "curse of dimensionality". Typical training sets are almost inevitably too small by orders of magnitude for the asymptotic theory to be a reliable guide.

The study of generalization ability and the development of methodology for network design are among the most important current areas of research.

4.3.5 Bayesian modeling of multilayer perceptrons

As indicated in Section 4.2.2, the "traditional" approaches to weight selection are closely related to maximum likelihood estimation under certain assumptions about noise processes assumed for the data. It is also natural to explore the Bayesian approach. Recall that D denotes the training data, let A denote the network architecture, including the activation functions, and let θ denote all parameters within the model. Usually, $\theta = (W, \beta)$ where β denotes parameters associated with the noise model for D; W is usually associated with the means. Then, if P generically denotes probability density function,

$$P(D, \theta, A) = P(D|\theta, A)P(\theta|A)P(A). \tag{13}$$

If the architecture (i.e., the model) is prescribed, then we can start from

$$P(D, \theta) = P(D|\theta)P(\theta). \tag{14}$$

The first factors on the right-hand sides of (13) and (14) are the likelihood terms. If, for instance, the expectation of z, given x, is $f(x, W)$ and if there is independent additive Gaussian noise with variance β^{-1}, the likelihood term is proportional to

$$\beta^{N/2}\exp\left\{-\frac{1}{2}\beta\sum_{r=1}^{N}\|z^{(r)} - f(x^{(r)}, W)\|_2^2\right\}.$$

Particular questions of interest concern (i) inference about the parameters, W, in particular, given A; (ii) about the relative plausibilities of different architectures; and (iii) about the predictive distribution of an unknown z, given its x. In theory, these are all standard Bayesian computations. For (i), we require the posterior

$$P(\theta|D) = P(D|\theta)P(\theta)/P(D), \tag{15}$$

where

$$P(D) = \int P(D|\theta)P(\theta)d\theta. \tag{16}$$

For (ii), we need ratios like

$$\frac{P(A_2|D)}{P(A_1|D)} = \frac{P(D|A_1)}{P(D|A_2)}\frac{P(A_1)}{P(A_2)}, \tag{17}$$

where A_1 and A_2 are two possible architectures and

$$P(D|A_i) = \int P(D|\theta_i, A_i)P(\theta_i|A_i)d\theta_i,$$

$i = 1, 2$. For (iii), we need

$$(18)\quad P(z|x, T) = \{P(D)\}^{-1} \int P(z|x, \theta)P(D|\theta)P(\theta)d\theta.$$

In (17), the ratio $\{P(D|A_1)/P(D|A_2)\}$ is called a Bayes factor; see Smith and Spiegelhalter (1980) and Kass and Raftery (1993). Even computation of relative values of densities from (15) and (18) is generally not trivial, especially as the prior density, e.g., $P(\theta)$, usually includes hyperparameters requiring a further Bayesian stage. Computation of $P(D)$ and the Bayes factor is daunting. If the likelihood were Gaussian, $f(x, w)$ were linear in w and conjugate priors chosen, explicit results are available, in theory. Such approximations were adopted by MacKay (1992a, b). Neal (1992a, 1993) uses variants of the Markov-chain Monte Carlo approach (Besag and Green, 1993; Smith and Roberts, 1993). In particular he finds that basic Gibbs sampling is not adequate and relies on the hybrid Monte Carlo method of Duane et al. (1987). Buntine and Weigend (1991) provide a nice general account of the Bayesian approach, also mentioning Gaussian approximations. They point out the familiar equivalence of some Bayesian maximum a posterior (MAP) prescriptions to smoothing techniques of the regularization type and comment on the link, alluded to in Section 4.3.4 above, with Rissanen's MDL approach.

There is much to do in this area, both in terms of computational developments in dealing effectively with hyperparameters and in exploiting the interpretation of Bayesian procedures as smoothing mechanisms. As Geman, Bienenstock and Doursat (1992) emphasize, unless regularization is imposed by smoothing in some appropriate way, there is little hope of realistic networks being trainable using practical training sets.

5. ASSOCIATIVE MEMORIES OF THE HOPFIELD TYPE

5.1 Architectures and Training

In the basic Hopfield network (Hopfield, 1982), each feature vector, x, is a multivariate binary (± 1) vector. The objective is to associate with x one of a set of m exemplars that have been stored in the memory. One can think of the stored exemplars as a training set consisting of one representative for each of the class-types (Example 3.4).

In contrast to multilayer perceptrons, the outputs of this network are not explicit functions of the inputs. Instead, they are *stable states* of an iterative procedure, albeit one that terminates in finite time.

The Hopfield network processes an input pattern, x, as follows. Set $y^{(0)} = x$ and compute

$$(19)\quad \begin{aligned} y_i^{(n+1)} &= f_h\left(\sum_{j=1}^{p} w_{ij} y_j^{(n)}\right), \\ i &= 1, \ldots, p; \qquad n = 0, 1, \ldots \end{aligned}$$

where the weight matrix $W = \{w_{ij}\}$ is defined in terms of the exemplars $\{z^{(1)}, \ldots, z^{(m)}\}$ by

$$W = p^{-1} \sum_{j=1}^{m} z^{(j)}(z^{(j)})^T,$$

but with $w_{ii} = 0$, for all i. The way in which W is constructed is called Hebbian learning (Hebb, 1949). In (19), all components of $y_i^{(n)}$ are updated synchronously (Little, 1974). In the true Hopfield models, the components are updated asynchronously, that is, one at a time according to some deterministic or random schedule. The network is depicted in Figure 9 and shows the intra-layer looping typical of so-called *recurrent networks*.

A vital step in investigating convergence of the algorithm (in terms of $y^{(n)}$ reaching a limit as $n \to \infty$, given x) is the identification of an "energy surface",

$$(20)\quad L(y) = -\frac{1}{2} y^T W y.$$

Since $y_i^2 = 1$, for all i, the diagonal elements of W are arbitrary as far as optimizing over y is concerned.

If W is symmetric, as in Hebbian learning, asynchronous updating leads to a decrease in energy provided that $y_i^{(n+1)} \neq y_i^{(n)}$. Iterative asynchronous updating, therefore, leads us to a local minimum of $L(y)$. All local minima are stable states of the updating rule. As Example 3.4 confirms, not all exemplars may be stable states, and convergence may occur to a stable state that is not an exemplar.

In general, the introduction by Hopfield (1982) of the energy function (20) has revealed links with statistical physics (Amit, Gutfreund and Sompolinsky, 1985a), general optimization theory and dynamical systems: for "energy function" read "Hamiltonian function," "objective function" or "Lyapunov function" and for "stable states" read "attractors." These analogies have led to calculations concerning storage capacity of such networks, although to make progress, we have to make certain assumptions about the structure of the m p-variate exemplars.

For instance, suppose all components of all exemplars are independently and randomly chosen ± 1.

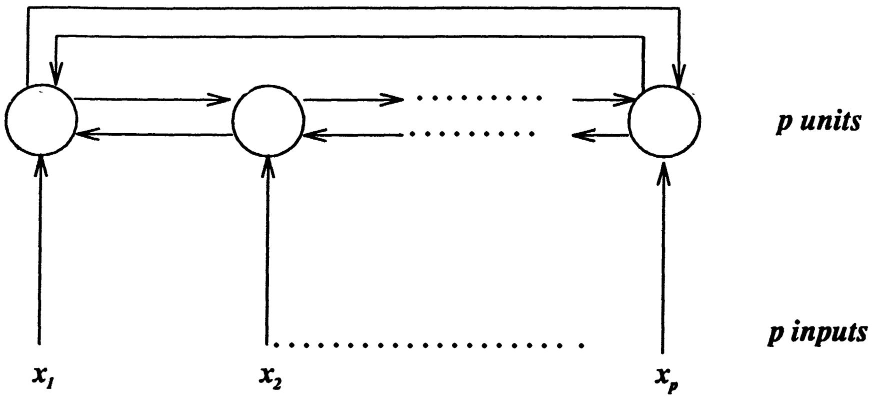

FIG. 9. *Architecture of Hopfield net.*

Then Amit, Gutfreund and Sompolinsky (1985b) showed that in the limit as $m,\ p \to \infty$ such that $m = \alpha p$, the efficient retrieval of memory requires that $\alpha \leq 0.14$. McEliece et al. (1987) showed that, if $m < p/(4\log p)$, then with probability one all m exemplars are stable states and, if $p/(4\log p) < m < p/(2\log p)$, then most exemplars are likely to be stable.

In Example 3.4, note that, with $m = 10$ and $p = 63$, $m(\log p)/p \approx 2/3$; it is not surprising that only a few of the exemplars are stable states. If, however, we store only the exemplars $\{1, 2, 3, 4\}$, they are all stable states. For more insight into the issue of memory capacity, see Newman (1988), Komlos and Paturi (1988) and Whittle (1991).

The basic Hopfield network was an important milestone in ANN research as an associative memory that implemented Hebb's ideas about learning and as a springboard for many future developments. For instance, Hopfield (1984), and Hopfield and Tank (1985) adapted the deterministic version to cater to continuous-valued variables and continuous-time updating. Bornholdt and Graudenz (1992) trained Hopfield-like networks using genetic algorithms (Goldberg, 1989).

5.2 Statistical Commentary

Of particular interest to applied probabilists, statistical physicists and statisticians is a further modification of the Hopfield network in which the hard-limiter activation function is replaced by a probabilistic rule based on a sigmoid nonlinearity; see Section 3.2. Thus, if y_i is to be updated, it becomes y_i', where

$$(21)\quad y_i' = \begin{cases} +1, \text{ with probability} \\ \qquad [1 + \exp\{-\sum_j w_{ij} y_j\}]^{-1} = f_s(w_i^T y) \\ -1, \text{ with probability } 1 - f_s(w_i^T y). \end{cases}$$

An asynchronous updating rule like (19) but based on (21) is equivalent to the Markov chain Monte Carlo technique known as Gibbs sampling or Glauber dynamics (Amit, 1989), and it leads to a formal link between the spin-glass models of statistical physics (Hertz, Krogh and Palmer, 1991), modern statistical image analysis (Geman and Geman, 1984) and the study of Boltzmann/Gibbs distributions in general.

Consider the Gibbs distribution defined by

$$(22)\quad \begin{aligned} p(y) &= \{C(W)\}^{-1} \exp\left\{ + \sum_{i<j} w_{ij} y_i y_j \right\} \\ &= \{C(W)\}^{-1} \exp\{-L(y)\}, \end{aligned}$$

where $L(y)$ is given by (20). Suppose that the internodal links are such that the Markov chain defined by an updating rule based on (21) is irreducible and ergodic. Then, in the limit, a random realization from (22) is generated and the stationary (Boltzmann/Gibbs) distribution (22) can be identified with the network. Such networks are called Boltzmann machines (Ackley, Hinton and Sejnowski, 1985; Hinton and Sejnowski, 1986) particularly versions that include hidden units in order to create added flexibility. There are several aspects of statistical interest.

- Explicit inclusion of a scale parameter in the definition of $L(y)$ that is interpretable as a statistical physics "temperature" leads to simulated annealing algorithms for finding minimizers of $L(y)$ (temperature $\downarrow 0$) or simulated realizations from (22) itself (temperature $\downarrow 1$). The resulting minimization algorithms have been used to attack combinatorial optimization problems such as the travelling salesman problem with mixed success (Aarts and Korst, 1989).
- General information-geometric results about exponential-family distributions relate the

structure of algorithms used for training Boltzmann machines to fit a training set or to approximate desired stationary distributions to maximum-likelihood methodology (Amari, 1990; Amari, Kurato and Nagaoka, 1992; Byrne, 1992).

- For networks with hidden units, one can derive training algorithms that are versions of the EM algorithm (Dempster, Laird and Rubin, 1977), and the corresponding M-step is a version of the iterative proportional fitting procedure used in analyzing multiway contingency tables and elsewhere (Bishop, Fienberg and Holland, 1975; Csiszar and Tusnady, 1984; Byrne, 1992). The case of polytomous, rather than binary, units is worked out in Anderson and Titterington (1993).

There are many interesting variations on these probabilistic networks; see Smolensky (1986); Campbell, Sherrington and Wong (1989); Hertz, Krogh and Palmer (1991); Amit (1989), for examples. Whittle's (1991) antiphon inserts randomness differently into the network at the input stage to a unit rather than the output. He assesses the capacity of his networks using information-theoretic criteria. The capacity of stochastic Hopfield-type networks is discussed in Hertz, Krogh and Palmer (1991), Amit (1989) and Amit, Gutfreund and Sompolinsky (1985b). Neal (1992b) develops Boltzmann-like machines based on belief networks (Pearl, 1988; Spiegelhalter and Lauritzen, 1990) and uses a Gibbs sampler to train the network on the basis of training data. He points out its superior learning speed over that of Boltzmann machines and indicates possible application to medical diagnosis problems. This is an area of interest to statisticians, because it is closely related to probabilistic expert systems (Lauritzen and Spiegelhalter, 1988) and graphical models (Whittaker, 1990). Somewhat different probabilistic networks are described by Gelenbe (1991a) and Bresshoff and Taylor (1990).

6. ASSOCIATIVE NETWORKS WITH UNSUPERVIZED LEARNING

6.1 Architecture and Training

The simplest associative networks are single-layer networks with m output units, all fully connected to the p inputs. In this context, the p-vector w_i denotes the connection weights between the inputs and the ith output unit and can be interpreted as the exemplar for that unit. In MAXNET (Lippmann, 1987), the output unit that fires is such that $\Delta(w_i, x)$ is smallest, where Δ is a measure of disparity and x is the input pattern. In the supervized case, with $\Delta(w_i, x) = \|w_i - x\|_2^2$, a gradient-descent learning rule for the $\{w_i\}$ can be derived as follows.

Recall from Section 4.1.1 that the training set is denoted by $D = \{(z^{(r)}, x^{(r)}), r = 1, \ldots, N\}$, where $z_{i^*}^{(r)} = 1$ if $x^{(r)}$ comes from cluster i^* and $z_i^{(r)} = 0$ for all other $i \in \{1, \ldots, m\}$. Define the objective function

$$E(W) = \frac{1}{2}\sum_{i=1}^{m}\sum_{r=1}^{N} z_i^{(r)}\|x^{(r)} - w_i\|_2^2. \tag{23}$$

Then a gradient-descent rule that modifies the set W of all weight vectors on the basis of incorporating the rth training case is given by the delta rule

$$\Delta w_i = \eta(x^{(r)} - w_i)z_i^{(r)}, \tag{24}$$

$i = 1, \ldots, m$. Of course, (23) can be minimized directly by

$$w_i = \sum_r z_i^{(r)} x^{(r)} \Big/ \sum_r z_i^{(r)},$$

the sample mean of patterns for which $z_i^{(r)} = 1$. In the unsupervized case, but still assuming that there are to be m clusters, (23) should be minimized with respect to the (missing) $\{z_i^{(r)}\}$ as well as the $\{w_i\}$. This leads to the m-means (usually described as "k-means") clustering algorithm. Rule (24) still obtains, provided we define

$$z_i^{(r)} = z_i^{(r)}(W) = \delta_{ii^*},$$

where

$$i^* = \arg\min_i \|x^{(r)} - w_i\|_2^2, \tag{25}$$

and δ denotes the Kronecker delta. The scheme represented by (24) and (25) is a simple example of *competitive learning*. The supervized version underlies Kohonen's (1989) learning vector quantization (LVQ) scheme for partitioning the input pattern space, and more general architectures are developed by Rumelhart and Zipser (1985). The same updating scheme is used within the adaptive resonance theory (ART) of Carpenter and Grossberg (1988), described in Example 3.5.

Extra features in Kohonen's (1989) self-organizing feature maps (Example 3.6) are the lateral connections between pairs of output nodes (also see Willshaw and Von der Malsburg, 1976), and the fact that weights associated with output nodes close to that which fires (through (25) also change. The delta rule is modified to become

$$\Delta w_i = \eta K(i, i^*)(x^{(r)} - w_i),$$

for all i, where $K(i, i^*)$ is a kernel-type function, monotonic decreasing in the distance between nodes i and i^* and zero outside a neighborhood of i^*. As the size of the training set increases, the neighborhood size is decreased as, usually, is η; see Kohonen et al. (1991) for applications and developments of this approach. In view of the learning rule, it turns out that the distribution of the m weight vectors should reflect the underlying probability density function of the input vectors. For details, see Ritter and Schulten (1986).

6.2 Statistical Commentary

Section 6.1 described only elementary versions of a large range of self-organizing neural networks trained by competitive learning algorithms based on (24) and (25). The essential points to note are the common learning rules and the relationship with statistical comparators from the literature on cluster analysis. See Hartigan (1975), Hand (1981), Gordon (1981), Van Ryzin (1977) and the report of the Panel on Discriminant Analysis and Clustering (1989). We commented in Section 6.1 on the link with the k-means clustering algorithm. Another approach is to assume that the unsupervized training data come from a mixture of m component distributions, that are often taken to be p-variate Gaussian. Training amounts to a statistical estimation exercise such as maximum likelihood estimation provided that the number of clusters, m, is specified. The problem of deciding what m should be from unsupervized training data is not straightforward, in spite of recent efforts. For a general background on mixtures see Titterington, Smith and Makov (1985) and McLachlan and Basford (1988). The latter discusses cluster modeling in detail, as does Titterington (1984), and Sections 4.3.4, 4.4.3 and 5.3 of Titterington, Smith and Makov (1985). The problem of deciding what m should be is discussed in Section 5.4 of Titterington, Smith and Makov (1985) and in Titterington (1990). An interesting recent approach is discussed by Lindsay and Roeder (1992).

