Neural Networks: A Review from a Statistical Perspective
Author(s): Bing Cheng and D. M. Titterington
Source: *Statistical Science*, Vol. 9, No. 1 (Feb., 1994), pp. 2-30
Published by: Institute of Mathematical Statistics
Stable URL: http://www.jstor.org/stable/2246275
Accessed: 08-06-2018 18:20 UTC

REFERENCES
Linked references are available on JSTOR for this article:
http://www.jstor.org/stable/2246275?seq=1&cid=pdf-reference#references_tab_contents
You may need to log in to JSTOR to access the linked references.

JSTOR is a not-for-profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact support@jstor.org.

Your use of the JSTOR archive indicates your acceptance of the Terms & Conditions of Use, available at http://about.jstor.org/terms

Institute of Mathematical Statistics is collaborating with JSTOR to digitize, preserve and extend access to *Statistical Science*

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

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.*

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.

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

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

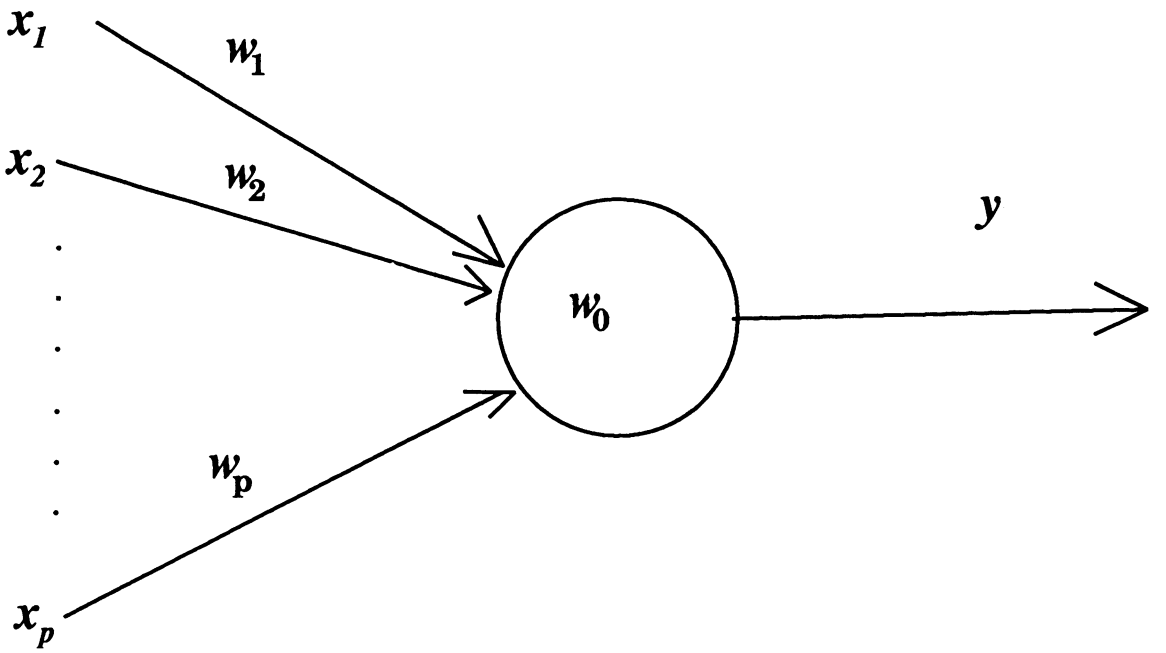

FIG. 1. *A simple (single-unit) perceptron.*

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

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

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.

