STATISTICAL LEARNING NETWORKS: A UNIFYING VIEW

Andrew R. Barron[1], University of Illinois
Roger L. Barron, Barron Associates, Inc.

Abstract

A variety of network models for empirical inference have been introduced in rudimentary form as models for neurological computation. Motivated in part by these brain models and to a greater extent motivated by the need for general purpose capabilities for empirical estimation and classification, learning network models have been developed and successfully applied to complex engineering problems for at least 25 years. In the statistics community, there is considerable interest in similar models for the inference of high-dimensional relationships. In these methods, functions of many variables are estimated by composing functions of more tractable lower-dimensional forms. In this presentation, we describe the commonality as well as the diversity of the network models introduced in these different settings and point toward some new developments.

1. Introduction

In the context of empirical inference of functions of many variables, a *network* is a function represented by the composition of many basic functions. The basic functions (which are also called elements, units, building blocks, network nodes, or sometimes artificial neurons) are constrained in form: typically nonlinear functions of a few variables or linear functions of many variables. By definition, a *learning network* estimates its function from representative observations of the relevant variables.

Several composition schemes for network functions and corresponding estimation algorithms are reviewed in this paper. Consideration is given to certain networks popular in the neurocomputing field such as perceptrons, madelines, and backpropagation networks. (For a collection of some of the key papers in this field see the volume edited by Anderson and Rosenfeld 1988.) Unfortunately many learning networks are inflexible in the form of the basic functions, inflexible in the connectivity of the network, and lack global optimization of the network function. More consideration is given here to globally optimized networks, networks with adaptively synthesized structure, and networks with nonparametrically estimated units. Particular attention is given to polynomial networks (R.L. Barron et al. 1964, 1975, 1984, Ivakhnenko 1971), projection pursuit (Friedman et al. 1974, 1981, Huber 1985) and transformations of additive models (Stone 1985, Tibshirani 1988). New composition schemes are suggested which combine the positive benefits of the above methods.

Although there are interesting analogies of statistically estimated network functions with the activity of networks of living neurons, we shall not constrain our network functions to be biologically viable models. Instead the focus is on the development of empirical modeling capabilities for network function so as to represent the input/output behavior of a wide range of complex systems for scientific and engineering applications.

Mathematical limitations of high-dimensional estimation are discussed. Bounds from nonparametric statistical theory show that reasonably accurate estimation uniform for all smooth functions (e.g. functions with bounded first partial derivatives) is not possible in high dimensions with practical sample sizes. Network strategies avoid some of the pitfalls of high-dimensionality by searching for structures parameterized by lower dimensional forms. The advantage is that for high-dimensional problems the variance (estimation error) associated with such networks can be much smaller than associated with more traditional approaches. As for the bias (approximation error), the evidence is that for many practically occurring functions accurate network approximations exist, in spite of the theoretical fact that high-dimensional functions can possess sufficiently irregular structure so as to preclude accurate estimation.

Some dynamic network models (such as the Hopfield network 1981) are differential equations (or difference equations) resulting from cycles present in the interconnected network. In this paper we restrict attention to static network models which have no loops in the network. Thus the network is a tree of interconnected functions which implements a single input/output function, which may be adjusted by the empirical estimation process, but otherwise is static.

2. Block Diagrams

We present a hypothetical network to get oriented to some terminology and notation. A function which is defined as a *composition*, such as

$$f(x_1, x_2, x_3, x_4) = g_0(g_1(g_3(x_1, x_2), g_4(x_1, x_3, x_4)), g_2(g_4(x_1, x_3, x_4), g_5(x_4))),$$

may also be written in terms of *intermediate variables*

$$f = g_0(z_1, z_2)$$

$$z_1 = g_1(z_3, z_4),\ z_2 = g_2(z_4, z_5)$$

$$z_3 = g_3(x_1, x_2),\ z_4 = g_4(x_1, x_3, x_4),\ z_5 = g_5(x_4),$$

or it may be drawn as a network diagram (Fig.1):

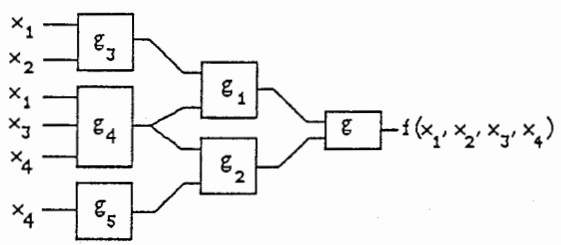

Fig.1. Example Network

The *layers* of a network are the sets of functions which occupy the same depth in the tree.

For a general notation for network functions, in which the indices on a basic function specify the position of the function in the tree relative to the root node, see Lorentz (1966). He called network functions *superposition schemes*. Lorentz made fundamental contributions to the theory of representing functions by compositions which are discussed later in this paper.

Representations for network functions are not unique. For instance, if some of the basic functions are absorbed into the functions to which they are input, then fewer elements are obtained, but the new elements have possibly greater input dimension.

Motivated by the application to modeling human vision, Rosenblatt (1962, ch. 4) called networks with arbitrary

[1] Work supported in part by an Office of Naval Research grant N00014-86-K-0670 and by a National Science Foundation Postdoctoral Research Fellowship.

elemental functions *perceptrons* (although subsequently the term has been used to refer to just one type of network with thresholded linear elements that Rosenblatt extensively studied). Our definition differs slightly from Rosenblatt's in that he allowed transformations to occur on the branches (interconnections) of the network. Such networks are represented in our form either by defining additional single input nodes or by absorbing each such transformation into the node to which the branch is directed.

3. The Building Blocks

For learning networks it is important to choose elements of the network with sufficiently general form that the resulting networks can approximate nearly any function of interest. It is also important to choose these elements with sufficiently small dimension or complexity that they can be accurately estimated. Different approaches to resolving the tension between these two seemingly conflicting objectives result in a variety of different learning network schemes.

Let the function $g(z)$ denote an element of the network, where z is the vector of intermediate variables (outputs from preceding elements or sometimes original input variables) which are input to the given node. The most common forms of elements roughly can be categorized as parametric or nonparametric.

Parametric elements: These are basic functions $g(z, \theta)$ which depend on a vector of unknown parameters. The parametric elements which have been proposed for learning networks usually take one of the following forms:

$$g(z, \theta) = h\left(\sum \theta_k z_k + \theta_0\right) \tag{1}$$

$$g(z, \theta) = \sum \theta_k \varphi_k(z) \tag{2}$$

or, more generally,

$$g(z, \theta) = h\left(\sum \theta_k \varphi_k(z)\right) \tag{3}$$

where φ_k, $k = 1, \ldots, m$, and h are fixed functions. The two most common choices for the φ_k are linear terms (coordinate functions), so that the sum simply implements a linear combination of the inputs as in (1), or polynomial terms of moderate degree. The nonlinear function h is typically chosen to be a nondecreasing function bounded by one (such as a unit step function) -- this is frequently incorporated in networks intended for binary classification. The parameters of each element are estimated from observed data, typically by a least squares or likelihood based criterion. The specific method used to estimate the parameters depends on the probabilistic structure of the data, the network synthesis strategy, and the intended use of the network (see section 4 below).

Nonparametric elements: Some of the element functions $g(z)$ may be regarded as unknown and constrained only in terms of basic smoothness properties (e.g. bounded derivative), or in some cases g is modeled as a stochastic process indexed by z (a Bayes formulation). Such functions are estimated by a smoothing technique such as local linear fits, smoothing splines, variable kernel estimation, truncated trigonometric series, variable degree polynomials, or stochastic process estimation. Typically parameters of the smoothing technique are selected by a criterion such as cross-validation, predicted squared error, or penalized likelihood. With nonparametric elements it is important that the dimension of the z variables be kept to a minimum. (Otherwise the statistical theory indicates that it would be difficult to estimate these element functions.)

Mixed parametric/nonparametric: In this case both types of elements appear in the network. A particularly interesting approach is to combine nonparametric elements, each of which depends only on one variable, with elements which implement linear combinations of many variables. It will be seen that networks of this mixed structure have the potential to approximate any function.

We use the notation $f(\underline{x}, \theta)$ to refer to the complete network function where $\underline{x}$ is the vector of all original input variables and θ is the vector of all parameters which appear in the network.

4. The Structure of the Data and Objective of Network Estimation

In practice, networks are estimated from a training sample of observations of relevant variables. The sample is tpically a sequence of input/output pairs $(\underline{X}_1, Y_1), \ldots, (\underline{X}_n, Y_n)$ where each $\underline{X}$ is a d-dimensional vector. We focus on the case in which the observations are independent, each with the same probability distribution $P_{\underline{X},Y}$. (Certain problems involving data with stationary serial dependencies can also be treated, in which case the relevant distribution is the conditional distribution given the past.) This probability distribution is assumed to depend on an unknown function $f(\underline{x})$: it is this function which neural networks seek to approximate. The assumed nature of this function depends on the objective of the problem (e.g. regression, prediction, classification, density estimation) and the criterion by which performance is measured.

Perhaps the most common use of learning networks is to seek a function $f(\underline{x})$ to minimize the *mean squared error* $E(Y - f(\underline{X}))^2$: that is, the function we wish to estimate is the conditional mean $f(\underline{x}) = E[Y \mid \underline{X} = \underline{x}]$. For problems of *curve fitting, regression,* or *prediction* this conditional mean function has traditionally been the principle object of interest for learning networks. (For certain *time-series* prediction problems the desired function takes on the specific form $f(\underline{x}) = E[Y_t \mid Y_{t-1} = x_1, \ldots, Y_{t-d} = x_d]$). In particular, this framework (associated with a squared error measure of loss) is appropriate when a function $f(x)$ is measured subject to (mean zero) Gaussian error at randomly distributed design points.