The important interface question here is to what extent the statistical approaches are relevant or computationally feasible for applications dealt with within the neural-network literature. One possibly useful tool is the Gaussian sums idea of Sorenson and Alspach (1971), in which a probability density function is estimated by an equally-weighted mixture of m p-variate Gaussian distributions. Provided m increases appropriately with N, the Gaussian sum provides an arbitrarily close estimate of the underlying density of the training data which is assumed to be a random sample. The method is essentially a radial basis function method, intermediate between a standard Gaussian mixture and a kernel-based density estimate using a Gaussian kernel. In practice, all these multivariate methods are prone to the curse of dimensionality alluded to in Section 4.3.4.

The delta rules (5) and (24) are reminiscent of recursive methods in statistics. A simple case is recursive updating of a sample mean. If $\bar{x}_n = n^{-1}\Sigma_{r=1}^{n} x^{(r)}$, then

$$(26) \quad \Delta\bar{x}_n \equiv \bar{x}_{n+1} - \bar{x}_n = (n+1)^{-1}(x^{(n+1)} - \bar{x}_n),$$

which is like (24) but with $\eta = \eta(n) = (n+1)^{-1}$. The recursion (26) is a simple stochastic approximation (Robbins and Monro, 1951; Fabian, 1968), and stochastic approximation theory is of value in investigating delta-type learning rules (White, 1992). Also of interest is the modification of (24) and (25) to versions that are not *decision-directed*. Rule (24) is decision-directed in that it assigns $x^{(r)}$ in an all-or-nothing way to one cluster. Alternatively, $x^{(r)}$ might be allocated partially, according to a randomized rule, to each cluster. This is similar to the process known as *learning with a probability teacher* and is related to the *softmax* procedure of Bridle (1990). For various recursive methods of this type, see Chapter 6 of Titterington, Smith and Makov (1985).

The history of identifying data with cluster centers under the nomenclature of *vector quantization* is a long one, and its importance to communication theory was recognized in the March 1982 Special Issue of the IEEE Transactions on Information Theory (Volume 28, pp. 127–202). In that context, the problem was that of "the mapping of vectors from an analog information source into a *finite* (our italics) collection of words for transmission over a digital channel...." (Gray, 1982). The justification of the terminology "vector quantization" is that a data-vector x is reduced or quantized to the indicator of the closest cluster center. The essential features of the k-means algorithm of MacQueen (1967) emanated from Lloyd (1957), reprinted in the Special Issue. Optimal quantization (i.e., optimal choice of cluster centers) is discussed by Gersho (1982), and earlier by Linde, Buzo and Gray (1980), in the form of the eponymous LBG algorithm; also see Luttrell (1990, 1991, 1992). Asymptotic results for the k-means method (Hartigan, 1978; Pollard, 1981) are extended by Pollard (1982a, b), and Kieffer (1982) investigates the rate of convergence of the empirical quantizers. A glance at current literature confirms quantization remains an important topic in information theory. See Gray (1990).

7. THE FUTURE FOR THE INTERFACE BETWEEN ANN MODELING AND STATISTICAL METHODOLOGY

Statisticians must continue to undertake critical comparisons in common areas such as discriminant analysis (pattern recognition) and cluster analysis (associative memories). We have shown that some statistical procedures, including regression, principal component analysis, density estimation and statistical image analysis, can be given a neural network expression. In addition, we have shown that there is scope for general statistical modeling in neural network contexts, and we have remarked that familiar criteria for model-choice can be and indeed have been applied to neural network models. Some of this methodology involves modern Monte Carlo approaches to inference. Applied probabilists may find of interest the structures of the stochastic Hopfield networks and associated developments (cf. Whittle, 1991).

In data analysis, a variety of interesting questions are suggested. Are neural networks that are not model-based useful in everyday contexts? If so, can they cope with complications such as missing data? Can it be established that a general statistical approach, such as projection pursuit regression or the flexible discriminant analysis of Hastie, Buja and Tibshirani (1992), will always be found that will work at least as well as an idiosyncratic network designed for a very specific application? (If not, then why not or when not?) Does the error-backpropagation learning rule, slow even with acceleration-motivated modifications, still have a place? Are systematic procedures for model choice useful? How best can regularization techniques be used to avoid overfitting? How far can theoretical work take us in assessing generalization ability? An increasingly common criticism of neural network methods is that they may provide good predictors but are difficult to interpret. How important is interpretability in particular applications?

There is an increasing emphasis on probabilistic and statistical ideas in the current neural-network literature. Some wheels are being reinvented and some tools are being reapplied in new areas. It is important for statisticians to be aware of this whole field and to be able to contribute in a critical but not destructive way. They should be prepared to discover some new ideas and, undoubtedly, new classes of large-scale challenging problems.

Kanal's (1993) personal view of the current status of pattern recognition contains much food for thought. He retraces the downs and ups of ANN research, remarking on its successes but noting that comparatively simple statistical procedures often perform as well or better. He supports the idea of hybrid networks to deal with complex problems or even the fusion of methods from several different approaches. He alludes to a hybrid network for classifying radar cross sections that consist of a lower layer of 17 triples each containing a linear vector quantizer, a back-propagation network (MLP) and an ART network. Each triple feeds upwards into one of 17 further back-propagation networks, the outputs from which feed into a final MAXNET that identifies the predicted pattern. It is important to evaluate when and to what degree such an intricate network offers superior performance to approaches from the standard statistical repertoire.

ACKNOWLEDGMENTS

B. Cheng wishes to thank the U.K. Science and Engineering Research Council for support provided by its Complex Stochastic Systems Initiative. We are extremely grateful to many people for making copies of their recent work available and to several individuals, in particular the Executive Editor, Editor and referees who reviewed the paper, for many constructive suggestions.

REFERENCES

AARTS, E. H. L. and KORST, J. H. M. (1989). *Simulated Annealing and Boltzmann Machines*. Wiley, New York.

ACKLEY, D. H., HINTON, G. E. and SEJNOWSKI, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science* 9 147–169.

AKAIKE, H. (1974). Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory* (B. N. Petrov and F. Csaki, eds.) 261–281. Akademia Kiedo, Budapest.

ALEKSANDER, I., ed. (1989). *Neural Computing Architectures: The Design of Brain-Like Machines*. North Oxford, London.

AMARI, S. I. (1990). Mathematical foundations of neurocomputing. *Proc. IEEE* 78 1443–1463.

AMARI, S. I., KURATO, K. and NAGAOKA, W. (1992). Information geometry of Boltzmann machines. *IEEE Trans. Neural Networks* 3 260–271.

AMIT, D. (1989). *Modelling Brain Function.* Cambridge Univ. Press.

AMIT, D. J., GUTFREUND, H. and SOMPOLINSKY, H. (1985a). Spin-glass models of neural networks. *Phys. Rev. A* 32 1007–1018.

AMIT, D. J., GUTFREUND, H. and SOMPOLINSKY, H. (1985b). Storing infinite numbers of patterns in a spin-glass model of neural networks. *Phys. Rev. Lett.* 55 1530–1533.

ANDERSON, J. A. and ROSENFELD, E., eds. (1988). *Neurocomputing: Foundations of Research.* MIT Press.

ANDERSON, N. H. and TITTERINGTON, D. M. (1993). Beyond the binary Boltzmann machine. Preprint.

ANTOGNETTI, P. and MILUTINOVIC, V. (1991). *Neural Networks: Concepts, Applications and Implementations. Vol. I.* Prentice-Hall, Englewood Cliffs, NJ.

Barron, A. R. (1989). Statistical properties of artificial neural networks. *Proceedings of the 28th IEEE International Conference on Decision and Control* 1 280–285. IEEE, New York.

Barron, A. R. (1991). Complexity regularization with application to artificial neural networks. In *Nonparametric Function Estimation and Related Topics* (G. Roussas, ed.) 561–576. Kluwer, Dordrecht .

Barron, A. R. (1992). Neural net approximation. In *Proc. 7th Yale Workshop on Adaptive and Learning Systems* (K. S. Narendra, ed.) 69–72. Yale Univ., New Haven.

Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inform. Theory* 39 930–945.

Barron, A. R. (1994). Approximation and estimation bounds for artificial neural networks. *Machine Learning* 14 115–133.

Barron, A. R. and Barron, R. L (1988). Statistical learning networks: a unifying view. In *Computing Science and Statistics: Proceedings of the 10th Symposium on the Interface* (E. G. Wegman, ed.) 192–203. Amer. Statist. Assoc., Washington, DC.

Barron, A. R. and Xiao, X. (1991). Comment on "Multivariate adaptive regression splines," by J. H. Friedman. *Ann. Statist.* 19 67–82.

Bas, C. F. and Marks, R. J. (1991). Layered perceptron versus Neyman-Pearson optimal detection. In *Proceedings of the 1991 IEEE Conference on Neural Networks* 1486–1489. IEEE, New York.

Baum, E. B. and Haussler, D. (1989). What size net gives valid generalization? *Neural Computation* 1 151–160.

Bengio, Y., DeMori, R., Flammia, G. and Kompe, R. (1992). Global optimization of a neural network—hidden Markov model hybrid. *IEEE Trans. Neural Networks* 3 252–258.

Besag, J. and Green, P. J. (1993). Spatial statistics and Bayesian computation. *J. Roy. Statist. Soc. Ser. B* 55 25–37.

Bienenstock, E. L. and Von der Malsburg, C. (1987). A neural network for invariant pattern recognition. *Europhysics Letters* 4 121–126.

Bishop, Y. M. M., Fienberg, S. E. and Holland, P.W. (1975). *Discrete Multivariate Analysis.* MIT Press.

Bornholdt, S. and Graudenz, D (1992). General asymmetric neural networks and structure design by genetic algorithms. *Neural Networks* 5 327–334.

Bourlard, H. E. (1990). How connectionist models could improve Markov models for speech recognition. In *Advanced Neural Computers* (R. E. Eckmiller, ed.) 247–254. North-Holland, Amsterdam.

Bourlard, H. E. and Morgan, N. (1991). Merging multilayer perceptron and hidden Markov models: some experiments in continuous speech recognition. In *Neural Networks: Advances and Applications* (E. Gelenbe, ed.) 215–239. North-Holland, Amsterdam.

Breiman, L. (1993). Hinging hyperplanes for regression classification and function approximation. *IEEE Trans. Inform. Theory* 39 999–1013.

Breiman, L. Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984). *Classification and Regression Trees.* Wadsworth, Belmont, CA.

Breiman, L. and Ihaka, R. (1984). Nonlinear discriminant analysis *via* ACE and scaling. Tech. Report 40, Dept. Statistics, Univ. California, Berkeley.

Bresshoff, P. C. and Taylor, J. G. (1990). Random iterative networks. *Phys. Rev. A* 41 1126–1137.

Bridle, J. S. (1990). Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neuro-computing: Algorithms, Architectures and Applications* (F. Fougleman-Soulie and J. Herault, eds.) 227–236. Springer, New York.

Bridle, J. S. (1992). Neural networks or hidden Markov models for automatic speech recognition: is there a choice? In *Speech Recognition and Understanding: Recent Advances, Trends and Application* (P. LaFace, ed.) 225–236. Springer, New York.

Broomhead, D. S. and Lowe, D. (1988). Multivariate functional interpolation and adaptive networks. *Complex Systems* 2 321–355.

Bryson, A. E. and Ho, Y. C. (1969). *Applied Optimal Control.* Blaisdell, New York.

Buntine, W. L. and Weigend, A. S. (1991). Bayesian backpropagation. *Complex Systems* 5 603–643.

Byrne, W. (1992) Alternating minimization and Boltzmann machine learning. *IEEE Trans. Neural Networks* 3 612–620.

Campbell, C., Sherrington, D. and Wong, K. Y. M. (1989). Statistical mechanics and neural networks. In *Neural Computing Architectures: The Design of Brain-Like Machines* (I. Aleksander, ed.) 239–257. North Oxford, London.

Carpenter, G. A. and Grossberg, S. (1988). The ART of adaptive pattern recognition by a self-organizing neural network. *Computer* 21 77–88.

Casdagli, M. (1989). Nonlinear prediction of chaotic time series. *Phys. D* 35 335–356.

Cheng, B. and Titterington, D. M. (1994). A small selection of neural network methods and their statistical connections. In *Statistics and Images II* (K. V. Mardia, ed.) Carfax, Abingdon. To appear.

Cox, D. R. and Snell, E. J. (1989). *Analysis of Binary Data.* Chapman and Hall, London.

Craven, P. and Wahba, G. (1979). Smoothing noisy data with spline functions. *Numer. Math.* 31 377–403.

Csiszar, I. and Tusnady, G. (1984). Information geometry and alternating minimization procedures. In *Statist. Decisions* Suppl. 1 205–237. Oldenbourg, Munich.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Math. Control Signals System* 2 303–314.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data *via* the EM algorithm (with discussion). *J. Roy. Statist. Soc. Ser. B* 39 1–38.

Duane, S., Kennedy, A. D., Pendleton, B. J. and Roweth, D. (1987). Hybrid Monte Carlo. *Phys. Lett. B* 195 216–222.

Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis.* Wiley, New York

Eckmiller, R. E. (ed.) (1990). *Advanced Neural Computers.* North-Holland, Amsterdam.

Eckmiller, R. E., Hartmann, G. and Hauske, G., eds. (1990). *Parallel Processing in Neural Systems and Computers.* North-Holland, Amsterdam.

Eckmiller, R. E. and Von Der Malsburg, C., eds. (1988). *Neural Computers.* NATO ASI Series. Springer, Berlin.

Efron, B. (1964). The perceptron correction procedure in nonseparable situations. Technical report RADC-TDR-63-533. Rome Air Development Center, Rome, NY.

Fabian, V. (1968). On asymptotic normality in stochastic approximation. *Ann. Math. Statist.* 39 1327–1332.

Farmer, J. D. and Sidorowich, J. J. (1989). Predicting chaotic dynamics. In *Dynamic Patterns in Complex Systems* (Kelso, J. A. S., Mandell, A. J. and Shlesinger, M. F., eds.) 265–292. World Scientific, Singapore.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Ann. Eugenics* 7 179–184.

Friedman, J. H. (1989). Regularized discriminant analysis. *J. Amer. Statist. Assoc.* 84 165–188.

Friedman, J. H. (1991). Multivariate adaptive regression splines (with discussion). *Ann. Statist.* 19 1–141.

Friedman, J. H. and Silverman, B. W. (1989). Flexible parsimonious smoothing and additive modeling (with discussion). *Technometrics* 31 3–39.

Friedman, J. H. and Stuetzle, W. (1981). Projection pursuit regression. *J. Amer. Statist. Assoc.* **76** 817–823.

Fukumi, M., Omatu, S., Takeda, F. and Kosaka, T. (1992). Rotation invariant neural pattern recognition systems with application to coin recognition. *IEEE Trans. Neural Networks* **3** 272–279.

Gawthrop, P. J. and Sbarbaro, D. G. (1990). Stochastic approximation and multilayer perceptrons: the gain backpropagation algorithm. *Complex Systems* **4** 51–74.

Gelenbe, E. (1991a). Theory of the random neural network model. In *Neural Networks: Advances and Applications* 1–20. North-Holland, Amsterdam.

Gelenbe, E., ed. (1991b). *Neural Networks: Advances and Applications.* North-Holland, Amsterdam.

Geman, S., Bienenstock, E. and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation* **4** 1–58.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6** 721–741.

Gersho, A. (1982). On the structure of vector quantizers. *IEEE Trans. Inform. Theory* **28** 157–166.

Girosi, F. and Poggio, T. (1990). Networks and the best approximation property. *Biol. Cybernet.* **63** 169–176.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, Reading, MA.

Gordon, A. (1981). *Classification.* Chapman and Hall, London.

Gray, R. M. (1982). Editorial for special issue. *IEEE Trans. Inform. Theory* **28** 127–128.

Gray, R. M. (1990). *Source Coding Theory.* Kluwer, Boston.

Hand, D. J. (1981). *Discrimination and Classification.* Wiley, New York.

Hartigan, J. A. (1975). *Clustering Algorithms.* Wiley, New York.

Hartigan, J. A. (1978). Asymptotic distributions for clustering criteria. *Ann. Statist.* **6** 117–131.

Hastie, T., Buja, A. and Tibshirani, R. (1992). Penalized discriminant analysis. Preprint.

Hastie, T. and Tibshirani, R. (1990). *Generalized Additive Models.* Chapman and Hall, London.

Hastie, T., Tibshirani, R., and Buja, A. (1992). Flexible discriminant analysis. Preprint.

Hebb, D. O. (1949). *The Organization of Behavior: A Neurophysiological Theory.* Wiley, New York.

Hecht-Nielsen, R. (1990). *Neurocomputing.* Addison-Wesley, Reading, MA.

Hertz, J., Krogh, A. and Palmer, R. G. (1991). *Introduction to the Theory of Neural Computation.* Addison-Wesley, Reading, MA.

Hinton, G. E. (1986). Learning distributed representations of concepts. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society* 1–12. Erlbaum, Hillsdale, NJ.

Hinton, G. E. (1989). Connectionist learning procedures. *Artif. Intell.* **40** 185–234.

Hinton, G. E. (1992). How neural networks learn from experience. *Scientific American* **267** 104–109.

Hinton, G. E. and Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1* (D. E. Rumelhart, G. E. Hinton and R. J. Williams, eds.) 282–317. MIT Press.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci. U.S.A.* **79** 2554–2558.

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Nat. Acad. Sci. U.S.A.* **81** 3088–3092.

Hopfield, J. J. and Tank, D. W. (1985). Neural computation of decisions in optimization problems. *Biol. Cybernet.* **52** 141–152.

Hornik, K., Stinchcombe, M. and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks* **2** 359–366.

Hunt, K. J., Sbarbaro, D., Zbikowski, R. and Gawthrop, P.J. (1992). Neural networks for control systems–a survey. *Automatica* **28** 1083–1112.

Johnson, R. C. and Brown, C. (1988). *Cognizers: Neural Networks and Machines That Think.* Wiley, New York.

Jones, L. K. (1992). A simple lemma on greedy approximation in Hilbert Space and convergence rates for projection pursuit regression and neural network training. *Ann. Statist.* **20** 608–613.