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

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

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

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

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

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

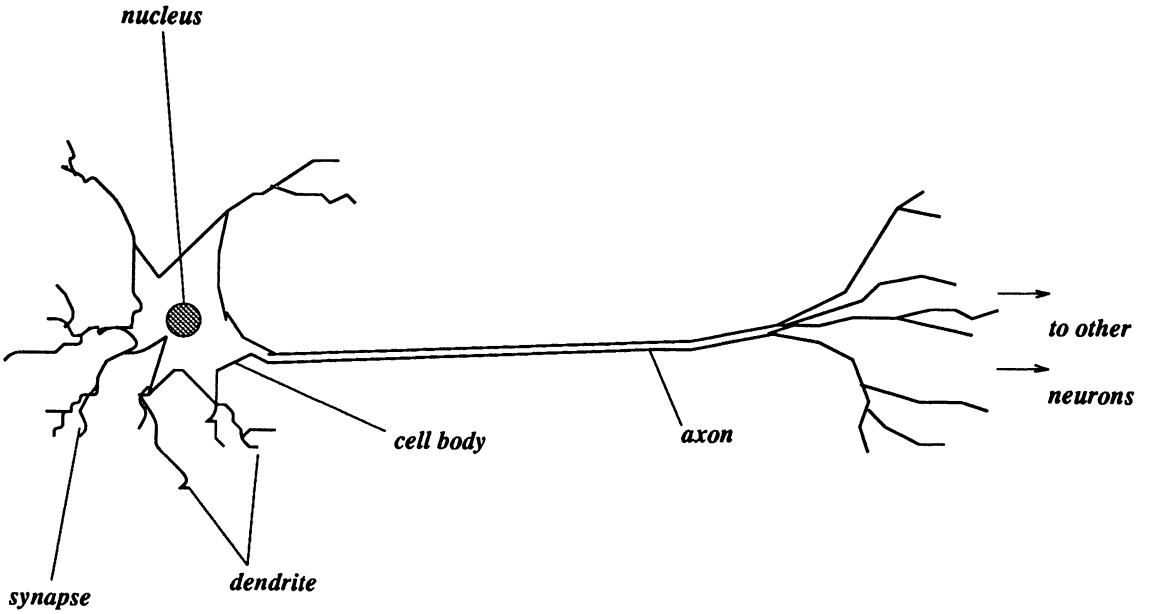

FIG. 2a. *Schematic diagram of real neuron.*

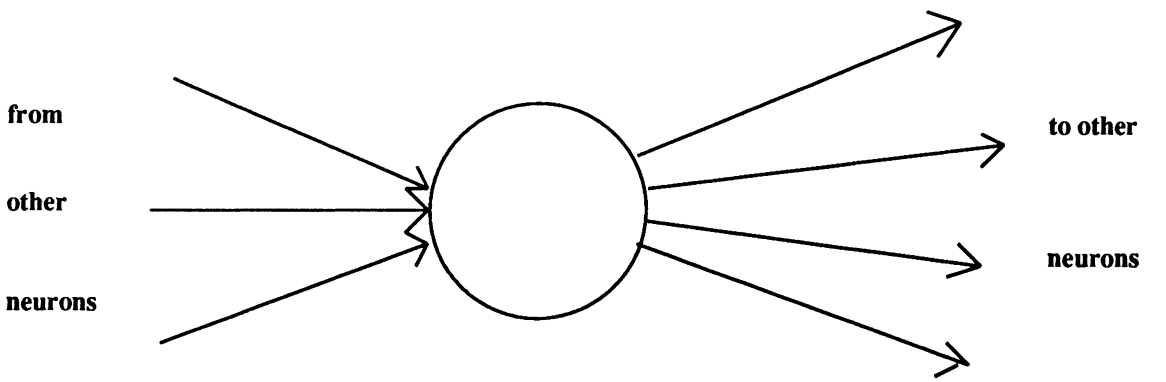

FIG. 2b. *Elemental version of* 2a.

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.