For *classification* problems, an optimal discriminant function is one for which the overall *probability of error* is minimized. Most often, learning networks have been utilized to seek an indirect solution to the classification problem by using the mean squared error as the criterion. For two-class classification with $Y \in \{0,1\}$ the conditional mean function reduces to the optimal discriminant $f(\underline{x}) = P[Y = 1 \mid \underline{X} = \underline{x}]$. Nevertheless, it may be more appropriate to seek to estimate the logistic regression function $f(\underline{x}) = \log(P[Y = 1 \mid \underline{x}]/(1 - P[Y = 1 \mid \underline{x}]))$ using likelihood-based criteria. In principle, *probability density estimation* can also be handled using learning networks and a likelihood criterion, in which case f is taken to be the logarithm of the joint density function of the random vector.

The intended use of estimated network functions $\hat{f}$ may dictate probability models and performance objectives other than those indicated above. For instance the object may be to *search* for the extreme points of a function f by using the extreme points of $\hat{f}$. For problems in *vehicle guidance,* the function $\hat{f}$ might estimate parameters of an optimum (two-point boundary-value) guidance law as a function of current and desired final vehicle states (in situations where the optimum f can only be obtained by extensive off-line iteration), in which case the ultimate performance objective is to minimize the final miss distance, rather than to minimize the mean squared error of the parameter estimates. Nevertheless, learning network methodologies have proven successful in some of these contexts (see R. L. Barron and Abbott 1988).

Most network algorithms have been designed for regression or classification with minimum mean squared error as the performance objective, and our attention will be focused primarily on this case.

5. Criteria for Network Estimation and Selection

Here we discuss model selection criteria needed for the estimation of network functions. Without the use of an appropriately penalized performance criterion, an overly complex network may be estimated which accurately fits the training data but will not prove to be accurate on new data.

Predicted squared error: If a network structure $f(x, \theta)$ is fixed and if the total number of parameters k is small compared to the sample size n, then the minimum mean squared error $min_\theta E\,(Y - f(X, \theta))^2$ is approximately achieved by seeking parameter estimates $\hat{\theta}$ that produce the minimum average squared error on the training set, $TSE = \frac{1}{n}\sum_{i=1}^{n} (Y_i - f(X_i, \hat{\theta}))^2$. However, if k is large compared to n, then the model may have small error on the given data, but it is likely to have large error on future data from the same distribution. This phenomenon is partly explained by noting that, under certain conditions (namely that the network depends linearly on the parameters and the true function $f(x)$ happens to be a member of the given k-dimensional family with error variance $\sigma^2 = E\,(Y - f(X))^2$), the mean squared error of an estimated network of fixed dimension k is not equal to the error variance σ^2 but rather is equal to $E\,(Y - f(X, \hat{\theta}))^2 = \sigma^2 + (k/n)\sigma^2$: see Mallows (1973), A.R. Barron (1984). This leads, in view of the fact that under the same conditions $E\,(TSE) = \sigma^2 - (k/n)\sigma^2$, to the predicted squared error PSE criterion as an unbiased estimator of the future performance:

$$PSE = TSE + \frac{2k}{n}\sigma^2. \tag{4}$$

This criterion is very similar to (and in some cases equivalent to) the C_p statistic proposed by Mallows (1973), the generalized cross-validation criterion of Craven and Wahba (1979), the final prediction error of Akaike (1970), and a specialization of the AIC proposed by Akaike (1973). For a recent treatment of these various criteria with emphasis on generalized cross-validation see Eubanks (1988, ch. 2). Calculations similar to those in Akaike (1973) show that PSE continues to be an asymptotically unbiased estimator of the mean squared error $E\,(Y - f(X, \hat{\theta}))^2$ even if $f(x, \theta)$ is not a linear function of θ, provided this function is sufficiently smooth.

Unfortunately, if the network function is selected so as to minimize PSE among a collection of functions of various parameter dimensions, then there is no general guarantee that the resulting minimum PSE will be an accurate estimate of the mean squared error of the estimated function. Indeed, if the true function f is a member of one of the finite-dimensional network families, then the PSE criterion has a tendency to overestimate the dimension (see Atkinson 1980, 1981). On the other hand, the work by Shibata (1984, 1986) shows in related contexts that if the true function $f(x)$ is not exactly representable by any of the finite dimensional models in a sequence $f_k(x, \theta_k)$ *for* $k=1,2,\ldots$ (but can nevertheless be approximated by such models), then selection of $\hat{k}$ by a criterion of the form given above is *optimal* in the sense that the resulting expected squared error $E\,(f(X) - \hat{f}(X))^2$ is asymptotically equivalent to $min_k E\,(f(X) - f_k(X, \hat{\theta}_k))^2$ *as* $n \to \infty$. It is not known if the results of Shibata carry over to the estimation of network functions. Nevertheless, in our experience with numerous practical cases (see Barron et al. 1984), networks selected by minimizing PSE have approximately minimal average squared error on independent sets of test data (in the sense that if the growth of adaptively synthesized networks is halted on an earlier layer or allowed to extend to a larger number of layers, then a significant increase in the average squared error on the test set does not usually occur).

If the error variance σ^2 is not known, an estimate $\hat{\sigma}^2$ can be used in its place in the PSE criterion; however, to avoid overfit care must be taken to avoid having $\hat{\sigma}^2$ much less than σ^2; in particular, $\hat{\sigma}^2$ should not be varied during the process of selecting k (A. R. Barron 1984). We suggest that nearest neighbor regression be used prior to network synthesis to determine a rough estimate of the error variance with the desired properties. To permit consistent estimation of f in the case that it can be exactly represented by a finite dimensional network (as well as in the case that it can be arbitrarily well approximated by networks of sufficient dimensionality) other criteria should be used which place a greater penalty on the dimensionality of the model (e.g. $\frac{k}{n} log\ n$ instead of $\frac{2k}{n}$). Criteria significantly different from PSE will not possess the optimum rate property of Shibata in the context that he considers; however, it is not known to what extent the convergence rate is slowed.

Likelihood based criteria: Suppose the random vectors $(\underline{X}, Y)$ have a conditional probability density function $p(y \mid \underline{x}, f)$ which depends in a known way on the value of f (whereas the true function $f(\underline{x})$ may be unknown). Let $f(\underline{x}, \theta)$ be a given network structure with a k-dimensional parameter θ. Assume that $\hat{\theta}$ is estimated so as to maximize the likelikhood $p(Y^n \mid \underline{X}^n, f(\cdot, \theta)) = \prod_{i=1}^{n} p(Y_i \mid X_i, f(X_i, \theta))$. Define the Akaike information criterion (Akaike 1973) by

$$AIC = -\ log\ p(Y^n \mid \underline{X}^n, f(\cdot, \hat{\theta})) + k \tag{5}$$

and define the minimum description length criterion (Rissanen 1978, 1983) by

$$MDL = -\ log\ p(Y^n \mid \underline{X}^n, f(\cdot, \hat{\theta})) + \frac{k}{2} log\ n. \tag{6}$$

These criteria are used to choose between models of various dimensions. Akaike derived the AIC as an asymptotic bias correction for the estimation of expected entropy loss, in much the same manner that PSE is an asymptotic bias correction for the estimation of expected squared error. Rissanen derived the MDL criterion as the length of a uniquely decodable code for quantizations of the data Y^n given the data X^n (ignoring terms which are asymptotically constant for k bounded). Unlike the optional Shannon code, Rissanen's code does not require knowledge of the function f. Instead, the MDL code uses quantized maximum likelihood estimates of the parameters of the function as a preamble of the code (using $\frac{1}{2}\log$ n bits per parameter). The criterion can also be derived as an asymptotic approximation for the Bayesian test statistics which minimize average probability of error in the selection of the model (see Schwarz, 1978, Clarke and A.R. Barron, 1988).

The validity of the derivations of AIC, MDL, and Bayes criteria require smoothness conditions. In particular the sample Fisher information matrix $\hat{I}$ of second partial derivatives with respect to θ of $-\frac{1}{n} log\ p(Y^n \mid \underline{X}^n, f(\cdot, \theta))$ (evaluated at $\theta = \hat{\theta}$) should be positive definite. A more precise form of the MDL or Bayes criterion uses $\frac{1}{2} log\ det(\hat{I})$ instead of $\frac{k}{2} log\ n$.

For regression with a Gaussian error distribution and known error variance, the AIC reduces to the PSE criterion and MDL reduces to a criterion equivalent to

$$TSE + (\frac{k}{n} \log n)\sigma^2. \qquad (7)$$

For classification problems with $Y \in \{0,1\}$, likelihood based criteria are defined by using the Bernoulli model $p(y \mid \underline{x}, f) = (f(\underline{x}))^y (1 - f(\underline{x}))^{1-y}$ (in which case care must be taken to use networks with $0 < f(\underline{x}) < 1$). The equally general logistic model $p(y \mid \underline{x}, f) = e^{yf(\underline{x})}/(1 + e^{f(\underline{x})})$ may be preferred for classification problems, since it forces satisfaction of the probability constraints $0 < p < 1$ without constraining the function f. For logistic regression the minus log-likelihood takes the form $\sum \log(1 + e^{f(\underline{X}_i, \theta)}) - \sum Y_i f(\underline{X}_i, \theta)$, which is minimized (e.g. by Newton's method in the context of various synthesis strategies) and then penalized by k or $\frac{1}{2} \log n$ as appropriate for the desired criterion.