Kanal, L. (1993). On patterns, categories and alternate realities. *Pattern Recognition Letters* **14** 241–255.

Kass, R. E. and Raftery, A. E. (1993). Bayes factors and model uncertainty. Technical Report 571, Dept. Statistics, Carnegie Mellon Univ.

Kieffer, J. C. (1982). Exponential rate of convergence for Lloyd's method 1. *IEEE Trans. Inform. Theory* **28** 205–210.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biol. Cybernet.* **43** 59–69.

Kohonen, T. (1989). *Self-organization and Associative Memory,* 3rd ed. Springer, Berlin.

Kohonen, T. (1990) Internal representations and associative memory. In *Parallel Processing in Neural Systems and Computers* (R. E. Eckmiller, G. Hartman and G. Hauske, eds.) 177–182. North-Holland, Amsterdam.

Kohonen, T., Makisara, K., Simula, O. and Kangas, J., eds. (1991). *Artificial Neural Networks* **1, 2**. North-Holland, Amsterdam.

Komlos, J. and Paturi, R. (1988). Convergence results in an associative memory model. *Neural Networks* **1** 239–250.

Kressel, U. W-G. (1991). The impact of the learning-set size in handwritten-digit recognition. In *Artificial Neural Networks* (T. Kohonen, K. Makisara, O. Simula and J. Kangas, eds.) **2** 1685–1690. North-Holland, Amsterdam.

Kuhnel, H. and Traven, P. (1991). A network for discriminant analysis. In *Artificial Neural Networks* (T. Kohonen, K. Makisara, O. Simula and J. Kangas, eds.) **2** 1053–1056. North-Holland, Amsterdam.

Lachenbruch, P.A. and Mickey, M.R. (1968). Estimation of error rates in discriminant analysis. *Technometrics* **10** 1–10.

Lapedes, A. and Farber, R. (1987). Nonlinear signal processing using neural networks. Technical Report LA-UR-87-2662, Los Alamos National Laboratory, Los Alamos, NM.

Lauritzen, S. L. and Spiegelhalter, D. J. (1988). Local computations with probabilities in graphical structures and their applications to expert systems (with discussion). *J. Roy. Statist. Soc. Ser. B* **50** 157–224.

Le Cun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. and Jackel, L. D. (1989). Backpropagation applied to handwritten Zip code recognition. *Neural Computation* **1** 541–551.

Le Cun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. and Jackel, L.D. (1990). Handwritten digit recognition with a backpropagation network. In *Advances in Neural Information Processing Systems II* (D. S. Touretzky, eds.) 396–404. Morgan Kaufmann, San Mateo, CA.

Levin, E., Tishby, N. and Solla, S. A. (1990). A statistical approach to learning and generalization in layered neural networks. *Proc. IEEE* **78** 1568–1574.

Lim, G. S., Alder, M. and Hadingham, P. (1992). Adaptive quadratic neural nets. *Pattern Recognition Letters* **13** 325–329.

Li, K.-C. (1987). Asymptotic optimality for C_p, C_L, cross-validation and generalized cross-validation: Discrete index set. *Ann. Statist.* **15** 958–975.

Linde, Y., Buzo, A. and Gray, R. M. (1980). An algorithm for vector quantizer design. *IEEE Trans. Communications Technology* **28** 84–95.

Lindsay, B. G. and Roeder, K. (1992). Residual diagnostics for mixture models. *J. Amer. Statist. Assoc.* **87** 785–794.

Lippmann, R. P. (1987). An introduction to computing with neural nets. *IEEE Acoustics, Speech and Signal Processing Magazine* 4(April) 4–22.

Little, W. A. (1974). The existence of persistent states in the brain. *Math. Biosci.* **19** 101–120.

Lloyd, S. P. (1957). Least squares quantization in PCM. Bell Labs memorandum. Reprinted (1982) in *IEEE Trans. Inform. Theory* **28** 84–95.

Lorentz, G. (1966). *Approximation of Functions*. Holt, Rinehart and Winston, New York.

Lowe, D. (1991). On the statistical inversion of RBF networks: a statistical interpretation. In *Proceedings of the Second IEE International Conference on Artificial Neural Networks*. IEE, London.

Lowe, D. and Webb, A. R. (1990). Exploiting prior knowledge in network optimization: an illustration from medical prognosis. *Network* **1** 299–323.

Lowe, D. and Webb, A. R. (1991). Optimal feature extraction and the Bayes decision in feed-forward classifer networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13** 355–364.

Luttrell, S. P. (1990). Derivation of a class of training algorithms. IEEE Trans. Neural Networks **1** 229–232.

Luttrell, S. P. (1991). Code vector density in topographic mappings: scalar case. *IEEE Trans. Neural Networks* **2** 427–436.

Luttrell, S. P. (1992). Self-supervized adaptive networks. *IEE Proceedings F—Radar and Signal Processing* **139** 371–377.

MacKay, D. J. C. (1992a). Bayesian interpolation. *Neural Computation* **4** 415–447.

MacKay, D. J. C. (1992b). A practical Bayesian framework for backpropagation networks. *Neural Computation* **4** 448–472.

McLachlan, G. J. (1992). *Discriminant Analysis and Statistical Pattern Recognition*. Wiley, New York.

McLachlan, G. J. and Basford, K. E. (1988). *Mixture Models: Inference and Applications to Clustering*. Dekker, New York.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proc. Fifth Berkeley Symp. Math. Statist. Probab.* 281–297. Univ. California Press, Berkeley.

Mallows, C. L. (1973). Some comments on C_p. *Technometrics* **15** 661–675.

Mardia, K. V., Kent, J. T. and Bibby, J. M. (1979). *Multivariate Analysis*. Academic, New York.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* **5** 115–133.

McEliece, R. J., Posner, E. C., Rodemich, E. R., and Venkatesh, S. S. (1987). The capacity of the Hopfield associative memory. *IEEE Trans. Inform. Theory* **33** 461–482.

Minsky, M. L. and Papert, S. A. (1969). *Perceptrons*. MIT Press.

Minsky, M. L. and Papert, S. A. (1988). *Perceptrons*, 2nd ed. MIT Press.

Moody, J. E. (1992). The effective number of parameters: an analysis of generalization and regularization in nonlinear learning systems. In *Advances in Neural Information Processing Systems* 4 (J. E.Moody, S. J. Hanson and R. P. Lippmann, eds.) 847–854. Morgan Kaufmann, San Mateo, CA.

Moody, J. E. and Darken, C. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation* **1** 281–294.

Mpitsos, G. J. and Burton, R. M. (1992). Convergence and divergence in neural networks: processing of chaos and biological analogy. *Neural Networks* **5** 605–625.

Muller, B. and Reinhardt, J. (1990). *Neural Networks: An Introduction*. Springer, Berlin.

Neal, R. M. (1992a). Bayesian training of backpropagation networks by the hybrid Monte Carlo method. Preprint.

Neal, R. M. (1992b). Connectionist learning of belief networks. *Artif. Intell.* **56** 71–113.

Neal, R. M. (1993). Bayesian learning via stochastic dynamics. In *Advances in Neural Information Processing Systems* **5** (C. L. Giles, S. J. Hanson and J. D. Cowan, eds.) 475–482. Morgan Kaufmann, San Mateo, CA.

Newman, C. M. (1988). Memory capacity in neural network models: rigorous lower bounds. *Neural Networks* **1** 223–238.

Nychka, D., Ellner, S., Gallant, A. R. and McCaffrey, D. (1992). Finding chaos in noisy systems. *J. Roy. Statist. Soc. Ser. B* **54** 399–426.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *J. Math. Biol.* **15** 267–273.

Panel on Discriminant Analysis, Classification and Clustering (1989). Discriminant Analysis and Clustering. *Statist. Sci.* **4** 34–69.

Parker, D. B. (1985). Learning logic. Technical Report 47, Center for Computational Research in Economics and Management Science, MIT.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA.

Perantonis, S. and Lisboa, P. J. G. (1992). Translation, rotation and scale invariant pattern recognition by high-order neural networks and moment classifiers. *IEEE Trans. Neural Networks* **3** 241–251.

Poggio, T. (1990). A parallel vision machine that learns. In *Models of Brain Functions* (R.M.J. Cotterill, ed.) 3–34, Cambridge Univ. Press.

Poggio, T. and Girosi, F. (1990). Networks for approximation and learning. *Proc. IEEE* **78** 1481–1497.

Pollard, D. (1981). Strong consistency of K-means clustering. *Ann. Statist.* **9** 135–140.

Pollard, D. (1982a). Quantization and the method of K-means. *IEEE Trans. Inform. Theory* **28** 199–205.

Pollard, D. (1982b). A central limit theorem for K-means clustering. *Ann. Statist.* **10** 919–205.

Quinlan, J. R. (1983). Learning efficient classification procedures and their application to chess end-games. In *Machine Learning* (R.S. Michalski, J. G. Carbonelli and T. M. Mitchell, eds.) 463–482. Tioga, Palo Alto, CA.

Richard, M. D. and Lippmann, R. P. (1992). Neural network classifiers estimate Bayesian a posteriori probabilities. *Neural Computation* **3** 461–483.

Ripley, B. D. (1993a). Statistical aspects of neural networks. In *Networks and Chaos – Statistical and Probabilistic Aspects* (O. E. Barndorff-Nielsen, J. L. Jensen and W. S. Kendall, eds.) 40–123. Chapman and Hall, London.

Ripley, B. D. (1994a). Neural networks and related methods for classification (with discussion). *J. Roy. Statist. Soc. Ser. B* **56**. To appear.

Rissanen, J. (1987). Stochastic complexity. *J. Roy. Statist. Soc. Ser. B* **49** 223–239.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *Ann. Math. Statist.* **22** 400–407.

Ritter, H. and Schulten, K. (1986). On the stationary state of Kohonen's self-organizing sensory mapping. *Biol. Cybernet.* **54** 99–106.

Rosenblatt, F. (1962). *Principles of Neurodynamics*. Spartan, New York.

Rujan, P. (1991). A fast method for calculating the perceptron with maximal stability. Preprint, Univ. Oldenburg, Germany.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986a). Learning internal representation by back-propagating errors. *Nature* **323** 533–536.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986b). Learning internal representation by back-propagating errors. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (D. E. Rumelhart, J. L. McClelland and the PDP Research Group, eds.). MIT Press.

Rumelhart, D. E., McClelland, J. L. and the PDP Research Group (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* MIT Press.

Rumelhart, D. E. and Zipser, D. (1985). Feature discovery by competitive learning. *Cognitive Science* **9** 75–112.

Sanger, T. D. (1989). Optimal unsupervized learning in a single-layer linear feedforward neural network. *Neural Networks* **2** 459–473.

Schwartz, G. (1978). Estimating the dimension of a model. *Ann. Statist.* **6** 461–464.

Seber, G. A. F. and Wild, C. J. (1989). *Nonlinear Regression.* Wiley, New York.

Sejnowski, T. J. and Rosenberg, C. R. (1987). Parallel networks that learn to pronounce English text. *Complex Systems* **1** 145–168.

Sethi, I. K. and Jain, A. K., eds. (1991). *Artificial Neural Networks and Statistical Pattern Recognition.* North-Holland, Amsterdam.

Shibata, R. (1981). An optimal selection of regression variables. *Biometrika* **68** 45–54.

Silverman, B. W. (1985). Some aspects of the spline smoothing approach to nonparametric regression curve estimation (with discussion). *J. Roy. Statist. Soc. Ser. B* **47** 1–52.

Silverman, B. W. (1986) *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, London.

Smith, A. F. M. and Spiegelhalter, D. J. (1980). Bayes factors and choice criteria for linear models. *J. Roy. Statist. Soc. Ser. B* **42** 213–220.

Smith, A. F. M. and Roberts, G. O. (1993). Bayesian computation by the Gibbs sampler and related Markov chain Monte Carlo methods. *J. Roy. Statist. Soc. Ser. B* **55** 3–23.

Smolensky, P. (1986). Information processing in dynamical systems: foundations of harmony theory. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (D. E. Rumelhart, J. L. McClelland and the PDP Research Group, eds.) 194–281. MIT Press.

Sorenson, H. W. and Alspach, D. L. (1971). Recursive Bayesian estimation using Gaussian sums. *Automatica* **7** 465–479.

Specht, D. F. (1990). Probabilistic neural networks. *Neural Networks* **3** 109–118.

Specht, D. F. (1991). A general regression neural network. *IEEE Trans. Neural Networks* **2** 568–576.

Spiegelhalter, D. J. and Lauritzen, S. L. (1990). Sequential updating of conditional probabilities on directed graphical structures. *Networks* **20** 579–605.

Stokbro, K., Umberger, D. K. and Hertz, J. A. (1990). Exploiting neurons with localized receptive fields to learn chaos. *Complex Systems* **4** 603–622.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions (with discussion). *J. Roy. Statist. Soc. Ser. B* **36** 111–147.

Tibshirani, R. (1992). Slide functions for projection pursuit regression and neural networks. Preprint, Univ. Toronto.

Titterington, D. M. (1984). Comments on "Application of the conditional population-mixture model to image segmentation" by S. C. Sclove. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6** 656–658.

Titterington, D. M. (1985). Common structure of smoothing techniques in statistics. *Internat. Statist. Rev.* **53** 141–170.

Titterington, D. M. (1990). Some recent research in the analysis of mixture distributions. *Statistics* **21** 619–641.

Titterington, D. M., Murray, G. D., Murray, L. S., Spiegelhalter, D. J., Skene, A. M., Habbema, J. D. F. and Gelpke, G. J. (1981). Comparison of discrimination techniques applied to a complex data set of head injured patients (with discussion). *J. Roy. Statist. Soc. Ser. A* **144** 145–175.

Titterington, D. M., Smith, A. F. M. and Makov, U. E. (1985). *Statistical Analysis of Finite Mixture Distributions.* Wiley, New York.

Tråvén, H. G. C. (1991). A neural network approach to statistical pattern classification by "semiparametric" estimation of probability density functions. *IEEE Trans. Neural Networks* **2** 366–377.

Van Ryzin, J. (1977). *Classification and Clustering.* Academic, New York.

Vapnik, V. N. (1982). *Estimation of Dependencies Based on Empirical Data.* Springer, New York.

Vijaya Kumar, B. V. K. and Wong, P. H. (1991). Optical associative memories. In *Artificial Neural Networks and Statistical Pattern Recognition* (I. K. Sethi and A. K. Jain, eds.) 219–241. North-Holland, Amsterdam.

Wahba, G. (1990). *Spline Models for Observational Data.* SIAM, Philadelphia.

Waltz, D. and Feldman, J. A. (1988). *Connectionist Models and their Applications.* Ablex, Norwood, NJ.

Webb, A. R. Lowe, D. and Bedworth, M. D. (1988). A comparison of nonlinear optimisation strategies for feed-forward adaptive layered networks. Memorandum 4157, RSRE, Great Malvern, UK.

Wendemuth, A. (1993). Learning optimal threshold and weights for the perceptron of maximal stability. Preprint, Dept. Theoretical Physics, Oxford Univ.

Werbos, P. J. (1974). Beyond Regression: new tools for prediction and analysis in the behavioral sciences. Ph.D. Thesis, Harvard Univ.

White, H. (1989). Some asymptotic results for learning in single hidden layer feedforward networks. *J. Amer. Statist. Assoc.* **84** 1008–1013.

White, H. (1990) Connectionist nonparametric regression: multilayer feedforward networks can learn arbitrary mappings. *Neural Networks* **3** 535–549.

White, H. (1992). *Artificial Neural Networks: Approximation and Learning Theory.* Blackwell, Oxford.

Whittaker, J. (1990). *Graphical Models in Applied Multivariate Statistics.* Wiley, New York.

Whittle, P. (1991). Neural nets and implicit inference. *Ann. Appl. Probab.* **1** 173–188.

Widrow, B. and Hoff, M. E. (1960). Adaptive switching circuits. In *1960 IRE Western Electric Show and Convention Record* 96–104. IRE, New York.

Willshaw, D. J. and Von Der Malsburg, C. (1976). How patterned neural connections can be set up by self-organization. *Proc. Roy. Soc. London Ser. B* **194** 431–445.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks* **5** 241–259.

Zhang, D. and Benveniste, A. (1992). Wavelet networks. *IEEE Trans. Neural Networks* **3** 889–898.

Comment

S. Amari

First of all, I would like to thank the Editor for giving me an opportunity to present my personal view on this interesting paper connecting the interdisciplinary field of neural networks and statistics. I also congratulate the authors for their excellent job of reviewing this difficult field in a very compact and comprehensive way.

The brain is an enormously complex system in which distributed information is processed in parallel by mutual dynamical interactions of neurons. It is still difficult, and challenging, to understand the mechanisms of the brain. Recently, the importance and effectiveness of brain-style computation has been widely recognized by the name of neural networks. Roughly speaking, there are three different research areas concerning neural networks. One is the experimental area based on physiology and molecular-biology, which is progressing rapidly and steadily. The second area is engineering applications of neural networks inspired by the brain-style computation where information is distributed as analog pattern signals, parallel computations are dominant and learning guarantees flexibility and robustness of computation. This area has opened new practical methods of pattern recognition, control systems, time-series analysis, optimization, memories, etc. The third area is concerned with theoretical (or mathematical) foundations of neurocomputing, which search for the fundamental principles of parallel distributed information systems with learning capabilities. From this standpoint, the actual brain is a biological realization of these principles through a long history of evolution.

Statistics has a close relation with the second applications area of neural networks, as the present authors have so clearly shown (also see Ripley, 1993a). Statistical methodology is indeed a very important tool for analyzing neural networks. On the other hand, neural networks provides statistics with tractable multivariate nonlinear models to be studied further. It also inspires statistical sciences with the notions of learning, self-organization, dynamics, field theory, etc. which statistics has so far paid little attention to. On the other hand, statistical sciences provides one of the crucial methods for constructing theoretical foundations of neurocomputing (e.g., Amari, 1990, 1993a). Without these foundations, it is difficult for neural network technology to take off from the present rather "easy and shallow" technology to a more fundamental one.