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

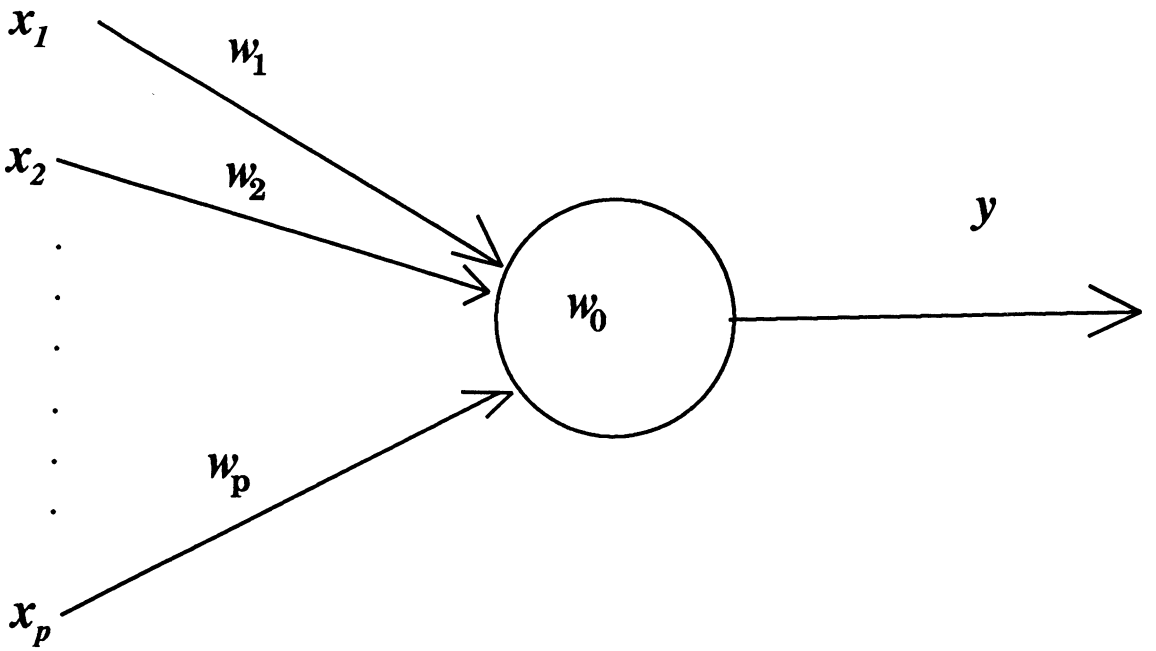

FIG. 2c. *The McCulloch-Pitts neuron.*

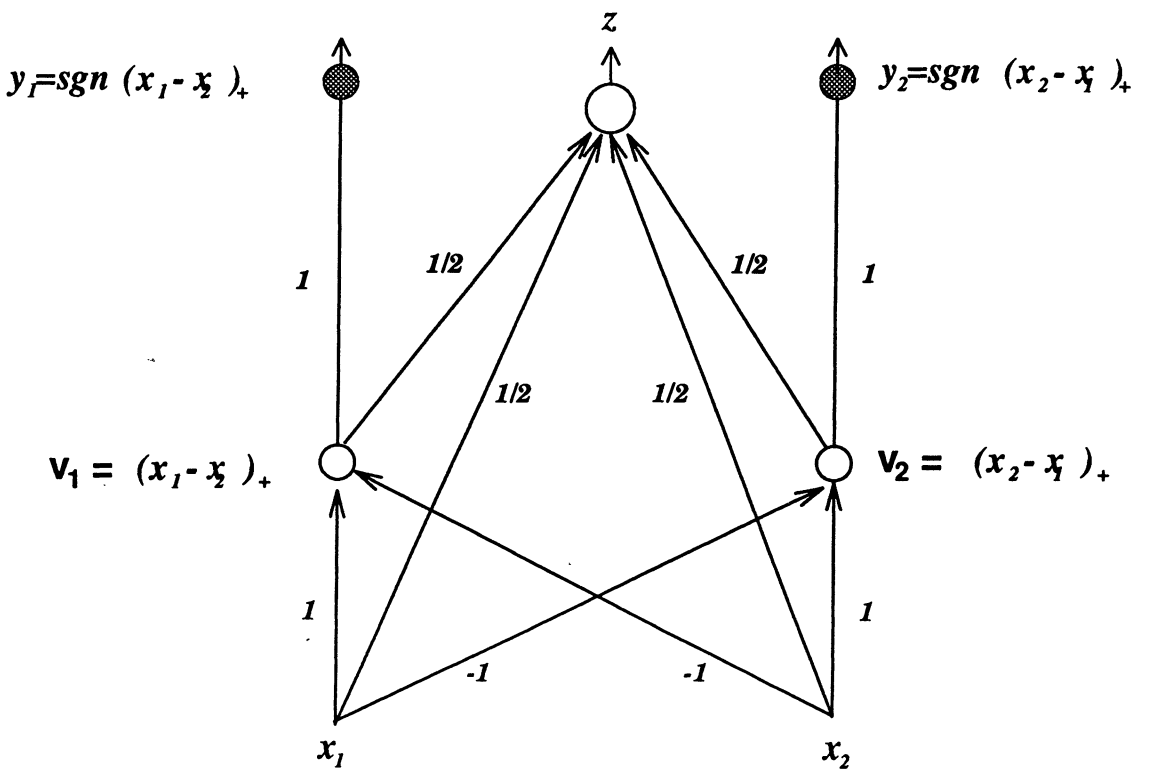

FIG. 3. *A network for finding the larger of two positive numbers, given eventually by* $z = \{\frac{1}{2}(x_1-x_2)_+ + \frac{1}{2}(x_1+x_2) + \frac{1}{2}(x_2-x_1)_+\}_+ = \max(x_1, x_2)$.

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

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

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 interlayer 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 $(30 \times 192) + 30 = 5790$ parameters. Layer H_3 is, in turn, fully connected to the output layer, requiring $(10 \times 30) + 10 = 310$ 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