Complexity regularization: In A.R. Barron (1985) the minimum description length criterion is extended to nonparametric contexts in which the description length need not reduce to the form of (6). Consistency results are obtained in A.R. Barron (1985, 1987) which show convergence ($as\ n \to \infty$) of distributions estimated by the complexity regularization. The specialization of the convergence results to the case of estimation of network functions is given in the Appendix.

6. Main Strategies for Network Synthesis

There are two main strategies for the synthesis of networks depending on whether the structure of the network is fixed or allowed to evolve during the synthesis process.

Fixed networks: In this approach a fixed composition structure (often relatively large) is preselected with the hope that the desired function can be accurately approximated by networks of the selected form. The problem of choosing parameters of the network so as to optimize a performance criterion may be regarded as a *global search of a highly multimodal surface.* In general, global convergence is difficult to guarantee; nevertheless, by choosing a network function which depends smoothly on the parameters it is often feasible to estimate sufficiently accurate network functions by certain global search techniques (e.g. techniques which alternate global random and local gradient search). Other methods for estimating network functions attempt to localize the search within each unit of the network by defining target values for each elemental function. More specifics are given in section 7 below.

The advantage of the fixed network approach is that certain structures are known to have the ability to approximate any continuous function (see section 13). However, for moderate sample sizes, these fixed structures may have too large a parameter dimension for the least squares or maximum likelihood estimators to be accurate. In this case, to prevent irregularity of the estimated function, it is useful to constrain the parameters so that the resulting network function is smooth or to penalize the performance criterion by incorporating a term for the lack of smoothness (e.g. the sums of squares of first partial derivatives of the network functions at the observations). Of course the criteria mentioned in section 5 above are not adequate when the dimension of the network is fixed in advance.

Adaptive networks: In this approach, the attempt is to estimate networks of the right size with a structure evolved during the estimation process to provide a parsimonious model for the particular desired function. Typically, the network is estimated one layer at a time, with the elements on each given layer selected to minimize the predicted squared error or complexity regularization criterion. The basic idea is that once the elements on a lower level are estimated, and the corresponding intermediate outputs z are computed, then the parameters in a given element $g(z, \theta)$ may be estimated by usual least squares or likelihood maximization techniques. It is most common for the elements on each layer to be greedily trained to attempt to best estimate the desired final output, even though the outputs of these elements are combined on succeeding layers. On the other hand, some methods developed in statistics select the element functions so as to work best in linear combination with the previously selected elements on a given layer.

Practical experience shows clear advantages of the adaptively synthesized networks over some of the globally optimized fixed network structures. (However, certain theoretically appropriate fixed structures have yet to be tried in practice; also, the smoothness penalty criteria have yet to be utilized with the larger fixed networks.) In most instances the adaptively synthesized networks are more parsimonious. Parts of the network which are inappropriate or extraneous for statistically modeling the given data are automatically not included in the final network. The drawback of the adaptive strategies is that they cannot be guaranteed to work. It is possible to find counterexamples of data corresponding to functions which are exactly modeled by a two-layer network, but no non-trivial first layer elements are selected by a given adaptive synthesis strategy.

Mixed adaptive/global strategies: After the best elements on each layer are computed, a numeric search can be used to update the estimates of parameters for ancestral nodes on earlier layers. An iterative scheme that alternates between estimation of the parameters of the given element and the estimation of the parameters of the ancestral nodes is suggested by the projection pursuit algorithm and its generalizations (see sections 10 and 12).

7. Some Early Network Developments

While linear models for regression and thresholded linear models for classification (e.g. of the form (1), (2), or (3)) have been long used in statistical practice (with the beginnings of the modern understanding due in large part to R.A. Fisher (1922, 1934, 1936) who introduced measures of statistical efficiency, explained the efficiency of maximum likelihood estimation, and derived the linear discriminant function for multivariate Gaussian classification), these same linear models were reintroduced (unfortunately with comparatively inefficient estimators) in the 1950's and 1960's as a basic ingredient in learning network models. The new and interesting twist was that more general classes of functions were modeled by combining these simpler models into a network. Here we mention some of the development which occurred in this period.

The forerunners in the network modeling field were McCulloch and Pitts (1943), who introduced the thresholded linear function as a model for the behavior of a neuron and, in that paper, analyzed the model not so much for its biological viability, which was discussed only briefly, but rather (in the language of theoretical computer science) as a basic computational unit with the property that any predicate with finite domain could be implemented by a network of such units.

There was a surge of interest in methods for the inference of networks (Hebb 1949, Ashby 1952, Farley and Clark 1954, Minsky 1954, von Neumann 1956, Rosenblatt 1957, Lee and Gilstrap 1960) culminating in some interesting and successful multiple layer estimation methods in the early 1960's due to Rosenblatt (1962), Widrow et al. (1960, 1962, see also 1987), and R.L. Barron et al. (1964, see Moddes et al. 1965, Gilstrap 1971, Barron et al. 1984). Although some of the networks due to Rosenblatt and Barron et al. used more general elemental functions than the original thresholded linear function, they did share the form (3) (transformed variables were combined linearly using free parameters). These heuristic multi-layer

methods were not well understood theoretically and (with the exception of Rosenblatt's book) they were not widely disseminated at that time. We emphasize that contrary to the popularly held current belief (initiated in the book by Minsky and Papert 1969 and perpetuated by statements as in Rumelhart et al. 1986, p.321), powerful rules were found for the estimation of multiple layer networks.

The methods of Widrow et al. and Rosenblatt for binary classification possessed many similarities. In particular, both authors exclusively utilized recursive estimation strategies in which the parameter estimates are updated with each new observation by an error correction procedure analogous to the Robbins-Monroe (1951) stochastic approximation (but without the full statistical efficiency known to hold for recursive least squares or recursive implementations of maximum likelihood). Moreover, both approaches were amenable to clear theoretical proofs of convergence properties in the case of single element networks (these results are well-explained in Nilsson (1965) and Duda and Hart (1973)). Widrow used a stochastic gradient method which he called the least mean squares (LMS) algorithm. Rosenblatt used a method (related to relaxation procedures for solving linear inequalities, Agmon 1954), which he called the perceptron algorithm: it finds a hyperplane which perfectly separates the two classes whenever the classes are linearly separable. The non-convergent behavior in the non-separable case was analyzed by Efron (1964).

For multiple layer networks the method of Widrow et al. (1960, 1962) was only explained in the case that first layer elements are adjustable and the succeeding layers are preselected. Widrow used iterations of his strategy to handle also the more general estimation problem, but this approach was not published until Widrow 1987, to which we refer the reader for a description.

For two and three layer networks of thresholded linear elements, Rosenblatt (1962, ch. 13) developed an algorithm which he called *back-propagating error correction* (unfortunately, this name recently has been reused for another algorithm for network estimation, as mentioned below). The objective of his method is recursively to estimate desired outputs for every element as well as to estimate the parameters. Naturally, given a desired output of an element Rosenblatt updates the parameter estimates in the element by his perceptron algorithm (here a parameter update occurs only if the actual output differs from the desired output). On the other hand, if the output of an element does match the desired value, then depending on whether the resulting final output of the network is in error, the desired intermediate variable is adjusted to reduce this error (again as in the perceptron algorithm but with the role of parameters and variables reversed). (Randomization is used to avoid certain degeneracies. In particular, with each step no update action is taken with probability $0<p<1$.) Rosenblatt advocated cycling through the data and the elements of the network in such a way that each combination (of datum and network element) potentially would be considered infinitely often. He presented a theorem (Rosenblatt, p. 294) to the effect that if the data are separable by the network (i.e. there exist parameter values for which the network function correctly classifies every point), then his estimation strategy will find such an error-free solution in a finite number of steps (with probability one).

The approach developed by R.L. Barron et al. (1964) and further explained in Moddes et al. (1965), Gilstrap (1971), and Barron et al. (1984) solved the multilayer network estimation problem by global search to minimize the sum of squared errors $\Sigma(Y_i - f(X_i, \theta))^2$. Barron et al. introduced an algorithm called *guided accelerated random search* (GARS) which alternated between global random search (using a spherical normal distribution centered at the current best point) and local gradient search (for which convergence was accelerated by a halving/doubling algorithm for the step size and by adjusting a variable subset of the parameters at the different steps). The particular elemental functions originally used by R.L. Barron et al. were quadratic functions in two variables $g(z,\theta) = \theta_0 + \theta_1 z_1 + \theta_2 z_2 + \theta_3 z_1 z_2$. A spirally-connected network with 24 input variables and seven layers was constructed (see fig. 2). Using 25-50 observations of simulated reentry vehicle positions during a given time frame $(t, t-\Delta t, \ldots, t-7\Delta t)$, networks were constructed to predict the final position and impact time of the vehicle. The parameters of the networks were constrained to values in the interval between -1 and +1. The GARS search routine converged to essentially the same extremum of performance for each of many randomly selected initial parameter vectors, suggesting that a non-unique global optimum was reached. Performance on an independent test set of observations suggested that despite the complexity of the network, and the small sample size, the estimated function was not overfit to training data. (However, overfit problems were later experienced with these large fixed networks on some industrial process modeling problems -- these experiences led in the early 1970s to the adoption of adaptive synthesis strategies discussed below.)