Artificial neural networks research has experienced ups and downs; up in the early sixties where the perceptron and the adaline were proposed and again a big up in the middle of the eighties until now. It is said that the dark period was around the seventies where little attention had been paid to ANN and that the Minsky-Papert critique gave rise to this down. However, I believe this prevailing story is merely a myth. We can point out the lack of supporting technology as the background of this fall. Computer technology had developed greatly through the sixties and seventies. Researchers on pattern recognition and artificial intelligence thought that it was easier and more powerful to use symbol processing in modern computers rather than to use neural networks technology. This was true, and information processing technology including artificial intelligence had been constructed successfully upon modern computers. However, hardware technology had further developed in the eighties such that it could support neural parallel computation. It was not a dream to construct neurochips or even neurocomputers. There are, of course, many other intellectual reasons to support the resuscitation in the eighties.

In the seventies, most researchers did not think that engineering applications of neural networks were realizable. The background technology was not yet matured at the time. However, it was not a dark period in theoretical study because many of the ideas were proposed in the "dark period" that were rediscovered or developed further to be the fundamental methods supporting the neural network methods today.

For example, the generalized delta rule for a multilayer perceptron was proposed in 1967 (Amari, 1967) where analog neurons were used and the stochastic descent algorithm was applied. The idea was also introduced in a Russian book (Tsypkin, 1973). I believe that there were not a few researchers who knew the idea at that time. It was

S. Amari is Professor, Department of Mathematical Engineering, University of Tokyo, Bunkyo-ku, Tokyo 113, Japan.

the great achievement of Rumelhart, Hinton and Williams (1986b) who not only rediscovered the old idea but have shown its effectiveness in practical problems.

The idea of associative memory of the Hopfield type was intensively studied in 1972 by Kohonen (1972), Nakano (1972) and Anderson (1972). Amari (1972) studied its dynamical characteristics, including both the symmetric connections case where memorized patterns are fixed points of dynamics and the asymmetric connections case where sequences of patterns are memorized and recalled. Hopfield introduced the new notion of the "energy" or Lyapunov function to analyze the associative memory model and opened the new approach of spin-glass analogy to this field. A lot of fundamental studies appeared by using the statistical-physical (spin-glass) method, although the statistical-mechanical theory of neural networks itself appeared in the seventies (Little, 1974; Amari, Yoshida and Kanatani, 1977) the latter of which treated more general non-equilibrium dynamics.

A fundamental idea of self-organizing neural networks was proposed by Von der Malsburg (1973). It was applied to the formation of neural topological maps (Willshaw and Von der Malsburg, 1976). The dynamical instability of such neural field dynamics was studied (Takeuchi and Amari, 1979), which guarantees the formation of patch structure and columns existing in the brain. Based on these works, Kohonen (1982) proposed an excellent idea of learning vector quantization (LVQ) and neural topological maps which are much more simple and efficient compared with the previous models. The possibility of neural principal component analyzer was also pointed out in the seventies (Amari, 1977). Grossberg's adaptive resonance theory (ATR) was proposed in 1976 (Grossberg, 1976).

The achievements in the seventies should not be too exaggerated. Not only old ideas were developed to be applied to practical problems, but a lot of new ideas emerged in the eighties. I would like to emphasize that we need much more fundamental new ideas and mathematical foundations in order to elucidate principles of neurocomputing. Statistical and probabilistic methods are very important for this purpose. The current applications have proved the usefulness of neurocomputing but are still superficial even though they have provided a strong impact on various fields of science and technology with novel nonlinear modeling.

Here, I would like to point out two more interesting topics related to statistics. One is the learning curve that shows how fast a learning machine can improve its behavior as the number of training examples increases. This problem is closely related to the asymptotic theory of statistical inference, but the behavior of a network is measured by the generalization error, not by the squared error of estimated parameters. The estimate of the generalization error can be applied to the model selection problem in which the statistical methods such as Akaike information criterion (AIC) and minimum description length (MDL) are useful. There are a number of approaches to this problem, for example, the computational learning theory approach (Baum and Hausler, 1989), statistical-mechanical approach (Levin, Tishby and Solla, 1990), information-theoretic one and statistical approach (Amari and Murata, 1993; Amari, 1993b). When a network behaves stochastically, the statistical asymptotic theory can easily be applied to this problem. However, when the underlying model is deterministic (or the 0 temperature case in physicists' terminology), the underlying model becomes nonregular in the sense that the Fisher information becomes infinitely large. Therefore, the regular statistical theory cannot be applied. However, we can still construct a universal theory (Amari, 1993b). This is one interesting fact about neural networks.

Another interesting topic concerns the expectation and maximization (EM) algorithm and information geometry. The EM algorithm is the technique of estimation when only partial data are observed. When a neural network includes hidden neurons, only input and output signals are observable as learning data and desired signals on the hidden neurons should be generated or estimated by some means. The EM algorithm is used in learning of hidden units of the Boltzmann machine (Amari, Kurato and Nagoako, 1992; Byrne, 1992). It is interesting that the procedures of the EM algorithm correspond to the e-geodestic projection and m-geodesic projection in the manifold of probability distributions, in the sense of differential geometry of statistical inference (Amari, 1985).

Recently, Jordan and Jacobs (1993) proposed a model called the mixture of expert networks in which one of the component networks is responsible for its own specific tasks. This enables parallel and distributed sharing of tasks. The missing or hidden data is which task should be processed by which network. This model is represented by a mixture of exponential families, and the EM algorithm as well as information geometry plays an essential role in such models.

Comment

Andrew R. Barron

Relationships between topics in statistics and artificial neural networks are clarified by Cheng and Titterington. There are fruitful concepts in artificial neural networks that are worthwhile for the statistical community to absorb. These networks provide a rich collection of statistical models, some of which are ripe for both mathematical analysis and practical applications. Many aspects of artificial neural networks are in need of further investigation. Here, I comment on approximation and computation issues and their impact on statistical estimation of functions.

APPROXIMATION

Attention is focussed on the most commonly studied feedforward networks (perceptrons) which have one or two "hidden" layers defined by composition of units of the form $\phi(wx + w_0)$, where ϕ is a hardlimiter or sigmoidal activation function and w_0, w denote the parameters (internal weights) that adjust the orientation, location and scale of the unit functions (Rosenblatt, 1962; Rumelhart, Hinton and Williams, 1986a). In the one hidden layer case, a linear combination of such units is taken with the internal weights adjusted so that the result approximates a target function. These networks may be regarded as an adjustable basis function expansion of ridge form similar to projection pursuit (Friedman and Stuetzle, 1981) and similar to sparse trigonometric series with adjustable frequency vectors. Linear combination of such adjustable basis functions can provide an accurate approximation with far fewer units than by linear combination of any fixed basis functions for certain classes of target functions when the number of input variables is greater than or equal to three (Barron, 1993). A consequence is that more accurate statistical function estimation is possible for such target functions (Barron, 1994).

These conclusions for one hidden layer networks are based, in part, on the following result developed in Jones (1992) and Barron (1993). Suppose a function $f(x)$ is such that $f(x)/V$ is in the closure of the convex hull of the set of units $\{\pm\phi(wx + w_0) : (w_0, w) \in R^{d+1}\}$, where ϕ is bounded by 1 for some positive number V. The closure is in the $L_2(\mu)$ norm, where μ is any given probability measure μ with bounded support on R^d. Then there are M such units with choices of weights depending on f and μ, such that their linear combination $f_M(x)$ (a single hidden layer network) achieves approximation error

$$\|f - f_M\| \le \frac{V}{\sqrt{M}},$$

where the norm is taken in $L_2(\mu)$. The surprising aspect is that the approximation rate as a function of M is independent of the dimension d. A subclass of functions that satisfy the condition are those that possess a bound on the first moment of the Fourier magnitude distribution. (This class includes all smooth positive definite functions and convex combinations of translates of such functions.) In contrast, approximation using any fixed M basis functions cannot achieve approximation error uniformly better than order $1/M^{1/d}$ for the same class of functions f, taking μ to be the uniform distribution on a d-cube (Barron, 1993).

It is of interest to characterize what classes of functions can be more parsimoniously approximated using two rather than one hidden layer in the network. Some functions such as the indicator of a cube or a ball are not accurately approximated by the ridge expansions represented by one-layer networks without resorting to a number of units exponentially large in the dimension. In these cases the network capabilities may be improved by inclusion of a second layer of threshold nonlinearities. Units on the second layer can provide indicators of the level sets of linear combinations of the first layer units. These level sets can be arranged to take arbitrary polygon shapes (Lippman, 1987). The linear combination of the outputs of the second layer then give piecewise constant approximations of a rather general form. One conclusion of the same flavor as above is that if a function f is such that $f(x)/V$ is in the closure of the convex hull of the set of signed indicators of K-sided polygons for some positive V, then there is a two hidden layer network function $f_{K,M}(x)$ with KM units on the first layer and M units on the second

Andrew R. Barron is Professor, Department of Statistics, Yale University, Box 2179, Yale Station, New Haven, Connecticut 06520.

layer such that $||f - f_{K,M}|| \leq V/\sqrt{M}$. It is not clear yet how much more general a class of functions this is than those in the convex hull of signed indicators of half-spaces. Another approach to examining approximation by two hidden layer networks is in Cybenko (1988). He shows that by using sigmoidal activation functions the second layer units can be arranged to implement localized kernel functions that are then linearly combined to provide the function approximation. He shows that the approximation error tends to zero but does not give a bound on the rate. It is not clear that localized basis expansions will be effective in high dimensions. Nevertheless, two hidden layer networks may provide one way to combine the positive benefits of global ridge approximations and local kernel approximations.

ESTIMATION

These multiunit perceptrons are nonlinearly parameterized models incorporated into least squares regression, classification and likelihood maximization. By combining results on network approximation with analysis of statistical risk, it is possible to bound the accuracy of neural network estimators in certain cases.

Frameworks exist for the analysis of the total risk of function estimation using neural networks or other nonlinear models for various choices of loss function. Analogous to the bias-variance decomposition of the mean squared error, the problem decomposes into separate consideration of the approximation error and the additional error due to estimation of the function from a finite sample (see, for instance, Haussler, 1992; Barron, 1991). With squared error loss, the estimation error can be bounded by the ratio of the number of parameters to the sample size times a logarithmic factor. The best rate of convergence for a network estimator occurs when the size of the network (indexed by the number of parameters) is chosen so that the estimation error is of the same order as the approximation error. In particular, the general risk bounds are applied to the case of one hidden layer networks in Barron (1994). There conditions are given such that the risk is bounded by

$$E||f - \hat{f}||^2 \leq O\left(\frac{V^2}{M} + \frac{Md}{N}\log N\right),$$

where M is the number of units, d is the input dimension, N is the sample size and V is as discussed above. This risk bound is of the order $V^2((d/N)\log N)^{1/2}$ with $M \sim (N/(d\log N))^{1/2}$. Thus, a satisfactorily small statistical risk is possible without requiring an exponentially large sample size. The estimator $\hat{f}$ that achieves these bounds is assumed to correspond to a global optimum of the empirical squared error loss, among one hidden layer networks with M units subject to certain constraints on the parameter values. It can be shown, under similar conditions, that the same risk bounds hold for any estimator that achieves an empirical squared error not larger than a prescribed value determined by the bound on the approximation error.

Since, in general, the network approximation error is not known in practice, data-based model selection criteria are useful to select a size of network that achieves approximately the best convergence rate permitted by the class of models. Such risk bounds are available for networks selected by certain complexity based criteria (Barron and Cover, 1991; Barron, 1991). It is an open problem whether risk bounds can be developed for networks selected by other criteria such as Akaike's AIC; such bounds would be analogous to the results available for linear models by Shibata (1981) and Li (1987).

COMPUTATION

In some cases, optimization of the appropriate objective function is proven to provide accurate estimators in the sense of statistical risk, as discussed above. However, there is no known algorithm for network estimation that is proven to produce accurate estimates of functions in a feasible amount of computation time. At the least, we should avoid having an average computation time that is exponential in the input dimension d. Ideally, the computation time should be bounded by small degree polynomial in N and d while achieving a satisfactory statistical risk bound (e.g., a fractional power of d/N) for a sensible class of target functions, where N is the sample size. It is not known whether such a feasible algorithm exists. Because of its potential practical implications, I regard the resolution of problems of this type as the most important task for theoretical research concerning neural networks.

Various algorithms have been suggested or used in practice that may or may not be appropriate for the function estimation task. Here, some of the standard approaches and associated problems are briefly mentioned. Many of the methods involve numerical search for an optimum of an empirical objective function. Unfortunately, this error surface for multiunit perceptrons is extremely multimodal as a function of the parameters (weights).

Gradient search and many of its variants, such as back-propagation, produce a local optimum of dubious scientific merit. The use of multiple starting points may rescue local search strategies, but it should be mathematically determined whether or not the number of restarts needed on the average is exponential in the size of the problem. The objective function may be regularized by the addition of a large enough convex penalty term (weight decay term) to reduce multimodality, but can it be demonstrated whether the function estimates remain statistically accurate in that case? A concern is that if a penalty term is multiplied by a constant large enough to guarantee convexity of the objective function, then the effect of the empirical loss term may be washed out.

Stochastic search strategies such as simulated annealing or guided random search can avoid traps of local optima to converge to a global optimum, but it needs to be proven whether an accurate estimate is reached in feasible time for perceptrons. See Bertsimas and Tsitsiklis (1993) for some of the issues associated with proving a computation rate for simulated annealing. Convergence theory for random search should reveal what advantage, if any, the search strategy has over exponential time algorithms such as exhaustive search over a suitable grid.

Likelihood maximization can be replaced by averaging with respect to a Bayesian posterior distribution using importance sampling or Metropolis algorithms, but it is not proven whether these algorithms will provide suitable solutions in feasible time for highly multimodal surfaces. Indeed, suppose it were not feasible to find points of high likelihood that provide an accurate estimator. It would then be surprising (but not necessarily impossible) for an averaging technique to produce an accurate estimator.

The computational task is simplified by certain estimation strategies that build up a network one unit at a time. At each stage, the parameters of a new unit are to be determined given that the smaller network has been estimated. In some cases, convex objective functions can be defined that are readily optimized at each stage. One such class of network methods use compositions of small polynomial units, each of which is linearly parameterized and optimized by least squares (Farlow, 1984; Barron and Barron, 1988). Another approach involves logistic sigmoidal units optimized by a relative entropy criterion; see below. It needs to be determined under what conditions functions can be accurately approximated by such iteratively constructed networks.

Some progress has been made in the case of a single hidden layer network with a squared error criterion. Optimizing such networks one node at a time provides a lower dimensional multimodal search task while still permitting an accurate approximation (Jones, 1992; Barron, 1993). In particular, suppose a function f is such that $f(x)/V$ is in the closure of the convex hull of the set of functions $\phi(wx + w_0)$ (and for simplicity, assume odd symmetry $\phi(-z) = -\phi(z)$). Let $f_0(x) = 0$ and for $M = 1, 2, \ldots$ iteratively define $f_M(x) = v_1 f_{M-1}(x) + v_2\phi(wx + w_0)$, where the internal weights w_0, w of the Mth unit are found to maximize the inner product of the function $r_{M-1}(x)$ and $\phi(wx+w_0)$, where $r_{M-1}(x) = f(x) - f_{M-1}(x)$ and then the external linear weights v_1, v_2 are optimized by ordinary least squares. Then $\|f - f_M\| \le 2V/\sqrt{M}$ which is the same order bound as stated above for noniterative approximation. Thus, the search has been reduced from $M(d + 2)$ dimensions down to $d + 1$ dimensions, but the objective function still may have multiple modes for each M. It remains to determine whether it is possible to provide approximate solutions to this simpler optimization (perhaps by a stochastic search or multistart algorithm) in a time that is not exponentially large in d.

An interesting approach worthy of further study is to choose w_0, w for unit M to minimize the average binary relative entropy $D(g, \phi) = g \log g/\phi + (1 - g)\log(1 - g)/(1 - \phi)$ between the functions $g(x) = 1/2 + r_{M-1}(x)/2V$ and $\phi(wx + w_0)$, with ϕ chosen to be the logistic sigmoid $\phi(z) = e^z/(1 + e^z)$ and $r_M(x) = f(x) - f_M(x)$. With this choice, the objective function is strictly convex in w_0, w and an approximate minimizer is readily computed for each M by gradient or Gauss-Newton search as in logistic regression. Now $r_M(x) = 0$ is a fixed point of these iterations. It may be possible to prove that $f - f_M$ tends to zero as $M \to \infty$. Does it have the same $1/\sqrt{M}$ approximation rate? The problem of computational feasibility of accurate network estimation would be solved by the positive resolution of this approximation question.

SUMMARY

I concur with the conclusions of Cheng and Titterington that research in statistics and artificial neural networks is mutually beneficial and that increased awareness of work in the respective disciplines should be encouraged. It should be important to each field not only to acknowledge existing work from both fields but also to put it to use to advance the state of the art. Combined use of approximation theory, mathematical statistics and computation theory are essential to the treatment of fundamental problems of function estimation and neural networks.

Comment

Elie Bienenstock and Stuart Geman

According to the authors, this paper has three principal goals: "informs a statistical readership about Artificial Neural Networks (ANNs), points out some of the links with statistical methodology and encourages cross-disciplinary research...." It seems to us that the authors have been spectacularly successful with regards to the first two of these goals, and it is likely that this paper will do much to further stimulate the already active scientific exchange between the statistics and neural modeling communities.