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

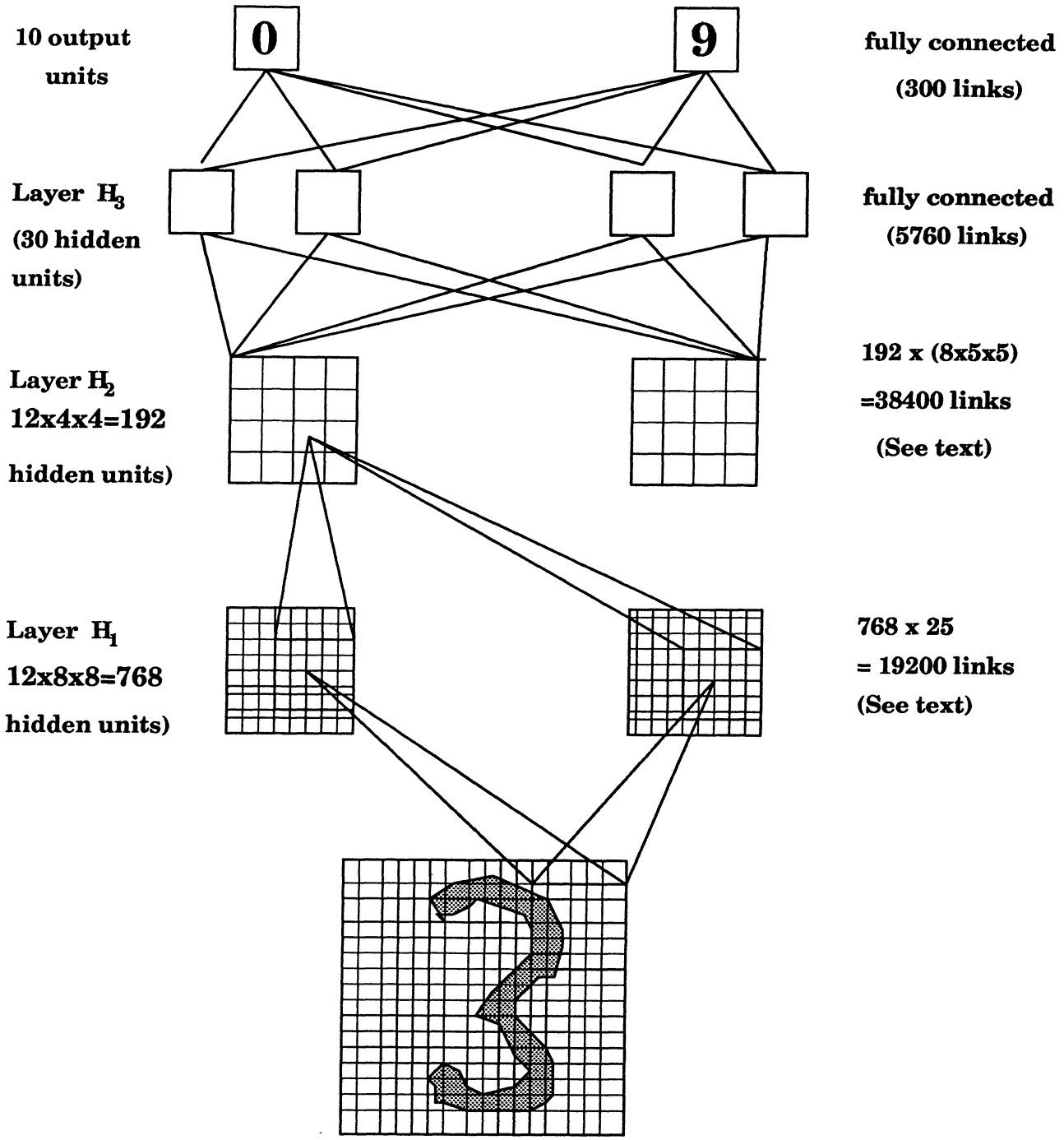

FIG. 4. *The network developed by Le Cun et al.* (1989) *for Zip-code recognition.*

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-