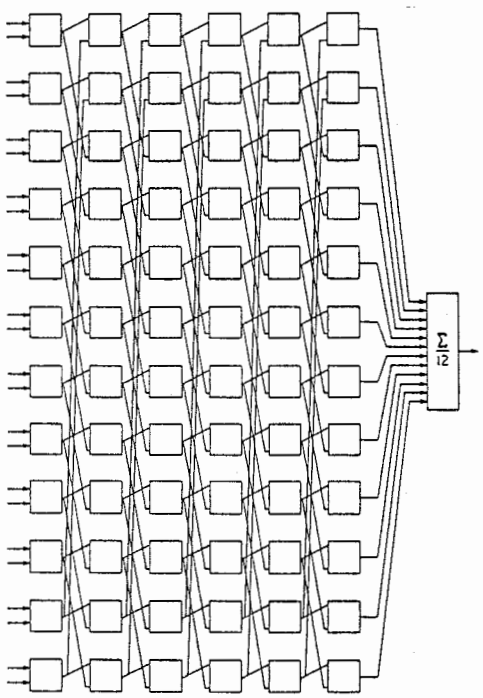

Fig. 2. Uniform Spiral 72-Element Network

The network of fig. 2, which consists of quadratic two-input elements, represents a family of sixth-degree polynomials. Since the network contains a total of 288 parameters, this family is a relatively low-dimensional manifold in the complete (593,775 dimensional!) family of sixth-degree polynomials in 24 variables. Nevertheless, the network had more than enough flexibility to yield accurate approximations for the specific application to re-entry vehicle trajectory predictions.

8. The Current Fashion

In recent work Rumelhart, Hinton, and Williams (in Rumelhart et al. 1986, ch. 8) propose that an implementation of the gradient descent algorithm be used to attempt to

minimize the sum of squared error for multiple layer feedforward networks. They use element functions of the form (1) with h equal to a logistic function: this choice is viewed as a smoothing of the step function to obtain a differentiable function of the parameters. Since the network is a composition of functions, the derivatives required for the gradient method are determined by the chain rule of calculus (starting at the final node and propagating back to the parameters in the first layer). Although it is recognized that the gradient method may be inappropriate in general for highly multi-modal surfaces, Rumelhart et al. found that it worked adequately on the simple examples that they considered. Hinton and Sejnowski (in Rumelhart et al. 1986, ch. 7) propose that a sequential random search algorithm (simulated annealing) be used to estimate the parameters of a Hopfield style network; they call their learning network a Boltzmann machine. These papers (see Rumelhart et al. p. 321) give the impression that multilayer search strategies for networks are novel to the 1980s. Clearly this is false in view of the methods we have discussed. In our experience (beginning in the 1960s) a combination of random and derivative-based search strategies, as in the GARS algorithm, is an effective technique for globally optimizing networks. In any event, much of the recent work (as in Rumelhart et al.) has ignored the developments in the 1970s and 1980s of the adaptive network strategies and the nonparametric statistical methodologies for specific network structures.

9. Networks with Adaptively Synthesized Structure

With the propensity of large fixed networks to result in overfit estimates, attention was turned in the 1970s to networks for which the structure is adaptively determined from the data. Such network strategies were introduced by Ivakhnenko (1971) and their development in the U.S. is traced in Barron et al. (1974, 1975, 1984, 1987).

The elements extensively utilized in these adaptively synthesized networks are second- and third-order polynomial functions in two variables. (One and three variable elements are also used in recent implementations.) For the method to work , the number of inputs of each element must be restricted so as to avoid a combinatorial explosion in the number of possibilities that the algorithm must check.

In brief, the basic strategy (using elements involving two variables) is depicted in fig. 3. On the first layer, all possible pairs of the inputs are considered and the best k_1 are temporarily saved. On the succeeding layers, all possible pairs of the intermediate variables z from the preceeding layer(s) are considered and the best k_2 (k_3, etc.) are saved. Finally, when additional layers provide no more improvement, the network synthesis stops. The final network consists only of the ancestors of the final element.

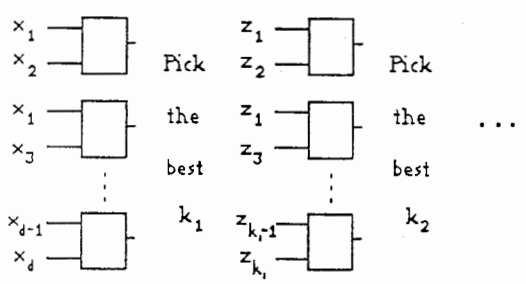

Fig. 3. An Adaptive Network Synthesis Strategy

In the original Ivakhnenko algorithm, the parameters within each element were estimated so as to minimize on a training set of observations the sum of squared errors of the fit of the element to the final desired output. Cross-validation on a separate testing set was used to rank and select the best elements on each layer and to select the number of layers. (Ivakhnenko called this division of the data into sets with different purposes in network estimation the *group method of data handling*, GMDH.) The need to construct complete quadratic polynomials for every pair of variables forced early implementations of the algorithm to restrict the number k of temporarily saved intermediate variables to be typically not more than 16.

Later algorithms developed by A.R. Barron (1979-1982, Polynomial Network Training Routine, PNETTR III and IV, Adaptronics, Inc.) incorporated a predicted squared error PSE criterion (related to the criteria of Akaike and Mallows as discussed above) at every phase of element selection in the network. Moreover, a method was developed whereby candidate pairs are prescreened before each layer (according to their predicted error in linear combination) thereby permitting more elements to be considered on each layer (typically k is between 30 and 60). This also permitted more complicated element calculations, i.e. third-degree polynomials with subset selection by the PSE criterion. Also the saved elements from all preceeding layers are candidate inputs to a given layer. Moreover, some one- and three-input elements are considered on each layer. The PNETTR algorithm was extensively applied to problems in nondestructive evaluation of materials, modeling of material characteristics, flight guidance and control, target recognition, intrusion detection systems, and scene classification; see Barron et al. (1984) and the references cited there. For an application of an earlier version of the algorithm to weather forecasting see A.R. Barron et al. (1977).

The more recently developed algorithm by J.F. Elder IV (1985-present, Algorithm for Synthesis of Polynomial Networks, ASPN, Barron Associates, Inc.) permits a choice of a minimum complexity or predicted squared error criterion. This algorithm has more user flexibility in the choice of one-, two-, or three-input elements and in the form of the polynomial elements (e.g. the degree may be adjusted within certain limits). Moreover, at each layer a new element is considered which is a linear combination of all elements on the preceeding layer.

Currently, a major applications thrust is use of adaptively-synthesized polynomial networks to initialize and/or re-initialize (in real time) two-point boundary-value guidance solutions for flight vehicles (R.L. Barron and Abbott 1988). Polynomial networks are trained off-line on a library of simulated optimum trajectories and interrogated on-line with information about existing and desired vehicle states. Interrogation yields numerical values of six initializing adjoint variables (Lagrange multipliers) in a calculus of variations formulation of the trajectory optimization solution. Because each new interrogation answers the *optimum-path-to-go* question, a guided trajectory need not be restored, when disturbed, to a preconceived nominal path, and optimality of trajectory energy management and accuracy of guidance are not compromised by disturbances within maneuvering limits of the vehicle. In the two-point-boundary-value guidance application, the role of the polynomial network is to compress a large library of multivariate trajectory information and render it in a form (the network) suitable for virtually instantaneous look-up and interpolation.

Fig. 4 is a diagram for networks trained to estimate two of the initializing adjoint variables for a specific flight vehicle guidance application. These networks were synthesized from a data base of 435 observations of the candidate variables. Ten variables were selected by ASPN for inclusion in the final model. The information presented in each box refers respectively to the index of the element (in the list of elements saved by ASPN during synthesis), the type of element (in terms of number of inputs), and the number of terms in each cubic expression after pruning according to a PSE

criterion. The "white" element computes a linear combination of its inputs.

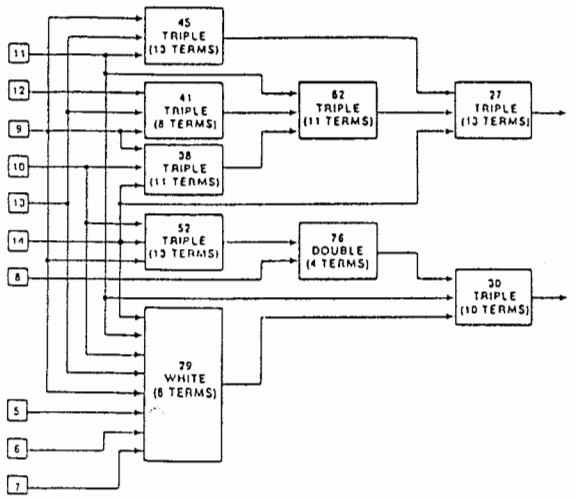

Fig. 4 An Adaptively Synthesized Polynomial Network

10. Projection Pursuit

The projection pursuit algorithm of Friedman et al. (1974,1981,1984) which is so popular in statistical circles has not previously been discussed in the context of learning networks. This algorithm adaptively synthesizes a three-layer network in the form of fig.5. The first-layer functions implement linear combinations $\Sigma\theta_{jk}x_j$ for ordinary projection pursuit (or $\Sigma\theta_{jk}\phi_{jk}(x)$ for a generalization of projection pursuit to be discussed below). The second-layer functions $g_k(z)$ are nonparametrically estimated functions of one variable. Finally, the third layer simply takes a linear combination $\Sigma\,\beta_k g_k$. Thus the function implemented is $f(x, \theta, \beta) = \sum_k \beta_k g_k(\sum_j \theta_{jk} x_k)$.