As Cheng and Titterington made clear, neural networks, at least the very popular examples reviewed in their paper, are not really new inasmuch as they represent variations on common statistical themes, especially nonparametric and semiparametric estimation and classification. Furthermore, Cheng and Titterington suggest that the tie to real neurons may be somewhat tenuous (we will amplify on this shortly). Nevertheless, despite this dubious biological connection and strong ties to already well-studied statistical methods, this field has attracted wide attention from within the government (principally the Department of Defense but also other branches including the Department of Commerce) as well as many sectors of industry. It has drawn many top science students at our top schools. In the meantime, many statistics departments complain that it is hard to find first-rate graduate students.

We would like to use this discussion to speculate about the reasons behind the fantastic growth of the neural modeling field, especially in light of the close ties to well-studied areas of statistics which have themselves been received with substantially less enthusiasm. There are many reasons for the remarkable popularity and visibility of neural networks. We will propose a few and suggest that some of them may be based partly on misconceptions.

THE APPEAL OF BRAIN MODELING

The endeavor is nearly irresistible: building models and machines possessing a measure of human intelligence, working through the puzzles of perception and cognition and "explaining" the brain. Indeed, many researchers in the neural modeling community believe that the kinds of networks discussed by Cheng and Titterington are meaningfully connected with biology, providing a starting point from which we can begin to organize and understand the overwhelmingly complex anatomical and physiological data, and from which new kinds of theoretically-directed biological experiments will emerge. Still, most neural modelers would agree that these attempts are nothing more than the crudest of approximations not to be taken seriously as models of real neurons or real neuronal interactions at the level of any important detail. Cheng and Titterington have already remarked that "it is clear that the brain does not learn by the generalized delta rule." It is also clear that there is very little in the way of feedforward networks in the brain (virtually all substantial pathways are reciprocated making it clear that the dynamics is not that of a feedforward network) and that the real equations of synaptic modification are a good deal more complicated than a Hebbian or gradient-descent rule. In short, ANNs are hardly neural.

THE APPEAL OF "GENERALIZATION"

Model-free generalization has served as a kind of Holy Grail in neural modeling: begin with a more-or-less *tabula rasa* (blank slate, or, in statistical parlance, "nonparametric") architecture and a realistically-sized training set for some challenging classification or estimation task and devise a learning rule powerful enough to discover the regularities and invariants that would extrapolate good performance beyond the training data. Such a device might be used to "beat the stock market" or solve the automatic target recognition (ATR) problem which has resisted many years of expensive R&D effort. But statisticians know that generalization (good performance on samples not in the training set) depends almost entirely on the extent to which the training set is representative, and/or the structure of the problem happens to accommodate the models used. It is too much to expect statistical methods to "discover," by themselves, complex and nontrivial structure such as the structure

Elie Bienenstock is Visiting Associate Professor (on leave from CNRS, Paris, France) and Stuart Geman is Professor, Division of Applied Mathematics, Brown University, Box F, Providence, Rhode Island 02912.

that defines classes of objects, invariant to lighting, shading, texturing, rigid and nonrigid shape deformations and viewing perspectives. The situation with pre-segmented hand-written numerals is quite special: this is a small class of essentially one-dimensional structures for which very large and comprehensive training sets are available.

Of course, the problem of recognizing handwritten numerals is an important one, and there are many other problems of equal importance which are equally amenable to neural network and related statistical approaches. However, it has been observed many times that for such problems simple nearest-neighbor methods (or variations on that theme) typically perform nearly as well (and often better) than neural networks [see, for example, the thorough experiments by Ripley (1993)]. Evidently, in these cases "generalization" is mostly a matter of *interpolation.*

We have argued elsewhere (Geman, Bienenstock and Doursat, 1992) that for many of the more ambitious problems for which neural networks have been proposed (such as ATR, unconstrained handwriting recognition or learning complex motor maps for robot arms with multiple degrees of freedom), the choice of a suitable statistical method may ultimately play only a minor role. The more substantial challenge may prove to be the choice of appropriate *representations*, in particular, representations in which generalization can, in fact, be viewed as a matter of interpolating a sufficiently rich but reasonably-sized training set. We would argue, for example, that unconstrained object recognition will require the development of representations that are already nearly invariant to pose, shape, lighting, etc., and that "learning" such representations from examples is nearly impossible with realistic training sets.

Cheng and Titterington remark that two principal steps in treating a practical problem are (i) the specification of an appropriate architecture, and (ii) network training from examples. We would like to suggest that substantial progress on the more ambitious problems for which neural networks have been proposed will require a shift in emphasis from issues of training to issues of architecture—which is to say, modeling.

PROBLEM SELECTION

Cheng and Titterington began their paper with a list of currently used—in some cases about-to-be-used—applications of ANNs. The list is impressive, and one could no doubt add more items to it, such as the various applications to high-energy physics (e.g., see Denby, 1993) to mention but one area. The fact that ANNs have been successfully applied to work with real data for substantial problems in speech synthesis (NETtalk), speech recognition, character recognition and robotics has certainly contributed much to their appeal. However, it should be mentioned that there is the tendency to somewhat exaggerate the successes. After about ten years of intense activity in the field, the number of concrete industrial applications is still rather limited. Many "applications" are really *demonstrations*, and it is often the case that neural nets are outperformed by (less general) *ad hoc* solutions. This, for example, is the situation with NETtalk, as Cheng and Titterington have pointed out.

PACKAGING

The importance of an appealing presentation cannot be ignored, even in science. Cheng and Titterington rightly remark that ANNs are sometimes perceived, from the perspective of statisticians, as "familiar entities" with a representation that is "usually pictorial." Although the last two words appear in parentheses in the paper, they could actually be taken as one of the main take-home messages. What is a radial-basis-function ANN if not a kernel method for regression *with a picture*? Figure 8 is the picture of a two-layer perceptron, but this is nothing more than a particular nonlinear regression model. In fact, wording itself can play a substantial role. Contrast the very intuitive notions used in the definition of Boltzmann machines—hidden units; clamped and unclamped dynamics; Hebbian synaptic plasticity—to the rather unappealing statistical terminology (to quote again from the paper): "a version of the iterative proportional fitting procedure used in analyzing multiway contingency tables." For that matter, also consider the phrase "Boltzmann machine" against "semiparametric estimation via maximum likelihood."

We would like to conclude by observing that, despite these reservations, there is little doubt that the popularity of ANNs has had, and continues to have, a very positive effect on scientific research. It has brought together scientists from diverse disciplines to work on important and interesting problems (numerous prominent theoretical physicists, mathematicians, computer scientists and biologists have adopted the field as a kind of second career), and it has done much to advertise the enormous potential of statistics for addressing a host of modern, "high-technology," problems. Cheng and Titterington's paper should be welcomed as further encouragement to this kind of important cross-disciplinary research.

ACKNOWLEDGMENTS

This work was supported by Army Research Office contract DAAL03-92-G-0115 to the Center for Intelligent Control Systems, NSF Grant DMS-92-17655 and Office of Naval Research Contract N00014-91-J-1021.

Comment

Leo Breiman

Cheng and Titterington have most commendably brought developments in the neural network field to the attention of statisticians. It is a notable public service. Since their title is worded "...A Review from a Statistical Perspective", room is left for other statistical perspectives.

When I first heard about neural networks some years ago, I was put off by what I considered to be the hype about doing things the way the brain does. The going propaganda seemed to be that here was a set of procedures modeled after the brain that did a miraculously accurate job in a wide variety of tasks. The functioning of these procedures was coded in esoteric language based on terms borrowed from brain mechanisms. The whole thing was reminiscent of the artificial intelligence publicity a decade or two ago.

But in going to neural network meetings, reading and refereeing their articles and talking to many practitioners over the last five years, my opinion has changed. The neural network community consists of different segments. Some are concerned with constructing mathematical network models of the brain. Others are concerned with networks as mathematical entities, that is, their connectedness, dynamics, etc. Probably the largest segment consists of the people doing work on pattern recognition and other predictive problems.

1. THE CHARACTERISTICS OF THIS LATTER COMMUNITY

They are *not* a neural network community. They use any methodology that works on their problems. Often, they use CART or MARS. They experiment with nearest neighbor methods, separating surfaces gotten by using linear programming, radial basis functions, hidden Markov chains, etc. New methodologies are constantly proposed, and many of these have little resemblance to standard neural networks. Unfortunately, much of the original, and now anachronistic, terminology is retained giving misleading impressions about what is going on.

They are very pragmatic and problem oriented. In fact, the field is better defined by the nature of the problems they work on then by any particular methodology. Typical problems are speech recognition and handwritten character recognition. The range of problems is characterized by high dimensional complex data, often with very large sample sizes (10^4 to 10^7). The goal is to find accurate predictors in classification, regression and time series.

Often, the methodology they use is hand-tailored to the problem they are working on. In this respect, the neural network technology is attractive in that the network and the number of internal nodes can be tinkered with and optimized for the problem. But other methods are employed if they give better results.

Their bottom line is the error rate on the relevant data set. Proposed new methodologies are judged in terms of their error rates on banks of known data sets. But there is little systematic research into the circumstances under which some methods work better than others. This may be because the work is so oriented toward particular problem solving and tailored methodologies.

The people involved are, by background, computer scientists, engineers and physical scientists. They are generally young, energetic and highly computer literate. They have the further good fortune not to have any formal statistical training so that they feel no compulsion to engage in the futile games of modeling data or in endless asymptotics. What they have borrowed from statistics is very slight.

There are important cultural differences between the statistical and neural network communities. If a statistician analyzes data, the first question he gets asked is "what's your data model?" The NN practitioner will be asked "what's your accuracy?" In

Leo Breiman is Professor, Department of Statistics, 367 Evans Hall, University of California, Berkeley, California 94720.

statistics, high dimensionality (number of parameters estimated) is 5, maybe 20, and 100 is impressive. In NN problems, 100 is moderate while 1000 and 10,000 are more like it. Statisticians go for interactive computing. A NN member might say "what, only an overnight run? It must be a pretty small problem."

Another difference is that statisticians tend to try and develop universal methodology. That is, methodology that can be applied, virtually unchanged, in every environment. For instance, CART has been used, in untinkered form, in dozens of different fields. The NN workers, as mentioned above, tinker and tailor, cut and slice until the suit fits the data.

2. LOOKING AT THE NEURAL NETWORK METHODOLOGY

In the present prediction context, what is given is a set of data consisting of the variables to be used as predictors (usually denoted as a vector x) together with the associated values of the things (responses) to be predicted. This data is known as the training set or as the learning set. The goal is to use this data to construct a predictor of future responses based only on knowing x.

The neural network configuration most often used in prediction is called the single layer feed forward network. This has been covered by Cheng and Titterington, but I want to go through it again for several reasons. First, because it is the type of neural network most widely used in prediction. Second, because its success in some important problems was largely responsible for the surge of interest in these methods. Finally, because its structure is simple, we can hope to get some idea of its workings.

The idea is this: let the sigmoid function $\sigma(x) = \exp(x)/(1+\exp(x))$. Then fit the data by linear combinations of σ (linear combinations of the predictor variables). In regression where the training data is of the form $(y_n, x_n), n = 1, \ldots, N$ and x has M coordinates $x_1, \ldots, x_M$, $x_1 \equiv 1$, fit the data by a sum of the form

$$\hat{y}(x) = \sum_k \alpha_k \sigma(\beta_k x).$$

In a J class problem, the training data is of the form $(j_n, x_n), n = 1, \ldots, N$, where each j_n is a class label taking value in $\{1, \ldots, J\}$. Then the conditional probability for each class is estimated by a function of the form

$$\hat{p}(j \mid x) = \sigma\left(\sum_k \alpha_{jk}\sigma(\beta_k x)\right),$$

and the decision rule is to predict the class corresponding to the vector x as j if

$$\hat{p}(j \mid x) = \max_i \hat{p}(i \mid x).$$

To estimate the coefficients in regression, the least squares error L is defined by

$$L(\alpha, \beta) = \sum_n \left(y_n - \sum_k \alpha_k \sigma(\beta_k x_n)\right)^2.$$

Then L is minimized using gradient descent. In classification, define $z_{jn} = 1$ if $j_n = j$, otherwise zero. Put

$$L(\alpha, \beta) = \sum_{j,n} \left(z_{jn} - \sigma\left(\sum_k \alpha_{jk}\sigma(\beta_k x_n)\right)\right)^2$$

and again minimize L by gradient descent. The gradient descent most commonly used is called backpropagation and consists of putting in one data case at a time and then taking a partial gradient step. The data set is circulated through until the convergence is deemed satisfactory.

This is a simple and easily programmed idea. Since its introduction, it has been used in a wide variety of important engineering and computer applications with almost universally "satisfactory" results. In fact, it has become an all purpose crank. For many hopeful users, it relieves the tedium of thinking.

For instance, consider a problem that consists of classifying 32 × 32 bit images with each pixel in 16 grey levels and such that there are 26 classes. Note that each prediction vector x is of dimension 1024. Before the NN technology, researchers would puzzle over the images and try to extract a few features (functions defined on each image) that would contain most of the relevant classification information. Having drastically reduced the dimensionality, some standard classification methods could be used on the feature values.

Now the procedure is to toss the data directly into the NN software, use tens of thousands of parameters in the fit, let the workstation run 2–3 weeks grinding away doing the gradient descent and, voila, out comes the result. Automatic feature selection has taken place.

There are pluses and minuses to the NN crank. Prior to the crank, the only widely available methods were nearest neighbor templates, linear methods and various kludges. The NN crank is a widely applicable nonlinear method that usually gives good results.

3. BUT ALL IS NOT TEA AND CRUMPETS

The NN crank may not work well without a lot of tuning and tinkering. A number of initial decisions have to be made to run the program. For instance, how many sigmoids to use in the fit? (In their language, how many nodes to use in the hidden layer?) Each additional sigmoid used introduces $M + 1$ additional parameters to estimate. If too many sigmoids are used, there is the possibility of overfitting the data; too few and the data may be underfit.

Another problem is what initial values of the parameters to use. Gradient descent finds a nearby local minimum and the nonlinear surface generated by sums of sigmoids is guaranteed to have many local minima. One way to find the global minimum is to run the procedure many times starting from randomly selected initial values. But the lengthy running times of neural networks rule this out.

In my discussions with many practitioners concerning this problem, I ran into two schools of thought. One was "don't worry, all local mins give about the same accuracy." The second, and more surprising, was "never run for so long that you get into a local minimum."

The latter prescription defies the usual descriptions of how neural networks methodology works. But it seems to be followed by many of the most experienced and successful practitioners. The idea is this: given that you are minimizing over thousands of parameters, if you fall into the bottom of a minimum then you are overfitting the data. The "smart thing" to do is to set aside a test set, stop the program at various times, run the test set down the current predictor and select that point in the run that gives minimum test set error.

There are other recipes for avoiding overfitting. For instance, another current recipe is the use of regularization (aka "weight decay" in NN terms). Here, instead of minimizing the error sum of squares, a penalty term is added consisting of the sums of squares of the coefficients multiplied by a parameter to be determined. This method takes repeated runs, much more computing, and does not seem to have been widely adopted. Still another recipe advanced to me by knowledgeable users is to stop the run at various times and delete "inactive" variables from the fitting procedure.

Experienced users know how to tinker, cut and paste. They have their own ways of adjusting the number of nodes in the hidden layer to get good performance, and of preventing overfitting. But most of this is folk wisdom, and there is, so far, no handbook on the sacred mysteries of neural network tinkering.

There is nothing wrong with tinkering, but not enough is known about how best to tinker. There is not enough known about performance of neural networks on simple simulated data. We need to know more about the whys and wherefores, ifs and buts of NN performance.

4. ALTHOUGH SOME METHODS ARE USUALLY GOOD, NO METHOD IS ALWAYS BEST

Neural networks cannot satisfy the desire for ultimate optimality. It has become increasingly clear to the NN community that no one prediction method will be universally most accurate on all data and that what is best depends on the structure of the data. Because of this, a cottage industry in the invention of new methods has risen.

The methods generally fall into one of two categories. The first I call global. These methods (like neural nets) use the training data to estimate a global prediction surface. Local methods make a local prediction for each new vector x. For instance, in classification, the class predicted for x may be the class of its nearest neighbor in the training data.

To understand what is different and new about neural networks, we give a brief and selective overview of global methods currently used in nonlinear analysis.

5. GLOBAL METHODS

All current global predictive methods use selection of elements from a large set of basis elements. That is, one specifies a set of basis functions $\{B(x, \theta)\}, \theta \varepsilon \Theta$, defined on the space of predictor vectors such that "most" functions of x are in the span of the basis. Then, in regression, one uses a predictor function of the form

$$\hat{y}(x) = \sum_k \alpha_k B(x, \theta_k).$$

In classification, conditional probabilities are estimated as

$$\hat{p}(j \mid x) = G\left(\sum_k \alpha_{jk} B(x, \theta_k)\right)$$

for some specified function G. Here are some examples:

Neural Nets: $\{B(x, \theta)\} = \{\sigma(\theta \cdot x)\}, \Theta = \{E^M\}$.

CART: $\{B(x, \theta)\} = \{I(x \varepsilon R); R$ a rectangle in E^M, I an indicator function$\}$ Basis elements $I(x \varepsilon R_k)$ are selected such that the $\{R_k\}$ are disjoint with union E^M.

MARS: $\{B(x, \theta)\} = \{\Pi_i(\pm(x_{mi} - \theta_{mi})^+\}$, i.e., basis elements are products of a finite number of univariate linear splines.

For all of these sets of basis functions, various completeness theorems are known. These have the form: for all $f(x)$ of some specified smoothness and any $\varepsilon > 0$, there exists $K, \{\alpha_k, \theta_k\}$, $k = 1, \ldots, K$ such that

$$\|f(x) - \sum_k \alpha_k B(x, \theta_k)\| < \varepsilon.$$

This is comforting, but it leaves open questions important to applications:

> What are good sets of basis functions?
> How can a "good" subset of basis functions
> be selected?

There are drawbacks to the basis elements used in CART and MARS. CART can lose accuracy because its basis elements are discontinuous and are aligned with the coordinate axes. The MARS basis elements are continuous but unbounded. Also, with a high dimensional data set, it is not computationally feasible to include basis functions that are products of more that a few univariate splines.