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

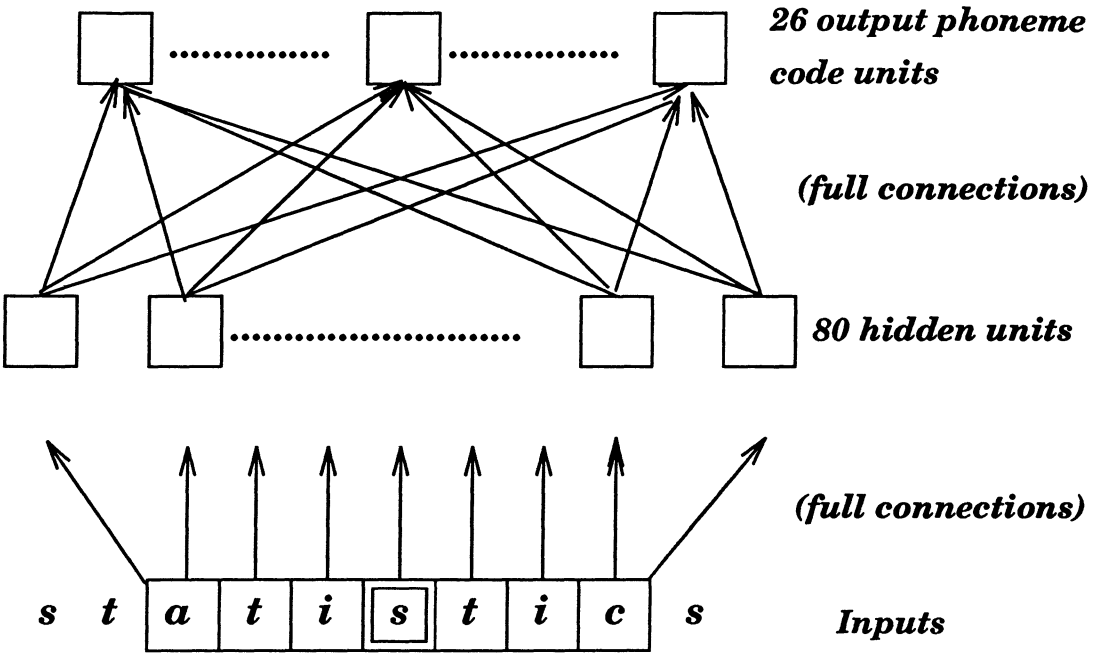

FIG. 5. *Structure of network used in NETtalk.*

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 $\{0, 1, \ldots, 9\}$. 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