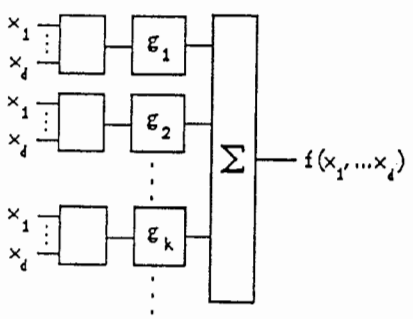

Fig. 5. Network Diagram for Projection Pursuit

The estimation strategy of projection pursuit proceeds vertically through the levels indicated in fig.5. On each level, an iterative Gauss-Newton algorithm is employed which alternates between estimation of the parameters θ from the first layer and the function g_k from the second layer so that in linear combination with the preceeding levels the fit is optimized (using the sum of squared errors or a likelihood criterion). Here the use of the optimized linear combination $\Sigma\beta_k g_k$ is a relaxation method suggested by Lee Jones (1986) as an improvement over the original method (which estimates g_k to fit the error $y - (g_1 + \ldots + g_{k-1})$).

To estimate the functions $g(z)$, Friedman et al. utilize a nonparametric smoothing technique involving locally linear functions (the linear fit at an arbitrary point z is estimated using the data in a neighborhood of that point). Nevertheless, the methodology also works with other one-dimensional nonparametric estimation techniques such as smoothing splines or variable degree polynomials.

Projection pursuit provides an excellent example of a learning network with both parametrically and nonparametrically estimated elements. Also, it demonstrates an effective iterative strategy for estimating the elements of a layer of a network to work well in combination with each other rather than in isolation.

An advantage of projection pursuit networks is that they have been amenable to theoretical examination of some of their approximation properties (Huber 1985, Donoho and Johnstone 1985, Jones 1987), although much work remains to be done in this direction. In particular it is known that any square integrable function can be approximated by a theoretical analog of projection pursuit, provided sufficiently many (vertical) levels of the network are utilized; however, the analogous result for data-driven estimation has yet to be established.

11. Additive Models and Transformations

Additive models represent functions of the form $\Sigma g_k(x_k)$, where in general the one-dimensional functions g_k are unconstrained and in practice usually are estimated non-parametrically. (In contrast, linear models estimate only the coefficients of linear combinations of fixed functions.) The theory for the estimation of additive functions is developed in Stone (1985). In particular, Stone demonstrates the surprising result that, unlike general functions of d variables, additive functions can be estimated with a convergence rate for the expected squared error which is as good as the rate which can be obtained for the estimation of one-dimensional functions ($n^{-2r/(2r+1)}$ instead of $n^{-2r/(2r+d)}$ where n is the sample size, r is the assumed order of smoothness, and d is the dimension; see section 14 below). Moreover, Stone showed that although not every function is additive, a best additive approximation to a function exists and can be estimated at the indicated rate. Stone's approach to estimating the additive functions is to use finite dimensional linear spaces of functions (such as splines, polynomials, or truncated trigonometric series – in particular Stone uses splines), so that the resulting additive approximation is then written in terms of a linear function of many fixed basis functions, in which case traditional least squares projection becomes applicable.

Winsberg and Ramsay (1980) and Tibshirani (1988) generalize additive approximation by permitting monotone transformations $h(y)$ of the dependent variable. By inverting this transformation, an approximation to the dependent variable is obtained in the form depicted in fig. 6 with $g=h^{-1}$. A related model is in Breiman and Friedman (1985) where noninvertible transformations h are permitted.

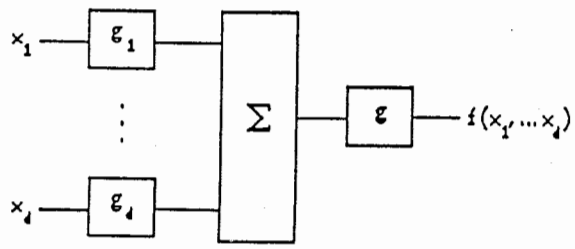

Fig. 6. Network for Transformations of Additive Models

Networks as in fig. 6 can be estimated by alternating between estimates of the transformation g and the first layer

functions g_k using methods similar to projection pursuit. In particular, suppose finite series approximations are used for each of the functions g_k. Given a current estimate of g (which is assumed to be a differentiable function), a Gauss-Newton type algorithm can be used for the estimation of the coefficients in a finite series approximation of the g_k. Then, given the current g_k, the new estimate of g can be obtained by any of several nonparametric methods (e.g. least squares projection onto a linear space of approximating functions, local linear smoothing, etc.). These steps are then iterated until only negligible improvement in the optimization criterion is observed.

Our purpose for mentioning additive models in the context of networks is that this structure is the one which is best understood theoretically (except perhaps for linear discriminate functions and linear regressions which have even less approximation capabilities) and, moreover, the additive structure is a basic building block for more elaborate networks which show some promise. Although additive models cannot represent interactions between variables, interactions can be obtained by taking sums of transformations of additive models as seen below.

12. Generalizations

It appears to us that certain extensions to the network forms of projection pursuit or transformations of additive functions lead naturally to a particular network structure which is known to have powerful approximation capabilities. The statistical estimation strategies associated with projection pursuit and additive models then lead to estimation strategies for these more complex network forms.

In particular, consider networks of the form given in fig. 7. This form may be regarded as a projection pursuit network, generalized to allow transformations of the original variables on the first layer. Using series approximations (e.g. polynomials) for these transformations, the projection pursuit estimation algorithm becomes applicable to this network as discussed in section 10. Alternatively, the network of fig. 7 may be thought of as a composition of additive functions. Specifically, the network consists of $2d+1$ additive functions with outputs $z_1, z_2, \ldots, z_{2d+1}$, say, which become the inputs to a final additive function with output f. Whereas none of the lower layer additive portions of the network can approximate every function, the composition of these functions can approximate any continuous function as discussed in section 13 below. In principle, any of the methods for estimating transformations of additive models can be used to estimate the k'th such function by fitting the model to the error resulting from the sum of the previous $k-1$ models. However, such iterative approximations may require more than the $2d+1$ levels indicated by the theory.

A specific implementation of a generalized projection pursuit algorithm which incorporates some of the features mentioned above is being developed by A.R. Barron and Gayle Nygaard. It will permit the use of polynomial, spline, or trigonometric series approximations for any of the transformations of the network. A new feature of this algorithm is that, when estimating g_k in fig. 7, the transformations $g_1, g_2, \ldots, g_{k-1}$ are backfitted to provide the best additive combination by projecting to sums of basis functions in the manner of Stone (1985). Moreover, after each transformation is estimated, a backward stepwise rule (using a penalized squared error or complexity criterion) is used to prune unnecessary terms from each element. In view of the relatively large (but fixed) size of the network structure, this pruning of the number of coefficients is essential to avoid overfit with moderate sample sizes. The most important generalization is to permit nonparametrically estimated transformations of the variables so as to achieve "projections" to surfaces more general than the hyperplanes utilized in traditional projection pursuit. It is then expected that fewer numbers of projections are required (perhaps as few as $2d+1$).

13. Mathematical Foundations

Consider continuous functions $f(x_1, \ldots, x_d)$ of d variables on a bounded set such as the unit cube $[0,1]^d$. Upon reflection it appears that all familiar functions of three or more variables are built up from the composition of various functions of one or two variables. (For instance a sum of d variables is a composition of $d-1$ bivariate sums.) Accustomed to the traps of mathematical analysis, one might speculate that there exist truly d-dimensional functions that cannot be represented in this way. On the contrary, Kolmogorov (1957), see also Lorentz (1966), proved the surprising result that every continuous function on $[0,1]^d$ can be exactly represented as a composition of sums and continuous one-dimensional functions.

Lorentz (1966) identified a particular composition scheme (depicted in fig. 2) which works for all functions of a given dimension. For any continuous function f on $[0,1]^d$, there exist continuous one-dimensional functions g_j and h_{jk} for $j=1, 2, \ldots, 2d+1$ and $k=1, 2, \ldots, d$ such that

$$f(x_1, \ldots, x_d) = \sum_j g_j \sum_k h_{jk}(x_k) \qquad (8)$$

Moreover, Lorentz demonstrated the existence of universal functions h_{jk} which do not depend on the function f (whereas the g_j do depend on f). In his proof, Lorentz constructs piecewise linear functions $g_j^{(\varepsilon)}$ with the property that for every x in the cube the *majority* (i.e. at least $d+1$) of the values $g_j^{(\varepsilon)}(\Sigma h_{jk}(x))$ (for $j=1, \ldots, 2d+1$) are within ε of $f(x)$. (This proof suggests that it might be more natural to use the median of $g_1(\Sigma h_{1k}), \ldots, g_{2d+1}(\Sigma h_{2d+1,k})$ instead of the sum to approximate f.) The proof of the existence of an exact representation involves a careful limiting argument with $\varepsilon \to 0$.

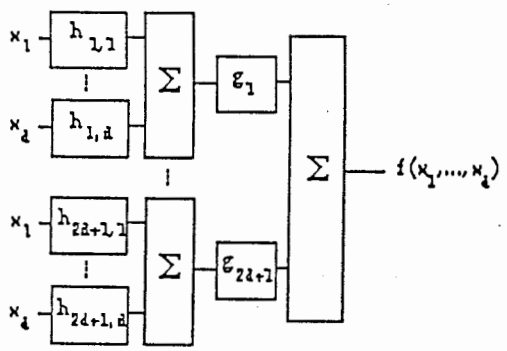

Fig. 7. Kolmogorov-Lorentz Network

In general the functions g_j for which the representation is valid may be rather irregular (e.g. nondifferentiable). It is reasonable to expect, that for sufficiently regular functions f, relatively smooth elements g_j and h_{jk} can be used in the representation, especially if the h_{jk} are allowed to depend on f.