The strategies familiar to statistics for selecting basis elements consists of stepwise "optimal" addition. For instance, in CART each current basis function is "optimally split" to give two new basis functions. A similar strategy is used in MARS. Because of this stepwise add-one-at-a-time approach and some clever algorithms, the basis selection procedure goes very rapidly. On the other hand, neural networks optimize the choices over all basis elements simultaneously using backpropagation.

6. WHAT IS UNIQUE AND DIFFERENT ABOUT NEURAL NETS?

Having come this far, we are in a position to venture some guesses as to why neural networks seem to give good results over a wide range of data bases. There may be two contributing factors. The first is that the basis element have desirable properties.

They are very smooth functions of linear functions and nicely bounded above and below. Their form, being close to zero in one portion of the space and close to one in another portion make them particularly good for approximating conditional probabilities and for approximating local ripples.

Another property may explain why NN users can throw in thousands of parameters and not have catastrophic overfitting. Usually, in starting the NN fit, one uses small random coefficients for the linear combinations in each sigmoid. If all of the coefficients in β are small, then $\sigma(\beta x) \cong .5 + .25\beta x$. Then, the sum over all sigmoid functions whose coefficients remain small collapses into a single linear function with the number of equivalent parameters equal only to the number of coordinates in the x-vector.

The other unique element in neural nets is the idea of simultaneously selecting all basis elements using backpropagation. My first impression of this method was that it was bound to fail by winding up in poor local minima. This does not seem to happen and the why is mysterious. It may be wound up in the nature of backpropagation. By this, I mean the particular procedure of entering one case at a time and then taking a partial gradient step.

For instance, the general wisdom is that one-case-at-a-time works better than putting in all of the data and doing "batch" gradient descent. Certainly, there are much faster methods for nonlinear optimization than gradient descent. But while these are faster, it is not known if they produce the accuracy given by backpropagation.

There is some research that claims to establish a link between backpropagation and stochastic optimization methods known to converge a.s. to the global optimum. If this is even partially true, then the method's largest drawback, its painfully slow running time, may also be a source of its consistent accuracy. Unfortunately, this is largely unexplored territory.

A possibility that Jerry Friedman and I are exploring is stepwise entry of sigmoid basis functions. We have designed a fast algorithm for stepwise entry of sigmoid functions patterned after the stepwise entry of hinge functions given in Breiman (1993). The procedure produces fits to the data in several orders of magnitude less running time than backpropagation. We have not done enough testing to know if the accuracy is competitive with neural networks using backpropagation.

7. CODA

I am fond of the saying "give a man a hammer and every problem looks like a nail". The NN community has their hammer. But they are also hard at work devising pliers, saws, chisels and a full repertory of tools, large and small. Interesting new methods are spawned at an almost alarming rate.

Among many recent results, here are a few that impressed me: A smart new metric leads to a nearest neighbor misclassification rate on optical character recognition about half that of a well-tinkered neural net procedure (Simard, Le Cun and Denker, 1993). Coding problems involving many classes into a sequence of two class problems results in significant decreases in error rates (Dietterich and Bakiri, 1991). Combining ("stacking") dissimilar classifiers also gives reduced error rates (Wolpert, 1992). Using linear programming methods to get nonlinear separating boundaries between classes gives error rates on optical character recognition lower than neural nets (Boser, Guyon and Vapnik, 1992).

Often the analogies and language used in the NN community obscure the data analytic reality. There is a lack of reflective introspection into how their methods work, and under what data circumstances. But these lapses are more than offset by the complexity, interest, size and importance of the problems they are tackling; by the sheer creativity and excitement in their research; and by their openness to anything that works.

Comment: Neural Networks and Cognitive Science: Motivations and Applications

James L. McClelland

Artificial neural networks have come and gone and come again—and there are several good reasons to think that this time they will be around for quite a while. Cheng and Titterington have done an excellent job describing that nature of neural network models and their relations to statistical methods, and they have overviewed several applications. They have also suggested why neuroscientists interested in modeling the human brain are interested in such models. In this note, I will point out some additional motivations for the investigation of neural networks. These are motivations arising from the effort to capture key aspects of human cognition and learning that have thus far eluded cognitive science.

A central goal of congnitive science is to understand the full range of human cognitive function. During the 1960s and 1970s, when symbolic approaches to human cognition dominated the field, great progress was made in characterizing mental representations and in capturing the sequential thought processes needed, for example, to solve arithmetic problems, to carry out deductive reasoning tasks, even to prove theorems of logic from given axioms. Indeed, by 1980 a general computer program for solving integro-differential equations had been written. These accomplishments are certainly very valuable, yet they still leave many scholars of cognition with the very strong feeling that something very important is missing. Efforts in machine recognition of spoken and visual input, machine understanding of language, machine comprehension and analysis of text, not to mention machine implementation of creative or insightful thought, all continue to fall short. A huge gap remains between the capabilities of human and machine intelligence.

The interest in the use of neural networks among cognitive scientists springs largely from the hope that they will help us overcome these limitations. Although it is true that there is much to be done before this hope can be fully realized, there are nevertheless good reasons for thinking that artificial neural networks, or at least computationally explicit models that capture key properties of such networks, will play an important role in the effort to capture some of the aspects of human cognitive function that have eluded symbolic approaches. In what follows I mention two reasons for this view.

The first reason arises in the context of a broad class of topics that can be grouped under the rubric of "interpretation." A problem of interpretation arise whenever an input is presented to the senses, be it a printed digit, a footprint, a scientific argument or a work of creative expression such as a poem or a painting. The problem is to determine what the thing is or what it is intended to signify. The problem is difficult because the direct data is generally insufficient so that the ability to determine the correct interpretation depends on context.

Let us consider two examples. The first, shown in Figure 1, is from Massaro (1975) and illustrates the role of context in letter recognition. The same input gives rise to two very different interpretations depending on the context in which it occurs. The second comes from very simple stories of a kind studied by Rumelhart (1977):

> Margie was playing in front of her house when she heard the bell on the ice

James L. McClelland is Professor of Psychology and Professor of Computer Science, Department of Psychology, Carnegie Mellon University, Baker Hall 345-F, Pittsburgh, Pennsylvania 15213.

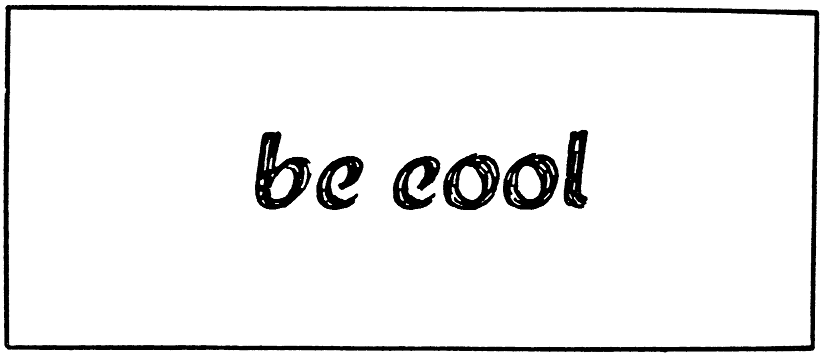

FIG. 1. *The same visual configuration can be interpreted as two different letters, depending on the context. Reprinted with permission from Massaro* (1975) *p.* 382.

> cream truck. She remembered her birthday money and ran inside the house.

In this case, human readers have no trouble figuring out that Margie's birthday money is probably in the house and that she probably ran in to get it so that she could buy herself ice cream. Obviously, this interpretation, engendered by the second sentence of the above story, would not arise if the context were changed:

> Margie lived in a dangerous neighborhood with lots of drug addicts always on the lookout for innocent passers-by to rob. She was coming home from a birthday visit to her grandmother when she saw a couple of the addicts loitering at the corner near her house. She remembered her birthday money and ran inside the house.

What the Massaro and Rumelhart examples have in common is the fact that the direct information—the shape of the character, the words in a sentence—is often not enough by itself to get the correct interpretation. But context is not in general enough by itself—indeed the context often provides only very general and indirect constraints. What one is left with is the sense that it is the aggregated influence of the sum total of the cues rather than any one operating individually that is of crucial importance. Indeed, in real situations it is often the case that ambiguity remains once all the factors have been taken into account. Many psychologists have long argued that it is reasonable to view all acts of interpretation as closely related to Bayesian inference, in that they involve the weighted combination of various direct and contextual cues together with prior biases. Signal detection theory (Green and Swets, 1966), based on a Bayesian analysis of decision making under uncertainty, is a centerpiece of this line of thinking.

As Cheng and Titterington point out, neural networks provide a natural domain for capturing perception and interpretation as probability optimization problems in which direct and contextual information is combined to reach the most likely interpretation given the available input. The use of graded (real-valued) connection weights allows the appropriate weighting of different sources of evidence. The process of settling to a stable attractor state captures nicely the multifaceted nature of most interpretation problems in which the interpretation of one part of an input both influences and is influenced by the interpretation of every other part. Human subjects often behave in ways that are highly consistent with optimal statistical methods (Massaro, 1989) and, indeed, connectionist models that share these properties have been highly successful in accounting for psychological data from perceptual decision tasks (McClelland and Rumelhart, 1981; McClelland and Elman, 1986; McClelland, 1991). A wide range of authors have argued for the use of similar models in sentence comprehension, story understanding, visual scene interpretation and many other related tasks based on the general fact that correct interpretation is not in general possible. The only way to maximize the probability of making the correct decision is to exploit all sources of information.

A second reason why neural networks are relevant to cognitive science arises in the area of learning. Psychological research on learning has gone through many different phases, including a phase lasting from around 1920 to nearly 1960 where it was dominated by stimulus-response theories (in which probabilistic formulations have proven very useful) and another that arose in the 1950s and persisted into the 1960s in which learning was conceptualized in terms of the formulation and testing of deterministic rules, within the symbolic tradition. This approach largely gave way in the 1970s and 1980s to a new approach based on the probabilistic use of accumulated knowledge from examples.

One of the most successful models in this tradition is a model of category learning due to Medin and Schaffer (1978). These authors argued that category learning occurs through the exhaustive storage of all examples in memory. When a test item is presented for categorization, it is compared to all of the examples in memory and each votes for its own category in proportion to its similarity to the test item. The probability of choosing a particular category is equal to the sum of the votes of all of the known exemplars in the category divided by the sum of all of the votes. The key point is that the responses subjects make are probabilistic, not deterministic; and they reflect the influence of specific examples rather than general rules. Neural network models are highly relevant to capturing this kind of learning since each experience leaves its own residue in the form of changes to the connection weights among the units in the network. Indeed, the Medin and Schaffer model can easily be formulated as a neural network model, and a recent, highly successful connectionist model of category learning due to Kruschke (1992) takes just this approach. Kruschke's model makes use of individual units to represent each exemplar and extends the Medin and Schaffer model by using an error correcting learning rule to modify the strengths of the contributions each exemplar makes to the activation of each of the possible categorization responses.

A related difficulty for deterministic rule systems arises in various domains of language. In general, language production and interpretation can both be thought of as mapping problems in which a message in one form of representation must be translated into another form of representation. As two examples, the problem of producing a verb to describe a state or action one wishes to convey, and the problem of producing a spoken sound that corresponds to a written word can both be thought of as mapping problems. In general, in natural languages such problems often involve what might be called quasi-regular—or even better probabilistic—structure. In mapping from spelling to sound, for example, there are important regularities; but at the same time there are many exceptions as well. Often, the exceptions are not simply isolated individual cases but are grouped together in clusters; for example, in English spelling there are many words that violate the rule that EA corresponds to the long E sound as in HEAT; most of these words—THREAD, TREAD, BREAD, etc.,—end in EAD but not all do (cf. DEAF) and not all of the words that end in EAD are exceptions to the standard EA correspondence (cf. BEAD; and the homographs READ and LEAD). Thus, the relationship between EA and its pronunciation is statistical. Similar statistical relations exist between the present and past tense forms of many of the English verbs; thus, many monosyllabic verbs with the short 'ih' vowel followed by a velar consonant (dig, swing) form the past tense by changing 'ih' to 'uh' (did-dug, swing-swung). Again, the regularity is statistical rather than deterministic (cf. sing-sang, and ring, which can be rang or ringed depending on the meaning intended).

One approach to learning mappings of this sort is to propose that they are handled by dual learning systems: one that learns the general rules and another that contains a list of the exceptions (Pinker and Prince, 1988; Coltheart et al., 1994). A different approach, first presented in the Rumelhart and McClelland (1986) model of past tense formation and the Sejnowski and Rosenberg (1987) NETtalk model for translation from spelling to sound, assumes that the entire quasi-regular system can be acquired in a single multilayer network. These systems share with the Medin and Schaffer model of category learning the property that individual items (in this case words)—especially those that occur frequently in the learner's experience—influence the response the network makes to other similar items. At the same time, they show how these effects of individual items can cumulate to produce outputs for novel items that conform to regularities that many examples share. There has been considerable debate about the adequacy of these one-process systems. The first models introduced did have some inadequacies, but recent models in both domains (MacWhinney and Leinbach, 1991; Plaut and McClelland, 1993) address the main concerns and demonstrate that a single system can be adequate to capture both the regularities and the exceptions. While it remains debatable whether the deeper aspects of language can be captured by neural network models, it seems clear, at least to this writer, that the problem of translation from streams of words to an appropriate semantic interpretation is quasi-regular (see McClelland, St. John and Taraban, 1989). Thus, it seems very likely that many of the statistical properties of neural network models will be evident in any successful model of language use and language acquisition.

To summarize, two very general and central tasks for cognitive systems—the task of interpretation and the task of learning—appear in essence to be statistical in nature. Artificial neural networks are attractive mechanisms for modeling such tasks because, as Cheng and Titterington make clear, neural networks are essentially devices that implement statistical processes. Given this, the current burgeoning of interactions between mathematical statistics and neural network research is a welcome development for cognitive science. Such interactions will lead to a deeper understanding of the interpretation and learning tasks, and may ultimately help us to address other cognitive tasks, perhaps including creative thinking and scientific discovery, as well.

ACKNOWLEDGMENTS

Support for the preparation of this article was provided by NIMH Project Program Grant MH47566-03 and by NIMH Research Scientist Award MH00385-13.

Comment

B. D. Ripley

Bing Cheng and Mike Titterington have reviewed many of the areas of neural networks; their paper overlaps the flood of books on the subject. I also recommend Weiss and Kulikowski (1991) (Segre and Gordon, 1993, provide an informative review) and Gallant (1993) for their wider perspective and Wasserman (1993) for coverage of recent topics. My own review article, Ripley (1993a), covers this and many of the cognate areas as the authors comment. The five volumes of the NIPS proceedings (*Advances in Neural Information Processing Systems*, 1989–1993, various editors) provide a very wide-ranging overview of highly-selected papers. Much of the latest work is available electronically from the ftp archive at `archive.cis.ohio-state.edu` in directory `pub/neuroprose`.

At the time I received this paper to discuss, I had recently attended a NATO Advanced Study Institute on *From Statistics to Neural Networks* (whose proceedings will appear as Cherkassky, Friedman and Wechsler, 1994), which despite the direction of the title revealed that current thoughts in neural networks are not to subsume statistics in neural networks but vice versa. Many researchers in neural networks are becoming aware of the statistical issues in what they do and of relevant work by statisticians which encourages fruitful discussions.

Cheng and Titterington concentrate on similarities between statistical and neural network methods. I feel the differences are more revealing as they indicate room for improvement on at least one side. However, I believe the most important issues to be those of practice which are almost ignored in the paper. Before I turn to those, there are two points I wish to attempt to clarify.

1. PROJECTION-PURSUIT REGRESSION

The connection between multilayer perceptrons (MLPs) and projection-pursuit regression (PPR) is much deeper than the authors appear to suggest. Other empirical comparisons (apart from my own cited in the paper) are given by Hwang et al. (1992a,b, 1993), and Barron and Barron (1988) viewed PPR from a network viewpoint. In the authors' notation PPR is

$$y_i = w_{0i} + \sum_k \gamma_i \psi_k(x^T v_k),$$

where I have allowed for multiple outputs. An MLP with linear output units is the special case of logistic ψ_k; of course both PPRs and MLPs can be given nonlinear output units. Since we can approximate any continuous ψ_k of compact support uniformly by a step function and can approximate (nonuniformly) a step function by a logistic, we can approximate ψ_k uniformly by a sum of logistics. This fact plus the (elementary) approximation result for PPR of Diaconis and Shahshahani (1984) gives the approximation results of Cybenko and others. There is a version of Barron's L_2 result for PPR by Zhao and Atkeson (1992). (This point of view, approximating ψ_k by a simple neural net of one input, corresponds to organized weight-sharing between input-to-hidden-unit weights for groups of units, a sensible procedure in its own right.)

These results suggest that the approximation capabilities of MLPs and PPR are very similar (suggesting an affirmative partial answer to the question in Section 7). However, PPR will have an advantage when there are many inputs, only a few combinations of which are relevant, in making better use of each projection and hence fewer projections and parameters. My suspicion is that this is commonly the case.

B. D. Ripley is Professor of Applied Statistics, University of Oxford, 1 South Parks Road, Oxford OX1 3TG, United Kingdom. This comment was written while on leave at the Isaac Newton Institute for Mathematical Sciences, Cambridge, United Kingdom.

2. HANDWRITTEN DIGIT RECOGNITION

The literature on handwriting recognition, especially studies of Zip-codes, is much misquoted and I suspect much misunderstood. Many of the best methods for handwriting recognition depend on choosing good features, and the lower levels of the Le Cun system can be thought of as feature extraction not classification and were originally optimized by hand. The actual results are often stated confusingly (including in Le Cun et al., 1990). There is a training set of size 9709, 7291 of which are handwritten, and a test set of 2007 handwritten characters plus some others. According to Vapnik (1992) the test-set error rate for the Le Cun system is 5.1% (102/2007) (but the 1990 variant would appear to achieve 4.6% (92/2007)) against 2.5% for humans and 3.3% for the best automated system to date. Undoubtedly, the automated systems have been optimized for this particular dataset, so these rates may be a little optimistic.