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

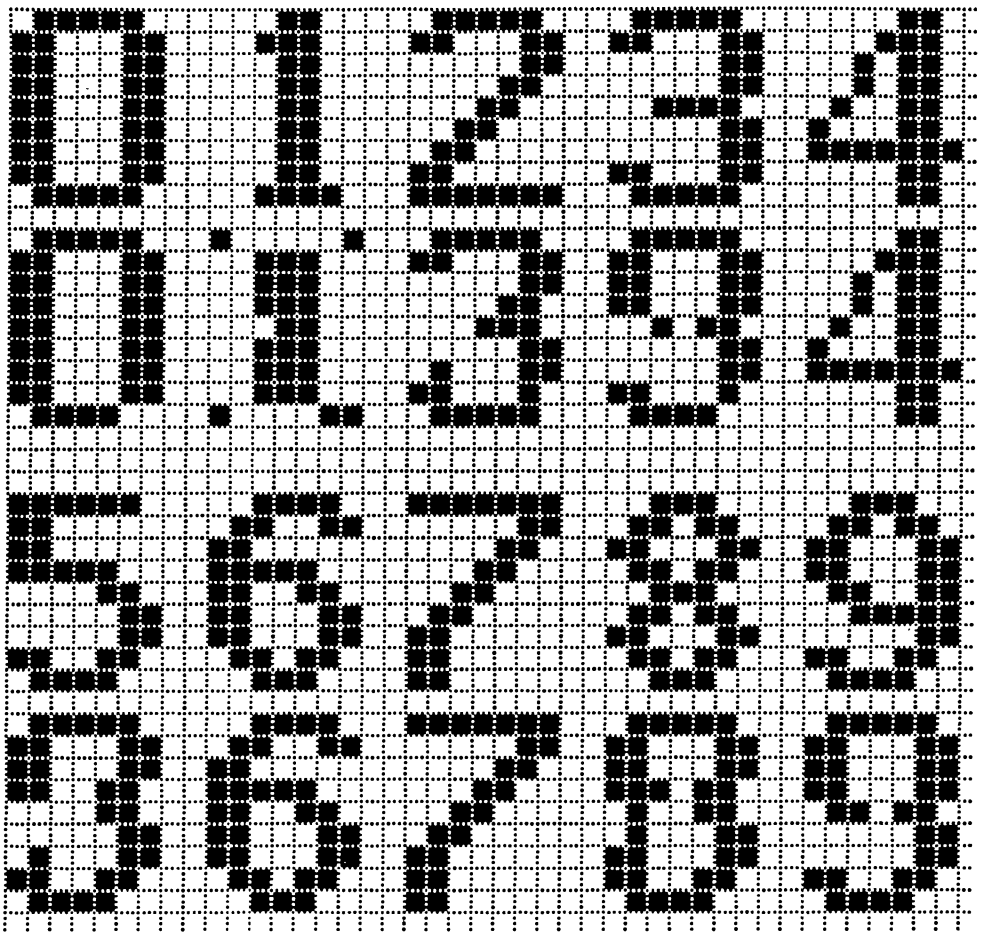

FIG. 6a. *Performance of Hopfield network on pure exemplars.*

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-

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

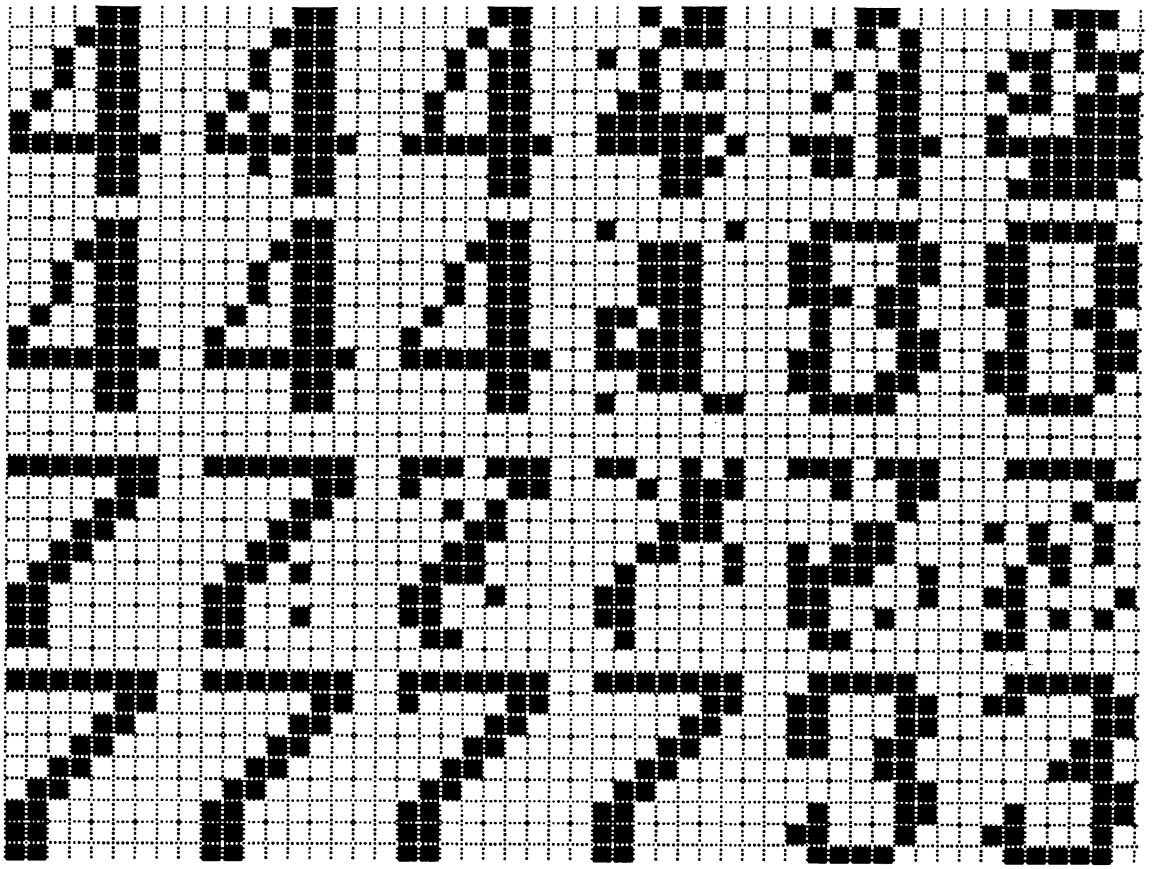

FIG. 6b. *Performance of Hopfield network on error-corrupted input.*