One way to quantify the smoothness of a function is the characteristic s. A function of d variables has characteristic $s = p/d$, where $p = r + \alpha$ if all derivatives of order r are Lipshitz continuous of order α where $0 < \alpha \leq 1$ (this is the case with $\alpha = 1$, $r = p - 1$ if the derivatives of order p are bounded). (This smoothness characteristic is used by Stone (1982) to obtain minimax rates of convergence of nonparametric estimators, see below.) Kolmogorov (1959), see also Lorentz (1966), proved that *not every function with a given smoothness characteristic can be represented as a composition of functions*

having a larger smoothness characteristic. This means, for instance, that there exist functions of ten variables which are differentiable up to order ten that cannot be represented by compositions using one-dimensional functions having more than one derivative.

The limitations expressed by these theoretical results do not preclude the possibility that many of the practically occurring functions which one might wish to estimate are representable in terms of low-dimensional functions of large smoothness characteristic. For instance, it might be true that infinitely differentiable functions can be represented in terms of compositions of infinitely differentiable functions of low dimensionality.

The appeal of the Kolmogorov-Lorentz representation compared to other familiar network structures is the economy of network nodes. A *fixed* number of one-dimensional continuous functions (namely $(d+1)(2d+1)$) suffices to give an approximation or even an exact representation.

Other network structures are known to possess approximation capabilities, but generally the number of network nodes depends on the function being approximated and the desired accuracy. Subsequent to our *Interface* presentation, George Cybenko informed us of some of his recent results (Cybenko 1988). Consider three-layer networks in which the element in the final layer takes a linear combination of its inputs and the first two layers are restricted to elements in the form of equation (1), each of which uses the same nonlinear transformation h. This function h is permitted to be any fixed continuous strictly increasing function with bounded range. Cybenko proved that *for any continuous function f on a d-dimensional cube and any $\varepsilon > 0$, there exists a three-layer network with elements of the form (1) that approximates f with error uniformly less than ε.* His proof is to show that the first two layers of the network may be used to implement kernel functions ("approximations to the identity") of appropriate bandwidths having arbitrary centers, from which the result follows by taking an appropriate linear combination. Cybenko also points out that two-layer networks are sufficient if quadratic ϕ functions are used in first layer elements of the form (3), for then certain kernel functions may be constructed by taking linear combinations of these elements. Although Cybenko does not refer to the rich collection of statistical literature on kernel approximation (see the books by Prakasa Rao 1983, Devroye 1987, or Eubanks 1988), it is apparent that results in this area could be utilized to bound the number of kernels (and hence the number of nodes in Cybenko's networks) required to achieve a given accuracy.

Some basic results in mathematical analysis which have impact on the approximation capabilities of network forms should not be overlooked. The Weierstrass theorem and its generalization to multivariate functions asserts that any continuous function on $[0,1]^d$ can be uniformly approximated by a sufficiently large degree polynomial. The polynomial approximations need not be restricted to the canonical sum of products form $\Sigma \theta_k x_1^{k1} \ldots x_d^{kd}$ (which is itself a large network of simple structure), indeed, the multivariate generalization of Weierstrass's theorem is seen to be an immediate corollary to the Kolmogorov-Lorentz representation theorem.

Other multivariate forms are known to approximate arbitrary continuous functions. For instance, finite trigonometric sums $\Sigma_k (\alpha_k cos(\pi k \cdot x) + \beta_k sin(\pi k \cdot x))$ can uniformly approximate any continuous function on $[0,1]^d$, provided the function is continously extended to satisfy boundary conditions on $[-1,1]^d$ (see Lorentz 1966, p.87). Here $k = (k_1, \ldots, k_d)$ and $k \cdot x = \Sigma_j k_j x_j$. We remark that the sin and cos functions have bounded variation, so they can be represented as the difference of monotone functions h. Consequently, the trigonometric sum is a two-layer network with first layer elements having the form (1). This gives a simple proof of Cybenko's theorem specialized to such h.

The Jackson theorems express bounds on the accuracy of a polynomial or trigonometric approximation in terms of the assumed smoothness of the function being approximated. (See Jackson 1930 for a lucid treatment of the univariate case and Lorentz 1966, especially pp. 87-90, for multivariate extensions.) For instance, if a function f has partial derivatives $\partial^r f / \partial x_k^r$ of order $r \geq 0$ which are Lipshitz of order $0 < \alpha \leq 1$, then there is a constant c such that for every $N \geq 1$ a polynomial approximation of degree N (in each coordinate) exists with error uniformly less than cN^{-p}, where $p = r + \alpha$. Unfortunately, Jackson type theorems are not known for polynomial approximations which take a network form other than a sum of products.

14. Some Limitations on the Statistical Accuracy of Learning Networks

In practice, learning network approximations are not obtained from completely known functions, but rather they are estimated from a training sample of observations of relevant variables. The sample is typically a sequence of input/output pairs $X_1, Y_1, \ldots, X_n, Y_n$ which is assumed to possess one of several possible probabilistic structures as discussed previously. There is a fundamental question which is addressed for this class of problems: *What is the relationship between the achievable accuracy and the size n of the sample?* Typically it is found that the answer depends on the class of possible functions. Especially critical are the dimension d and the regularity of the function. Results from approximation theory play a key role in these statistical considerations. The presently known answers, which we discuss below, are somewhat discouraging, especially with regard to practical contraints imposed on the dimensionality. To understand better and to avoid the pitfalls of high dimensionality, it is suggested that new approximation theory and estimation results are needed for specific network composition strategies.

Stone (1982) has fundamental results concerning a class of nonparametric estimation problems which includes curve or surface fitting with normally distributed errors and binary classification with unknown conditional class probability functions. Attention is restricted to functions on a bounded set with a given smoothness characteristic $s = p/d$ (in the sense that all cross partial derivatives of total order r are Lipshitz of order α and $p = r + \alpha$ as above). Stone establishes that *the optimal rate of convergence is $\varepsilon_n = n^{-s/(2s+1)}$* for the L^q norms *$(0<q<\infty)$ and $\varepsilon_n = (n^{-1} \log n)^{s/(2s+1)}$* for the L^∞ norm. This means that there exist estimators $\hat{f}_n$ (depending only on the sample) such that the ratio $\| f - \hat{f}_n \| / \varepsilon_n$ is bounded in probability for all functions f of the given smoothness class. Conversely, for any sequence of estimators $\hat{f}_n$ there exist sequences of functions f of the given smoothness class for which the ratio $\| f - \hat{f}_n \| / \varepsilon_n$ is bounded away from zero in probability, as $n \to \infty$. To achieve the optimal rate of convergence, Stone (1982) uses *local polynomial regression.* The value of the estimator $\hat{f}_n(x)$ at a point x is obtained by a weighted least squares polynomial fit using all data points for which the distance from x is less than δ_n. Stone chooses the sequence δ_n to converge to zero at rate $n^{-1/(2p+d)}$ and he chooses the local polynomials to have total degree r.

For convergence of the mean integrated squared error (MISE) uniformly over all functions which have a bound on the L^2 norm of derivatives of order p, the optimal convergence rate is of the form $n^{-2p/(2p+d)}$. Indeed, a consequence of Stone's result is that this asymptotic rate cannot be improved. This rate is achieved in regression contexts by multivariate smoothing splines (Cox 1984) and in some cases by least squares polynomial regression and trigonometric series regression, see Cox (1988). A. R. Barron (1988) has analogous results for the

estimation of a log-density function. For the special case $d=1$, asymptotic (and in some cases exact) minimax estimators are found in Efroimovich and Pinsker (1983) for density estimation, and Nussbaum (1985) and Speckman (1985) for regression. In these univariate cases the constant $c(p)$ is determined in the asymptotic minimax error $c(p)n^{-2p/(2p+1)}$. For $d>1$, it appears that the corresponding constant c(p,d) for exact asymptotics $c(p,d)n^{-2p/(2p+d)}$ is not yet explicitly determined. Determination of the behavior of this constant for large d would be useful, since it would help determine whether practical minimax estimation is possible in high dimensions.

Observe that unless the degree of smoothness p is large compared to the dimension d, the optimum rate of convergence $n^{-p/(2p+d)}$ is disappointingly slow. For instance, with dimension $d = 8$ and smoothness $p = 2$, a sample of size $n \geq 10^6$ (one million!) would be required to make $n^{-p/(2p+d)}$ be not greater than 1/10.

The slow rates for optimal estimation of smooth functions in high dimensions suggest that to understand the practical success of certain high-dimensional estimation strategies it may be necessary to use notions of the regularity of a function other than differentiability to quantify the limits on statistical accuracy. One possibility is to assume proximity of the desired function to functions of low Kolmogorov complexity. It may then be possible to obtain rate of convergence results as well as the consistency results referred to in section 5 (for networks selected by complexity regularization). This is a topic of further investigation.

In recent work by Baum and Haussler, the Vapnik-Chervonkis dimension of families of network functions is characterized and used to quantify the statistical reliability of estimated networks for binary classification. Using results of Cover (1965, 1967) on the number of possible dichotomies of a sample by networks of thresholded linear elements, Baum (1988) has bounded the Vapnik-Chervonkis dimension in terms of the total number of coefficients in the network. Let $0<\varepsilon_1<\varepsilon_2<1$ be given. Suppose it is observed that the fraction of errors of an estimated network is less than ε_1 on a training sample of size n. Then it is of interest to bound the conditional probability that a fraction of at least ε_2 errors will be incurred by this network on an independent test sample. Baum and Haussler (1988) have some results in this direction, assuming that the total number of coefficients is sufficiently small compared to the sample size.

The advantage of the Baum and Haussler approach is its uesfulness in retrospective analysis: i.e., *given* that an accurate estimate has been found on training data, what is the probability of error likely to be on new data? This approach avoids questions concerning the approximation capabilities of a network: in particular, the probability that an estimated network will achieve a certain accuracy is not determined.