Later workers have suggested that the handcrafting is not necessary and report similar results from simpler applications of neural networks: Knerr, Personnaz and Dreyfus (1992) and Martin and Pitman (1990, 1991). Grother and Candella (1993) report best results for the *probabilistic neural network* of Specht (1990), that is kernel discriminant analysis, using up to 64 principal components of the 128^2 image data as input features. These are all general purpose methods, at least as much so as penalized discriminant analysis (PDA) and achieve error rates of around 2.5%. Against this, the value of PDA, with an error rate of 8.2%, is surely overstated.

This example shows the difficulty of quoting error rates without reference to the Bayes risk. The latter can often be estimated (Fukunaga, 1990; Ripley, 1994b); but in this case, it must be close to the error rate achieved by humans. It is also potentially confusing to quote per-digit error rates when the task depends on correctly reading whole Zip-codes. That task has some redundancy (not all possible Zip-codes are valid nor equally probable) and high correlation in the errors for the separate digits. The residual error rate contains both segmentation errors in isolating the digits and plain errors (wrongly labelled digits). There is substantial interwriter variability, and careful studies (such as that of Grother & Candella) use different writers for the training and test sets.

3. OPTIMIZATION IN FITTING MLPs

The comments in Sections 4.2.2 and 4.3.2 hide a series of very important practical points. Some authors argue that the point of the back-propagation gradient-descent algorithm is *not* to minimize $E(W)$ since doing so will lead to over-fitting. The regularization approach is to add a term to penalize rough functions (such as weight decay) and so change the objective to a function we really do want to minimize. Other people believe in stopping early as a means of regularization, although why travelling along a path in the wrong direction to the nearest point to a goal is thought a good procedure beats me. (It also occurs in statistical approaches to tomography, e.g., Vardi and Lee, 1993.)

What is clear is that no experienced worker attempts to minimize $E(W)$ alone, and this makes comparisons of methods difficult. A typical approach is to stop when the error measure on a validation set starts to rise. This has a number of difficulties:

- To repeat the point, there is no guarantee that the path taken is sensible.
- In my experience, the error on the validation set often rises for a while then falls dramatically before rising again; therefore, it is impossible to know that the best point on the path has yet been reached.
- The use of a validation set wastes data, and I suspect that often the test set is used. One example, in a textbook, is Thornton (1992, p. 199).

A further difficulty is the prevalence of local minima, which are much more common than comments in the literature (e.g., Thornton, 1992, Section 13.6) suggest—it needs careful work to discover many of the minima of the error surface.

Schiffmann, Joast and Werner (1992) and Jervis and Fitzgerald (1993) report studies of a wide range of optimization techniques on a narrow range of problems, and both review the literature. Their conclusions differ, and their experience differs from my own. It does seem that the more sophisticated methods (such as quasi-Newton and conjugate gradients) do best in hard optimization problems, often dramatically so (e.g., Grother and Candella, 1993), but can be beaten by on-line gradient descent methods on simpler tasks.

The back-propagation algorithm can be extended to compute second derivatives in some or all directions (Bishop, 1992; Buntine and Weigend, 1993; Pearlmutter, 1994). Interesting developments in this area include RProp (Riedmiller and Braun, 1992) and scaled conjugate gradients (Møller, 1993) which can make use of Pearlmutter's techniques.

It is worth noting that in the Bayesian approach the effort of minimization is redirected to integration over the weights, either by a saddlepoint approximation or by Monte-Carlo methods (e.g., Neal, 1993). (We will almost never be interested in the weights *per se* despite the emphasis of Section 4.3.5.) It is not yet clear how much effort is needed to do the integration well.

4. METHODS FOR CLASSIFICATION

The authors mention Hastie, Tibshirani and Buja's FDA in Sections 4.3.1 and 7. These authors and I studied Breiman and Ihaka's unpublished 1984 paper to see if such simple results had a simple explanation and rederived the results via canonical correlations. My version will appear in Ripley (1993b, 1994b) and in detail in Ripley and Hjort (1994).

For two normally distributed classes with a common covariance matrix, it is well known that the sample linear discriminant (LDF) is more efficient than logistic discrimination since it uses full rather than conditional maximum likelihood and that the LDF can be found by regression up to the additive constant.

We can think of the linear regression as the best linear approximation to the posterior probabilities and extend this to more than two classes. As a principle of classifier design, this has been used (Duda and Hart, 1973; Devijver and Kittler, 1982; Fukunaga, 1990) under the name of *minimum (mean) square error* classifiers. Unlike the linear discriminant, this procedure classifies by the nearest target or equivalently the largest component of a regression for each class indicator. What Breiman and Ihaka showed is that the regressions span the same space as the canonical variates and that the linear discriminant classifies by choosing the nearest target in a non-Euclidean metric in that space.

Neural networks (at least, MLPs and RBFs) are nonlinear regressions. This suggests a number of ways to use them for classification:

- (What Hastie, Tibshirani and Buja, 1992, called FDA). Regress the class indicators on the input variables and use LDA in the space of fitted values. Equivalently, encode the classes in scores and regress the scores on the inputs.
- Use the functions in a nonlinear model for the log posterior probabilities. This is sometimes known as *softmax* in this field and fitted via maximum likelihood and is possibly penalized by, say, weight decay.
- Use the functions for separate nonlinear logistic models for each class *versus* the rest, as in an MLP with logistic output units. Although apparently less sensible than the previous method, this is by far the most commonly used, for example, in the Le Cun study.
- Choose well-separated scores for the classes and regress on the inputs (Dietterich and Bakiri, 1991).

The authors appear to prefer the first method, but they probably have no practical experience. I have found a number of difficulties, over many experiments, that stem from the need to estimate the within-class covariance in the space of fitted values. For fits from nonlinear regressions (including MLPs, RBFs, MARS and projection pursuit regression) the covariance matrix can be dominated by outliers; and even with robust estimation, it can be insufficiently well determined. My current preference is for the second approach, but this raises problems for techniques such as MARS that are tailored to least-squares fitting. My impression is that how the flexible family of functions is used is much more important than which family is chosen.

5. WHAT CAN NEURAL NETWORKS ACHIEVE?

It is no accident that all the real examples Cheng and Titterington chose are classification problems; in my reading, these form over 90% of the applications with regression techniques being used in time series (Weigend and Gershenfeld, 1993) and control (Miller, Sutton and Werbos, 1990). Great advances have been claimed for neural networks, but more careful studies have shown that in many of the cited examples statistical methods can do as well or even much better. (For NETtalk, Wolpert, 1990; for digit recognition, Grother and Candela, 1993; for the sonar problem of Gorman and Sejnowski, 1988a, b; Ripley, 1994a.) Often linear methods or k-nearest neighbour methods, used carefully, will do as well as neural networks.

There should, though, be a place for methods between the linear parametric methods and wholly nonparametric methods for highly-parametrized methods such as MLPs, RBFs, MARS and projection pursuit regression, especially in problems with significantly curved structure and relatively few data points.

One thing clients often require is to be able to *understand* the classifier. This is difficult with blackbox systems such as neural networks and is often claimed as an advantage of machine-learning systems such as tree- and rule-induction systems (Quinlan, 1993; Thornton, 1992). This may be true if there is a simple true classifier. In other cases, the true relationship between classes appears to be too complicated to be perceived easily (such as the forensic glass example in Ripley, 1994a, b). Humans often find rules easiest to comprehend; and any classifier can be approximated by a rule system, for example, by generating examples from it and inducing rules from these (as in Gallant, 1993 or Quinlan, 1993).

Issues of choosing model complexity and assessing performance and "generalization" (Section 4.3.4) are among the most important open questions. There is some evidence that methods such as cross-validation and AIC are too "local" to fully assess the variability of very flexible methods; therefore some of the assessed benefits of nonlinear methods may be illusory. [On "generalization", Haussler (1992) is a far-reaching extension of the ideas of VCdim to which statisticians, especially David Pollard and Luc Devroye, have contributed; and Anthony and Biggs (1992) is an introductory text on the seminal ideas of Blumer et al., 1989.]

One thing statisticians can contribute to the debate is experience in careful use of sophisticated nonlinear methods. Software is readily available, including in S, and I would encourage statisticians to experiment rather than quote inadequately designed propaganda studies.

To end on a positive note, some very impressive applied statistics is being done using neural networks, and the explosive growth of the subject has opened the eyes of some statisticians (including myself) to the complexity of problems that may be fruitfully attacked by nonlinear methods. I and others have been particularly impressed by some work of my Oxford Engineering Science colleague, Lionel Tarassenko, on analyzing sleep EEG data using both Kohonen nets and radial basis functions to detect structure and anomalous signals (Roberts and Tarassenko, 1993, 1994).

Comment

Robert Tibshirani

Cheng and Titterington's paper is a scholarly overview of the field of neural networks. It should raise the statisticians' awareness of this interesting and important field. One of the authors' objectives was to encourage cross-disciplinary research between neural network researchers and statisticians. Here at the University of Toronto, I have been collaborating informally with Geoffrey Hinton of the Computer Science department, and I think that this collaboration has been fruitful for both of us.

First I would like to make a general point drawing a distinction between statistics and neural networks:

Statisticians tend to work with more interpretable models, since measuring the effects of individual input variables, rather than prediction, is often the purpose of the analysis.

Having said that, there is still much that one field can learn from the other. I will briefly summarize some of the main points:

WHAT THE STATISTICIAN CAN LEARN FROM NEURAL NETWORK RESEARCHERS

1. We should worry less about statistical optimality and more about finding methods that work, especially with large data sets.
2. We should tackle difficult real data problems like some of those addressed by neural network researchers, like character and speech recognition and DNA structure prediction. As John Tukey has said, it is often better to get an approximate solution to a real problem than an exact solution to an oversimplified one.
3. Models with very large numbers of parameters can be useful for prediction, especially for large data sets and problems exhibiting high signal-to-noise ratios.
4. Modelling linear combinations of input variables can be a very effective approach because it provides both feature extraction and dimension reduction.
5. Iterative, nongreedy fitting algorithms (like steepest descent with a learning rate) can help to avoid overfitting in models with large numbers of parameters.
6. We (statisticians) should sell ourselves better.

WHAT THE NEURAL NETWORK RESEARCHER CAN LEARN FROM STATISTICIANS

1. They should worry more about statistical optimality or at least about the statistical properties of methods.
2. They should spend more effort comparing their methods to simpler statistical approaches. They will be surprised how often linear regression performs as well as a multilayered percep-

Robert Tibshirani is Associate Professor, Department of Preventive Medicine and Biostatistics, University of Toronto, 12 Queens Park, Toronto, Ontario M5S 1A8, Canada.

tron. They should not use a complicated model where a simple one will do.

The history of the projection pursuit regression (PPR) model illuminates some of the differences between the two fields. As noted by Cheng and Titterington, the PPR model has the same form as a single layer perceptron. When PPR was introduced into the statistics field by Friedman and Stuetzle in 1981, it did not have much practical impact. There are a number of possible reasons for this. Computationally, it was ahead of its time: many statisticians still do not feel comfortable using very large amounts of computation in an analysis. In addition, statisticians do not often tackle the large prediction problems that can often benefit from such an approach. Finally, the particular fitting (learning) procedure might have been too greedy to work effectively with large number of projections.

In contrast, neural network researchers have developed and applied the PPR model to some difficult problems with considerable success. In recent years, they have further improved their results by applying classical statistical techniques such as regularization, cross-validation and Bayesian modelling. This suggests that both fields should be listening and learning from each other. Cheng and Titterington's paper will help this cause.

Rejoinder

Bing Cheng and D. M. Titterington

We are very grateful to the discussants for the time and effort they have expended in commenting on our paper. When we submitted the revised version of the paper, we felt some trepidation that, in spite of our best effort at brevity, the paper still seemed very long in comparison to many other contributions to the journal, and yet we were fully aware that we had not done justice to important aspects of the field. Fortunately, some of our sins of omission have been absolved by the choice of discussants, and we are happy to regard many of their comments as complementary to our presentation. *Statistical Science* can, therefore, be said to be publishing a 10-author review of the interface between statistics and neural network research rather than a two-author review plus discussion. We are glad that the discussants include representives from what one may call (against Breiman's advice) the mainstream neural-network community (McClelland), as well as distinguished statisticians with both short and long (in terms of time) records of involvement in the area. We apologize to all discussants for not having space to respond to each of the many points they have made.

Later in our rejoinder we shall remark on some points raised by individual discussants, and we shall finish by pulling together views about the future of the interface. First, we mention three areas of research on which several discussants expressed views. These areas were implicitly identified by Amari and, in slightly different form, by Breiman.

- Mathematical modeling of real cognitive processes
- Theoretical investigations of networks and neurocomputing
- Development of useful tools for practical prediction and pattern recognition

MODELING OF REAL COGNITIVE PROCESSES

The dominant discussant here is McClelland. He emphasizes the fact that machine intelligence still has far to go to emulate many human mental processes, a view echoed by Bienenstock and Geman. McClelland sounds more hopeful than they do that concepts closely akin to artificial networks, presumably as known today, might prove to be key aspects. Furthermore, he suggests that the mechanics of statistics will be important in the development of such realistic cognitive machines: first, manipulation of probability models using Bayes' theorem could be the way to mimic the brain's approach to data analysis ("interpretation"); second, nondeterministic elements seem to be inevitable in modeling any realistic learning process. Practical realization of such models does, however, seem to be a daunting prospect. In the "interpretation" question, for instance, the equivalent of a prior distribution will have to include representation of all useful contextual and background information.

However, it seems clear from McClelland's penultimate paragraph that there are important new developments in areas such as speech processing, even in irritatingly irregular languages such as English and even then in the arguably less irregular American version. We are, nevertheless, doubtful about regarding pronunciation as necessarily statistical. Given enough context, in terms of neighboring letters, the correct pronunciation is presumably determined. Our very naive view would, therefore, be that a system of general rules with exceptions may be more natural; but perhaps the necessary complexity renders the approach too unwieldy.

MATHEMATICAL THEORY OF NETWORKS AND NEUROCOMPUTING

Major contributors under this heading are Amari and Barron, both of whom feel the lack of theoretical underpinning of certain aspects of the subject, in spite of what Amari calls the superficial, if positive, achievements of ANN models in applications. He mentions various approaches to the question of estimating generalization error, and he reemphasisizes the relevance of information geometry and the EM algorithm in studying and training Boltzmann machines. Also see Titterington and Anderson (1994).

Barron provides important details of the latest results in approximation theory, and his discussion of the case of feed-forward networks with two hidden layers is of particular interest. The ability to bound the risk associated with estimation procedures is of great value, although it appears that much still needs to be achieved in the practical area of bounding the risk associated with data-based model selection criteria: the practitioner wants to know how well he or she is doing in particular applications. Barron emphasizes the fact that the scale of computation time required for network estimation is still a problem and earmarks this area as "the most important task for theoretical research in neural networks." For instance, he solicits theoretical work both to tidy up the fuzzy (in a nontechnical sense) and sometimes contradictory folklore about techniques such as gradient search and to identify whether or not associated optimization techniques can be shown to produce a good solution in a reasonable time.

USEFUL TOOLS FOR PATTERN RECOGNITION

We shall use this heading to draw together the discussion about the practicalities associated with, in particular, the use of feed-forward networks in prediction and classification.

Breiman sets the scene with a clear description of what he calls the single (hidden) layer feed-forward network. (In the main paper, of course, we use the nomenclature "two-layer" for this architecture.) He highlights the "tinkering and tailoring" approach to network design and the practical awkwardnesses encountered when trying to estimate parameters by optimizing a multi-minima surface. Should one try several starting points and compare the resulting local minima? Should one use a validation set to determine the stopping point of the algorithm? Or, should one prevent overfitting by regularization? These are the sorts of questions to which Barron would like some theoretical answers to reinforce guidelines formulated from empirical studies. They are clearly relevant in complicated optimization problems beyond the estimation of parameters in feed-forward neural networks. However, the practice of optimization is clearly a messy bussiness. Ripley issues caveats about the use of a validation set and comments that the empirical experiences of himself and other investigators are different. At best, this suggests that behavior of optimization procedures in this area depends more than one might like on the particular application. Systematic recommendations may, therefore, be hard to come by, leading us to fall back on Breiman's experience-backed tinkering. To this end, well-designed and carefully assessed empirical studies such as those in Ripley's papers are clearly valuable, especially if they can include a wide range of really large problems.

Amari and Ripley highlight the issues of model complexity and assessment of generalization as important questions, reinforcing feedback we gleaned from our spy at the NATO Advanced Study Institute at Les Arcs. Although these are both theoretical questions, they are clearly related to practice, and we noted with interest the mention of stepwise model construction in the contributions by Barron and, in particular, Breiman. This is surely an important area for development.

Related to this is the question of the resulting classifier's interpretability. This is clearly one area where Ripley feels that multilayer perceptrons fail in comparison to some of their competitors. In addition, he is concerned about the computational demands they make and their complexity relative to other approaches. Classification trees, in contrast, have easily interpreted rules. Sometimes, however, interpretability can be a double-edged sword, especially if the classification tree is interpreted as portraying a physical explanation of the differences among the classes. In multiple regression with many covariates, many regressions based on different subsets of the covariates may provide predictors of comparable abilities; however, some of the models may not include covariates that influence the response in a direct, physical way. Similarly, many classification trees are likely to be closely comparable in terms of performance, and there is no guarantee that the particular one chosen by a computer package represents a rule that directly reveals a physical process.

As promised earlier, we shall come back to some general issues at the end, but we now highlight a few points made by some of the individual discussants.