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

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

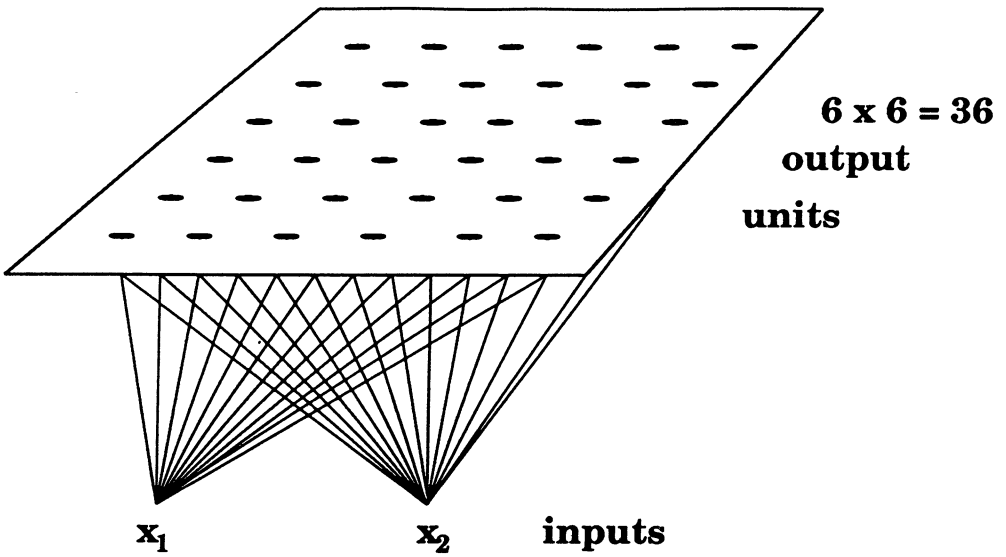

FIG. 7a. *Kohonen network.*

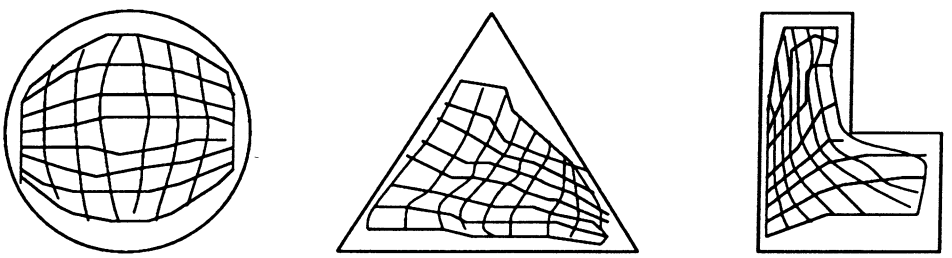

FIG. 7b. *Schematic performance on uniform data on various spaces.*

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.