15. Conclusions

Historically, neural networks, adaptive polynomial learning, and nonparametric statistical inference are fields of inquiry with distinct perspectives and separate lines of development which have crossed paths only on occasion. However, by examining the purpose, scope, and methodologies in these fields, considerable commonality is revealed. In each case, network functions are used to approximate possibly complex multivariate relationships by composition of many simpler relationships. Moreover, strategies for the synthesis of these networks from observable data are developed. To understand the performance of these strategies and to suggest improved methodologies, practical experience is supplemented by an understanding of the basic disciplines of mathematical approximation theory and statistical decision theory. Conversely, it behooves the practitioner in multivariate nonparametric statistical inference to become aware of the benefits and experiences in the use of multiple-layered networks for classification, regression, and related problems.

In our experience the most successful learning network methodologies adaptively grow the network structure, using all the observational data (in batch rather than recursively) and using an appropriate model selection criterion to ensure a parsimonious network. Moreover, the best strategies employ network structures which are not limited in their approximational capabilities. The principle examples of these successful methodologies are adaptively synthesized polynomial networks and projection pursuit.

It appears to us that several different approaches lead inevitably to one network structure and similar synthesis strategies: namely the network of fig. 7 (introduced by Kolmogorov and Lorentz) estimated by a generalization of projection pursuit which incorporates additive projections or estimated by polynomial network strategies specialized to this structure. This network considerably extends the capabilities of existing projection pursuit and additive regression models, yet retains enough of the regularity of these models that it may be amenable to further theoretical and practical examinations of its properties. Nevertheless, we should not restrict all attention to just one network structure. Hopefully, by consideration of a variety of different compositions, empirically selecting the best (say by complexity regularization), discovery of the true relationships can occur.

Appendix: Convergence of networks estimated by complexity regularization

In this appendix we specialize some results from A.R. Barron (1985,1987) to show convergence of estimates of network functions. In general the theory is concerned with the selection of a probability distribution using random data $W^n = (W_1, W_2, \ldots, W_n)$. It is assumed that Γ is a countable collection of probability distributions which are candidates for the estimate of the distribution of the process $W_1, W_2, \ldots$ and that $L(P)$, $P \in \Gamma$ are positive numbers which satisfy the Kraft-McMillan inequality $\sum_{P\in\Gamma} 2^{-L(P)} \leq 1$. (Here $L(P)$ may be regarded as the length of a uniquely decodable code or $2^{-L(P)}$ may be regarded as a discrete prior probability.) Short lengths $L(P)$ are desired for as large as possible a set of distributions that can be computed, so ideally, we would let $L(P)$ be the Kolmogorov complexity (relative to a fixed universal computer) and Γ would be the set of all computable distributions; however, the determination of such an ideal complexity is practically infeasible. Nevertheless, the complexity principle provides a useful guide in selecting reasonable sets of distributions and assigning priors geared toward parsimonious distributions. When the distribution is known except for a function f of d variables on which the distribution P_f depends, then families of network functions and corresponding description lengths can be used to yield an effective criterion for selecting an appropriate network.

In general the *complexity regularization* estimator $\hat{P}_n$ is defined to achieve

$$\min_{P\in\Gamma}\{-\log p^n(W_1,\ldots,W_n) + L(P)\} \qquad (9)$$

Here the density functions p^n are taken with respect to a fixed dominating measure. Logarithms are taken base 2. When $W_1,\ldots,W_n$ are discretized random variables, then $-\log p(W_1,\ldots,W_n)$ (upon rounding up to the nearest integer) is the length of a Shannon code for these variables based on the distribution P and the term $L(P)$ is the length of a preamble required to specify which distribution. A more general form of complexity regularization is to minimize

$$CR = -\log p^n(W^n) + \lambda L(P) \qquad (10)$$

where λ may be regarded as a Lagrange multiplier. Unless $\lambda = 1$, CR does not have the same total description length interpretation. Nevertheless, the solutions $\hat{P}_n$ which minimize CR for $\lambda > 0$ do have the valid interpretation as maximum likelihood estimators subject to complexity constraints. Such estimators were first proposed by Cover (1972). Our convergence results require that $\lambda \geq 1$ be fixed, although in one case $\lambda > 1$ is required.

We mention several general convergence results. First suppose that the distributions P in Γ are stationary and ergodic. Let P^* denote the true probability law which governs the process. The first result is that *if $P^* \in \Gamma$ then the estimated distribution is exactly correct, $\hat{P}_n \equiv P^*$, for all large n, with probability one.* For the remaining results suppose that the variables W_i are independent and identically distributed with respect to P^*, and likewise that independence holds for the distributions in Γ, whence $p(W_1,\ldots,W_n) = \prod p(W_i)$. Moreover, it is assumed that the true density function p^* can be approximated by densities in Γ in an information theoretic sense: that is, there exist densities in Γ for which the relative entropy $\int p^* \log p^*/p$ is arbitrarily small. This leads to the second result that $\hat{P}_n \Rightarrow P^*$ (in the sense of weak convergence) with probability one; moreover, if the densities in Γ are uniformly equicontinuous then $\hat{p}_n \rightarrow p^*$ in L^1. Since the uniform equicontinuity is not easy to guarantee in general, we mention a third result which makes no such requirement. *If $\lambda > 1$ and if densities in Γ can approximate p^* in the relative entropy sense, then $\hat{p}_n \rightarrow p^*$ in L^1, that is* $\lim \int |\hat{p}_n - p^*| = 0$, *with probability one.* The second and third results continue to be valid (with convergence in probability statements replacing convergence with probability one) when the set Γ_n and the numbers $L_n(P)$ are allowed to depend on the sample size n, provided that there exists a sequences of densities p_n in Γ_n for which $\lim \int p^* \log p^*/p_n = 0$ and $\lim L_n(P_n)/n = 0$.

For the estimation of network functions we take $W_i = (\underline{X}_i, Y_i)$ which is assumed to have a distribution P_f which depend on the function we desire to estimate. A denumerable (possibly finite) collection S_n of parameterized families of network functions $f(x,\theta)$ is considered. We assume that the sequence of collections is increasing $S_1 \subset S_2 \subset \cdots$ and that $L(f)$, $f \in S$ are lengths of codes which specify the structure, but not the parameter values, of networks in $S = \bigcup_n S_n$. For each network family f, the parameter vector (which has dimension denoted by k_f), is assumed for convenience to take values in the unit cube $[0,1]^{k_f}$. (Families with larger rectangular parameter spaces can be reduced to this case by scaling and appropriately modifying the definition of f). We restrict attention to the lattice $\Omega_{n,k}$ of points with coordinates of the form $i/\sqrt{n}$ for integers $0 \leq i < \sqrt{n}$ and we use $(1/2)\log n$ bits per parameter to describe these points.

For each parametrized network $f(x,\theta)$ in S_n, let $\hat{\theta}_n$ be estimated by the method of maximum likelihood restricted to the parameter values of the given precision. Thus $\hat{\theta}_n$ achieves

$$p(W^n \mid f(\cdot,\hat{\theta}_n)) = \max_{\theta \in \Omega_{n,k_f}} P(W^n \mid f(\cdot,\theta)). \qquad (11)$$

The complexity regularization estimator is the network $\hat{f}_n$ defined to achieve

$$\min_{f \in S_n} \{-\log p(W^n \mid f(\cdot,\hat{\theta}_n)) + \lambda \frac{k_f}{2} \log n + \lambda L(f)\}. \quad (12)$$

We remark that other precisions than $(1/2)\log n$ bits could be used in the definition, provided the maximum likelihood estimator is suitably restricted. (For smooth families, a second order Taylor series argument shows that the present choice achieves roughly the best tradeoff between complexity and likelihood. In some cases an improved tradeoff is obtained using local reparametrizations as dictated by the Fisher information matrix, as in A.R. Barron (1985, p. 74). With $\lambda = 1$, the specialization of the complexity regularization criterion given in (12) is very much the same as Rissanen's MDL criterion. However, the $L(f)$ term (omitted by Rissanen) can be important, especially when there is a large variety of families under consideration.

As a special case of interest consider function fitting problems with Gausssian errors. In this case, for given $\underline{X}$, the conditional distribution of the error $Y - f(\underline{X})$ is normal with mean zero and variance σ^2. The $\underline{X}_i$ are assumed to be randomly selected, independently, from a distribution which does not depend on f. Then the complexity regularization criterion reduces to

$$CR = \frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i - f(X_i,\hat{\theta}))^2 + \lambda \frac{k_f}{2} \log n + \lambda L(f). \qquad (13)$$

Let f^* be the true function which we desire to estimate. Assuming the the network in S are continuous functions of their parameters, the information theoretic closure condition reduces (in the Gaussian case) to the condition that $\inf_{f \in S} \inf_\theta E(f^*(X) - f(X,\theta))^2$, i.e. the true function must be approximable in the L^2 sense by members of network families under consideration. In which case, networks $\hat{f}_n(X)$ which are selected to minimize (13) (with $\lambda > 1$) are guaranteed to converge to $f^*(X)$ in probability.

References

H. Akaike (1970) Statistical predictor identification, *Ann. Inst. Stat. Math*, 22 203-217.

H. Akaike (1973) Information theory and an extension of the maximum likelihood principle, *Proc. 2nd Int. Symp. Inform. Theory*, B.N. Petrov and F. Csaki (Eds), 267-281 Akademia Kiado, Budapest.

J.A. Anderson and E. Rosenfeld (1988) *Neurocomputing: Foundations of Research* MIT Press.

S. Agmon (1954) The relaxation method for linear inequalities, *Canadien J. Math*, 6 382-392,

U.K. Ashby (1952) *Design for a Brain*, Wiley, New York.