AMARI

We appreciated the extra historical perspective provided by Amari. We readily agree that it is wrong to lay all the blame for the dark period on Minsky and Papert. We are grateful for the unsurprising information that some of the basic ideas appeared in papers that predate the most familiar references. We too have recently read with interest the work of Jordan and Jacobs (1993), and we have a general interest in the various roles that the EM algorithm plays in this area.

BIENENSTOCK AND GEMAN

The comments of Bienenstock and Geman will strongly influence the final part of our discussion. Here, we merely note their discussion of the appeal of generalization and highlight their warning that it seems virtually impossible that adequate training sets will be available for the training of such sophisticated devices as fully successful object-recognizers. The demands of generalizability are too high. Bienenstock and Geman recommend that future emphasis should be on modeling rather than training. This is somewhat related to Ripley's point about using a family of functions well rather than worrying about which family of functions to choose.

BREIMAN

We apologize to Breiman that we have continued to use the phase "neural-network community" even in this rejoinder if only for the fact that we prefer not to write "nonmainstream-statistical community"! Breiman and others emphasize the problem-oriented approach of the "other" community. One might be indignant, sad or indifferent about the fact that they have the "good fortune not to have any formal statistical training". We hope (and assume) that this is not meant to imply that formally trained statisticians should not try to get involved!

RIPLEY

We are glad that Ripley has included the formulation of a more general projection-pursuit regression, and we are grateful for the recent references on various topics, including handwritten digit recognition. To the latter, we contribute the recent special issues of *Pattern Recognition* (March 1993) and *Pattern Recognition Letters* (April 1993). No doubt, several dozen further references relevant to our review will appear before its publication date: a measure of the speed of current development at this interface!

TIBSHIRANI

We look forward with great interest to the fruits of the Hinton-Tibshirani collaboration. We note the two-way flow of benefits between the two communities and are mildly surprised that statisticians appear to have something to gain in more ways than do neural network researchers. The two points labelled "1" have bearing on the final part of our rejoinder. So, in a way, does point 6 about statisticians being good self-sellers, which echoes the spirit of remarks of Bienenstock and Geman. At the risk of offending many nonarchetypal (and other) friends on both sides of the Atlantic, we should be less surprised about point 6 if most statisticians were British and most neural-network researchers non-British. Perhaps statisticians have just been around longer and are perceived as nondynamic; or perhaps, yet again, the title of their profession engenders an image of unimaginativeness.

STATISTICS AND NEURAL NETWORKS: THE WAY FORWARD

In this final part of our rejoinder, we try to formulate a perspective of the future synergy, if any, between research in statistics and neural networks.

The development and application of (artificial) neural networks has clearly been explosive and accompanied by much hype. Hype can cause two types of reaction. First, it can, like any successful and energetic advertising campaign, stimulate great interest and many acolytes as a result of appealing packaging, ambitiously stated goals and apparently successful applications, as described by Bienenstock and Geman. On the other hand, hype can be offputting. Breiman appears to have been affected this way initially, and we must confess that an instinctive suspicion of glossy advertising was the stimulus of our early reading in the area a few years ago. We felt that there must be something of statistical interest going on but surely nothing fundamentally novel.

At one level, the conclusion is anticlimactic. In the context of classification, in particular, neural-network models provide nonlinear predictors that are, under certain weak conditions, universal approximators. However, they both overlap with procedures that are well known to statisticians and, in many applications, seem to have no advantage over simpler methods. In addition, implementation of the predictors in practice appears to involve either simply recourse to a black-box application or a development process that is much less systematic than would satisfy statisticians. We might, therefore, decide to refer to the lessons offered by the comparative experiments of Ripley that they would be better advised to use different tools, and abandon the field.

We feel that this would surely be a mistake. While feed-forward networks, trained by the generalized delta rule, may not be cure-all classifiers let alone a realistic prototype for real neural structures, some of the contexts in which they have been used involve data of great volume and complexity. It is clear that the frontiers of complexity will continue to be attacked and, in principle, statisticians ought to be involved. Perhaps, as Breiman and Tibshirani indicate, statisticians will have to stray from their traditional paradigms in order to make meaningful impact, and many people will, no doubt, be reluctant to do so. Perhaps the paradigms can be suitably adapted; they are, after all, increasingly permeating the neural-network literature. In any case, many of the underlying problems of interest are of deep practical significance. As Bienenstock and Geman remark, they are attracting extremely able people who will inevitably have very clever ideas. It would be unworthy of statisticians to dismiss all of these as being too ad hoc, and it would certainly be foolish to be so blinkered as not to become informed about and involved in the key developments in this area.

We are pleased that all the discussants were positive about the involvement of statisticians at the interface, that McClelland saw the possible benefit of this to the cognitive science community and that the others indicated that statisticians would be enriched by participating. It seems to us that much of the challenge of the future has to involve the treatment of very large-scale problems, that whatever develops from current neural-network research should not be cursorily ignored and that statisticians have a contribution to make.

ADDITIONAL REFERENCES

AMARI, S. (1967). Theory of adaptive pattern classifiers. *IEEE Trans.* **16** 299–307.

AMARI, S. (1972). Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Trans. Comput.* **21** 1197–1206.

AMARI, S. (1977). Neural theory of association and concept-formation. *Biol. Cybernet.* **26** 175–185.

AMARI, S. (1985). *Differential-Geometrical Methods of Statistics. Lecture Notes in Statist.* **28**. Springer, Berlin.

AMARI, S. (1993a). Mathematical methods of neurocomputing, In *Chaos and Networks — Statistical and Probabilistic Aspects* (O.E. Barndorff-Nielsen et al., eds.) Chapman and Hall, London.

AMARI, S. (1993b). A universal theorem on learning curves. *Neural Networks* **6** 161–166.

AMARI, S. and MURATA, N. (1993). Statistical theory of learning curves under entropic loss criterion. *Neural Computation* **5** 140–153.

AMARI, S., YOSHIDA, K. and KANATANI, K. (1977). A mathematical foundation for statistical neurodynamics. *SIAM J. Appl. Math.* **33** 95–126.

ANDERSON, J. A. (1972). A simple neural network generating interactive memory. *Math. Biosci.* **14** 197–220.

ANTHONY, M. and BIGGS, N. L. (1992). *Computational Learning Theory: An Introduction.* Cambridge Univ. Press.

BARRON, A. R. and COVER, T. M. (1991). Minimum complexity density estimation. *IEEE Trans. Inform. Theory* **37** 1034–1054.

BERTSIMAS, D. and TSITSIKLIS, J. (1993). Simulated annealing. *Statist. Sci.* **8** 10–15.

BISHOP, C. (1992). Exact calculation of the Hessian matrix for the multilayer perceptron. *Neural Computation* **4** 494–501.

BLUMER, A., EHRENFEUCHT, A., HAUSSLER, A. and WARMUTH, M. (1989). Learnability and the Vapnik-Chervonenkis dimension. *J. Assoc. Comput. Mach.* **36** 926–965.

BOSER, B., GUYON, I. and VAPNIK, V. (1992). A training algorithm for optimal margin classifiers, *Digest Fifth ACM Workshop on Computational Learning Theory*, 144–152. ACM, New York.

BUNTINE, W. L. and WEIGEND, A. S. (1993). Calculating second derivatives on feed-forward networks. *IEEE Trans. Neural Networks*. To appear.

CHERKASSKY, V., FRIEDMAN, J. H. and WECHSLER, H., eds. (1994). *From Statistics to Neural Networks. Theory and Pattern Recognition Applications.* Springer, New York.

COLTHEART, M., CURTIS, B., ATKINS, P., and HALLER, M. (1994). Models of reading aloud: Dual-route and parallel distributed processing approaches. *Psychological Review.* To appear.

CYBENKO, G. (1988). Continuous valued neural networks with two hidden layers are sufficient. Technical Report, Dept. Computer Science, Tufts Univ.

DENBY, B. (1993). The use of neural networks in high-energy physics. *Neural Computation* **5** 505–549.

DEVIJVER, P. A. and KITTLER, J. V. (1982). *Pattern Recognition. A Statistical Approach.* Prentice-Hall, Englewood Cliffs, NJ.

DIACONIS, P. and SHAHSHAHANI, M (1984). On non-linear functions of linear combinations. *SIAM J. Sci. Statist. Comput.* **5** 175–191.

DIETTERICH, T. G. and BAKIRI, G. (1991). Error-correcting output codes: a general method for improving multiclass inductive learning programs. In *Proceedings of the Ninth National Conference on Artificial Intelligence*. AAAI Press.

FARLOW, S. J. (1984). *Self-Organizing Methods in Modeling: GMDH Type Algorithms*. Dekker, New York.

FUKUNAGA, K. (1990). *Introduction to Statistical Pattern Recognition*, 2nd ed. Academic, London.

GALLANT, S. L. (1993). *Neural Network Learning and Expert Systems.* MIT Press.

GORMAN, R. P. and SEJNOWSKI, T. J. (1988a). Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks* **1** 75–89.

GORMAN, R. P. and SEJNOWSKI, T. J. (1988b). Learned classification of sonar targets using a massively parallel network. *IEEE Transactions on Acoustics, Speech and Signal Processing* **36** 1135–1140.

GREEN, D. M. and SWETS, J. A. (1966). *Signal Detection Theory and Psychophysics.* Wiley, New York.

GROSSBERG, S. (1976). Adaptive pattern classification and universal recording. *Biol. Cybernet.* **23** 121–134.

GROTHER, P. J. and CANDELA, G. T. (1993). Comparison of hand-printed digit classifiers. U.S. National Institute of Standards and Technology report NISTIR 5209.

HAUSSLER, D. (1992). Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Inform. and Comput.* **100** 78–150.

HWANG, J.-N., LAY, S.-R., MAECHLER, M., MARTIN, D. and SCHIMERT, J. (1993). Regression modeling in back-propagation and projection pursuit learning *IEEE Trans. Neural Networks*. To appear.

HWANG, J.-N., LI, D., MAECHLER, M., MARTIN, D. and SCHIMERT, J. (1992a). A comparison of projection pursuit and neural network regression modeling. In *Advances in Neural Information Processing Systems 4* (J. E. Moody, S. J. Hanson and R. P. Lippmann, eds.), 1159–1166. Morgan Kaufmann, San Mateo, CA.

HWANG, J.-N., LI, D., MAECHLER, M., MARTIN, D. and SCHIMERT, J. (1992b). Projection pursuit learning networks for regression. *Engineering Applications Artificial Intelligence* **5** 193–204.

JERVIS, T. T. and FITZGERALD, W. J. (1993). Optimization schemes for neural networks. Cambridge Univ. Engineering Dept. Report CUED/F-INFENG/TR144.

JORDAN, M. I. and JACOBS, R. A. (1993). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*. To appear.

KNERR, S., PERSONNAZ, L. and DREYFUS, G. (1992). Handwritten digit recognition by neural networks with single-layer training. *IEEE Trans. Neural Networks* **3** 962–968.

KOHONEN, T. (1972). Correlation matrix memories. *IEEE Trans. Comput.* **C-21** 353–359.

KOHONEN, T. (1984). Self-organized formation of topologically correct feature maps. *Biol. Cybernet.* **43** 59–69.

KRUSCHKE, J. K. (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review* **99** 22–44.

MACWHINNEY, B. and LEINBACH, J. (1991). Implementations are not conceptualizations: Revising the verb learning model. *Cognition* **40** 121–158.

MARTIN, G. L. and PITMAN, J. A. (1990). Recognizing hand-printed letters and digits. In *Advances in Neural Information Processing Systems 2* (D.S.Touretzky, ed.) 405–414. Morgan Kaufmann, San Mateo, CA.

MARTIN, G. L. and PITMAN, J. A. (1991). Recognizing hand-printed letters and digits using backpropagation learning. *Neural Computation* **3** 258-267.

MASSARO, D. W. (1975). *Experimental Psychology and Information Processing*. Rand-McNally, Chicago.

MASSARO, D. W. (1989). Testing between the TRACE model and the fuzzy logical model of speech perception. *Cognitive Psychology* **21** 398–421.

MCCLELLAND, J. L. (1991). Stochastic interactive processes and the effect of context on perception. *Cognitive Psychology* **23** 1–44.

MCCLELLAND, J. L. and ELMAN, J. L. (1986). The TRACE model of speech perception. *Cognitive Psychology* **18** 1–86.

MCCLELLAND, J. L. and RUMELHART, D. E. (1981). An interactive activation model of context effects in letter perception, Part I: An account of basic findings. *Psychological Review* **88** 375–407.

MCCLELLAND, J. L., ST. JOHN. M: and TARABAN, R. (1989). Sentence comprehension: A parallel distribution processing approach. *Language and Cognitive Processes* **4** 287–335.

MEDIN, D. L. and SCHAFFER, M. M. (1978). Context theory of classification learning. *Psychological Review* **85** 207–238.

MILLER, W. T. III, SUTTON, R. S. and WERBOS, P. J., eds. (1990). *Neural Networks for Control*. MIT Press.

MØLLER, M (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* **6** 525–533.

NAKANO, K. (1972). Association—A model of associative memory, *IEEE Trans. Systems Man Cybernet.* **SMC-2** 381–388.

PEARLMUTTER, B. A. (1994). Fast exact multiplication by the Hessian. *Neural Computation*. **6** 147–160.

PINKER, S. and PRINCE, A. (1988). On language and connectionism: Analysis of a parallel distribution processing model of language acquistion. *Cognition* **28** 73–194.

PLAUT, D. C. and MCCLELLAND, J. L. (1993). Generalization with componential attractors: Word and nonword reading in an attractor network. In *Proceedings of the 15th Annual Conference of the Cognitive Science Society* 824–829. Erlbaum, Hillsdale, NJ.

QUINLAN, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.

RIEDMILLER, M. and BRAUN, H. (1992). RPROP—a fast adaptive learning algorithm. Technical Report, Univ. Kahlsruhe.

RIPLEY, B. D. (1993b). Neural networks and flexible regression and discrimination. In *Statistics and Images* (K.V. Mardia, ed.) *Advances in Applied Statistics* **1**. Carfax, Abingdon.

RIPLEY, B. D. (1994b). Flexible non-linear approaches to classification. In *From Statistics to Neural Networks. Theory and Pattern Recognition Applications* (V. Cherkassky, J. H. Friedman and H. Wechsler, eds.). Springer, New York.

RIPLEY, B. D. and HJORT, N. L. (1994). *Pattern Recognition and Neural Networks—A Statistical Approach*. Cambridge Univ. Press.

ROBERTS, S. and TARASSENKO, L. (1993). Automated sleep EEG analysis using an RBF network. In *Neural Network Applications* (A.F. Murray, ed.). Kluwer, Boston.

ROBERTS, S. and TARASSENKO, L. (1994). A probabilistic resource allocating network for novelty detection. *Neural Computation*. To appear.

RUMELHART, D. E. (1977). Understanding and summarizing brief stories. In *Basic Processes in Reading: Perception and Comprehension* (D. LaBerge and S. J. Samuels, eds.) 265–303. Erlbaum, Hillsdale, NJ.

RUMELHART, D. E. and MCCLELLAND, J. L. (1986). On learning the past tenses of English verbs. In *Parallel Distributed Processing: Explorations in the microstructure of cognition. Volume II* (J. L. McClelland, D. E. Rumelhart and the PDP Research Group, eds.) 216–271. MIT Press.

SCHIFFMANN, W., JOOST, M. and WERNER, R. (1992). Optimization of the backpropagation algorithm for training multilayer perceptrons. Preprint, Institute of Physics, Univ. Koblenz.

SEGRE, A. and GORDON, G. (1993). Book review: *Computer Systems that Learn* by S.M. Weiss and C.A. Kulikowski. *Artif. Intell.* **62** 363–378.

SIMARD, Y. S., LE CUN, Y. and DENKER, J. (1993). Efficient pattern recognition using a new transformation distance. *Neural Information Processing Systems*, **5**. Morgan Kaufman, San Mateo, CA.

TAKEUCHI, A. and AMARI, S. (1979). Formation of topographic maps and columnar microstructures. *Biol. Cybernet.* **35** 63–72.

THORNTON, C. J. (1992). *Techniques in Computational Learning. An Introduction*. Chapman and Hall, London.

TITTERINGTON, D. M. and ANDERSON, N. H. (1994). Boltzmann machines. In *Probability, Statistics and Optimization: A Tribute to Peter Whittle* (F. P. Kelly, ed.). Wiley, Chichester. To appear.

TSYPKIN, Y. Z. (1973). Foundations of the theory of learning systems. Academic, New York.

VAPNIK, V. (1992). Principles of risk minimization for learning theory. In *Advances in Neural Information Processing Systems 4* (J. E. Moody, S. J. Hanson and R. P. Lippmann, eds.) 831–838. Morgan Kaufmann, San Mateo, CA.

VARDI, Y. and LEE, D. (1993). From image deblurring to optimal investments: maximum likelihood estimation for positive linear inverse problems. *J. Roy. Statist. Soc. Ser. B* **55** 569–612.

VON DER MALSBURG, Ch. (1973). Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik* **14** 85–100.

WASSERMAN, P. D. (1993). *Advanced Methods in Neural Computing.* Van Nostrand Reinhold, New York.

WEIGEND, A. S. and GERSHENFELD, N. A., eds. (1993). *Times Series Prediction: Forecasting the Future and Understanding the Past.* Addison-Wesley, Reading, MA.

WEISS, S. M. and KULIKOWSKI, C.A. (1991). *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems.* Morgan Kaufmann, San Mateo, CA.

WOLPERT, D. H. (1990). Constructing a generalizer superior to NETtalk via a mathematical theory of generalization. *Neural Networks* **3** 445–452.

ZHAO, Y. and ATKESON, C. G. (1992). Some approximation properties of projection pursuit learning networks. In *Advances in Neural Information Processing Systems 4* (J. E. Moody, S. J. Hanson and R. P. Lippmann, eds.) 936–943. Morgan Kaufmann, San Mateo, CA.