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

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

$$w_j \to w_j + \Delta w_j, \tag{4}$$

where

$$\Delta w_j = \eta(z-y)x_j = \eta\delta x_j. \tag{5}$$

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^Tw. 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

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

and that the single output y is

$$y = f(\phi(v,w)). \tag{7}$$

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

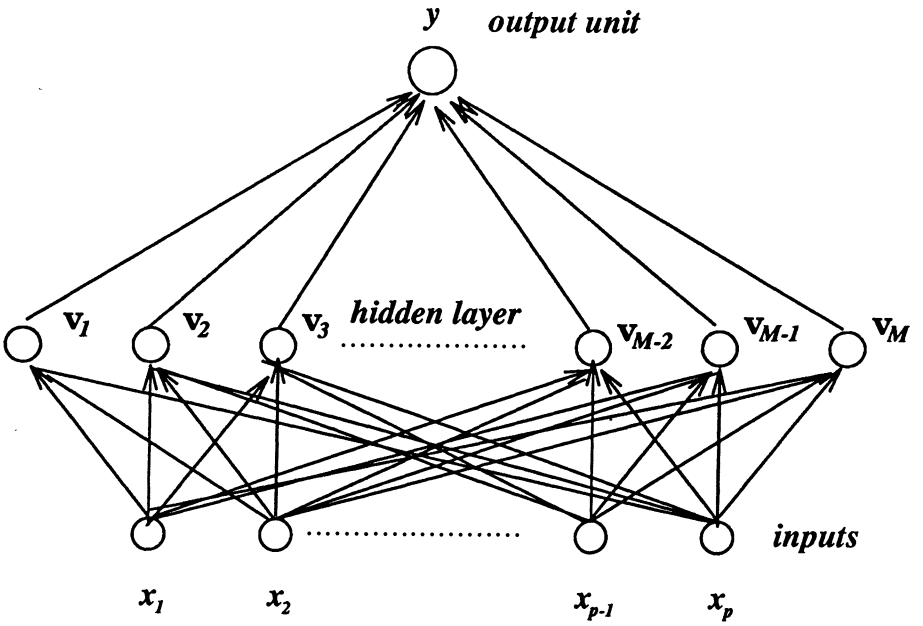

FIG. 8. *A single-output 2-layer perceptron.*

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

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

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

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

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

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,

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

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 $KL(z,y) = \sum_{j=1}^{m} z_j \log(z_j/y_j)$, 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

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

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 (back-propagation, 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

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

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.

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

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.

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

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

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

or possibly involving a mixture of chaos and randomness,

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

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

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

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 back-propagation 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,

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

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

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

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

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

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

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

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

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",

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

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.

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

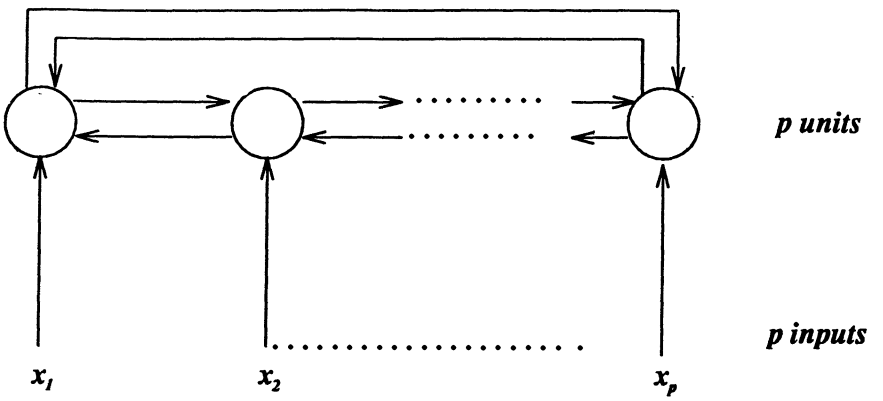

FIG. 9. *Architecture of Hopfield net.*

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

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

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

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

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

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

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 $\bar{x}_n = n^{-1}\sum_{r=1}^{n} x^{(r)}$, 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

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

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.

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

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.

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

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.

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

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,

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms

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.

This content downloaded from 205.208.121.54 on Fri, 08 Jun 2018 18:20:45 UTC
All use subject to http://about.jstor.org/terms