A.C. Atkinson (1980) A note on the generalized information criterion for choice of a model, *Biometrica*, 67 413-418.

A.C. Atkinson (1981) Likelihood ratios, posterior odds and information criteria, *J. Econometrics*, 16 15-20.

A.R. Barron, F.W. van Straten, and R.L. Barron (1977) Adaptive learning network approach to weather forecasting: a summary, *Proc. IEEE Int. Conf. Cybernetics and Society*, 724-727.

A.R. Barron (1984) Predicted squared error: a criterion for automatic model selection, *Self-Organizing Methods in Modeling*, S.J. Farlow (Ed.), Marcel Dekker, New York.

A.R. Barron (1985) *Logically Smooth Density Estimation*, Ph.D. Thesis, Stanford University.

A.R. Barron (1987) The exponential convergence of posterior probabilities with implications for the consistency of Bayes density estimators, submitted to *Ann. Statist.*

A.R. Barron (1988) Approximation of densities by sequences of exponential families, submitted to *Ann. Statist.*

R.L. Barron, R.F. Snyder, E.A. Torbett, and R.J. Brown, (1964) *Advanced Computer Concepts for Intercept Prediction*, Adaptronics, Inc. Final Technical Report, Army Nike-X Project Ofc., Redstone Arsenal, AL, November 1964. (See Vol.I: *Conditioning of Parallel Networks for High-Speed Prediction of Re-entry Trajectories.*)

R.L. Barron (1974) Theory and application of cybernetic systems: an overview, *Proc. 1974 IEEE Nat. Aerospace and Elect. Conf.*, 107-118.

R.L. Barron (1975) Learning networks improve computer-aided prediction and control, *Computer Design*, August 1975, 65-70.

R.L. Barron, A.N. Mucciardi, F.J. Cook, J.N. Craig, and A.R. Barron (1984) Adaptive learning networks: development and application in the United States of algorithms related to GMDH, *Self-Organizing Methods in Modeling*, S.J. Farlow (Ed.), Marcel Dekker, New York.

R.L. Barron and D. Abbott (1988) Use of polynomial networks in optimum, real-time, two-point boundary-value guidance of tactical weapons, *Proc. Military Comp. Conf.*, May 3-5, Anaheim, CA.

E.B. Baum (1988) On the capabilities of multilayer perceptrons, *J. Complexity* (in press).

E.B. Baum and D. Haussler (1988) What size net gives valid generalization?, *IEEE Int. Symp. Inform. Theory*, June 19-24, Kobe, Japan.

L. Breiman and J.H. Friedman (1985) Estimating optimal transformations for multiple regression and correlation (with discussion), *J. Am. Stat. Assoc.*, 80 580-619.

T.M. Cover (1965) Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition, *IEEE Trans. Elect. Comp.*, 326-334.

B. Clarke and A.R. Barron (1988) Information theoretic asymptotics of Bayes methods, submitted to *IEEE Trans. Inform.*.

T.M. Cover (1967) Capacity problems for linear machines, *Statistical Classification Procedures*, 283-289.

D.D. Cox (1984) Multivariate smoothing spline functions, *SIAM J. Numer. Anal.*, 21 789-813.

D.D. Cox (1988) Approximation of linear regression on nested subspaces, *Ann. Stat.*,18.

G. Cybenko (1988) *Continuous Valued Neural Networks with Two Hidden Layers are Sufficient*, Tech. Report Dept. Computer Science, Tufts Univ. Medford, Mass.

L. Devroye (1987) *A Course in Density Estimation*, Birkhauser, Boston, Mass.

D. Donoho and I.M. Johnstone (1985) Discussion on projection pursuit, *Ann. Stat.*, 13 496-500.

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

R.L. Eubanks (1988) *Spline Smoothing and Nonparametric Regression*, Marcel Dekker, New York.

S.Y. Efroimovich and M.S. Pinsker (1982) Estimation of square-integrable probability density of a random variable, *Problems in Information Transmission*, 18 175-189.

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

B. Farley and W. Clark (1954) Simulation of self-organizing systems by digital computer, *IRE Trans. Inform. Theory*, 4 76-84.

R.A. Fisher (1922) The goodness of fit of regression formulae and the distribution of regression coefficients, *J. Roy. Stat. Soc.*, 85 597f.

R.A. Fisher (1934) Probability likelihood and quantity of information in the logic of uncertain inference, *Proc. Roy. Soc. A*, 146 1f.

R.A. Fisher (1936) The use of multiple measurements in taxonomic problems, *Ann. Eugenics*, 7 179-188.

J.H. Friedman and J.W. Tukey (1974) A projection pursuit algorithm for exploratory data analysis, *IEEE Trans. Computers*, 23 881-889.

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

J.H. Friedman, W. Stuetzle and A. Schroeder (1984) Projection pursuit density estimation, *J. Amer. Stat. Assoc.*, 79 599-608.

L.O. Gilstrap Jr. (1971) Keys to developing machines with high-level artificial intelligence, *Proc. ASME Design Eng. Conf.*, ASME Paper 71-DE-21.

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

J. Hopfield (1982), Neural networks and physical systems with emergent collective computational abilities, *Proc. Nat. Ac. of Sciences*, 79 2554-2558.

P.J. Huber (1985) Projection pursuit (with discussion), *Ann. Stat.*, 13 435-525.

A.G. Ivakhnenko (1971) Polynomial theory of complex systems, *IEEE Trans. Systems, Man, Cybernetics*, 1 364-378.

D. Jackson (1930) *The Theory of Approximation*, Am. Math. Soc., New York.

L. Jones (1986) Convergence of generalized projection pursuit in the nonsampling case, unpublished manuscript.

L. Jones (1987) On a conjecture of Huber concerning the convergence of projection pursuit regression, *Ann. Stat.*, 15 880-882.

A.N. Kolmogorov (1957) On the representation of continuous functions of several variables by superpositions of continuous functions of one variable and addition, *Dokl*, 114 679-681.

A.N. Kolmogorov and V.M. Tikhomirov (1959) Entropy and ε-capacity of sets in function spaces, *Uspehi*, 14 3-86.

R.J. Lee and L.O. Gilstrap (1960) Learning machines, *Proc. Bionics Symp.*, USAF Wright Air Development Division, Dayton, Ohio, TR60-600 437-450.

G.G. Lorentz (1976) The 13th problem of Hilbert, in *Mathematical Developments Arising from Hilbert Problems*, F.E. Browder (Ed.), Am. Math. Soc., Providence, R.I.

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

M.L. Minsky (1954) *Neural-Analog Networks and the Brain Model Problem*, Ph.D. Thesis, Princeton Univ.

M.L. Minsky and S. Papert (1969) *Perceptrons: An Introduction to Computational Geometry*, M.I.T. Press, Cambridge, Mass.

W.S. McCulloch and W. Pitts (1943) A logical calculus of the ideas immanent in nervous activity *Bull. Math. Biophysics*, 5 115-133.

R.E.J. Moddes , R.J. Brown, L.O. Gilstrap Jr., et al. (1965) Study of Neurotron Networks in Learning Automata, Adaptronics Inc., Air Force Avionics Laboratory, Dayton, Ohio, AFAL-TR-65-9.

A.N. Mucciardi (1972) Neuromime nets as the basis for the predictive component of robot brains, *Cybernetics, Artificial Intelligence, and Ecology*, H.W. Robinson and D.E. Knight (Eds.), Spartan Books, 159-193, 4th Ann. Symp. Am. Soc. Cybernetics, October 1970.

N.J. Nilsson (1965) *Learning Machines: Foundations of Trainable Pattern-Classifying Systems*, McGraw-Hill, New York.

J. von Neumann (1956) Probabilistic logics and the synthesis of reliable organisms from unreliable components, *Automata studies*, C.E. Shannon and J. McCarthy (eds), Princeton Univ. Press, 43-98.

M. Nussbaum (1985) Spline smoothing in regression models and asymptotic efficiency in L2, *Ann. Stat.*, 13 984-997.

Prakasa Rao (1983) *Nonparametric Functional Estimation*, Academic Press, Orlando.

J. Rissanen (1978) Modeling by shortest data description, *Automatica*, 14.465-471.

J. Rissanen (1983) A universal prior for integers and estimation by minimum description length, *Ann. Stat.*, 11 416-431.

J. Rissanen (1984) Universal Coding, Information, Prediction, and Estimation, *IEEE Trans. Inform. Theory*, 30 629-636.

H.E. Robbins and Monroe (1951) A stochastic approximation method, *Ann. Math. Stat.*, 22.400-407.

F. Rosenblatt (1958) *The Perceptron: A Theory of Statistical Separability in Cognitive Systems*, Cornell Aeronautical Laboratory Report No. VG-1196-G-1.

F. Rosenblatt (1962) *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Spartan Books, Washington, D.C.

D.E. Rumelhart, G.E. Hinton, and R.J. Williams (1986) Learning representations by back propagations, *Nature* , 323 533-536.

D.E. Rumelhart and J.L. McClelland (1986) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Vol.1: *Foundations*, M.I.T. Press, Cambridge, Mass.

G. Schwarz (1978) Estimating the dimension of a model, *Ann. Stat.*, 6 461-464.

R. Shibata (1984) Approximate efficiency of a selection procedure for the number of regression variables, *Biometrika*, 71 43-49.

R. Shibata (1986) Selection of the number of regression variables: a minimax choice of generalized FPE, *Ann. Inst. Stat. Math.*, 38 459-474.

S. Shrier, R.L. Barron, and L.O. Gilstrap (1987) *Proc. IEEE 1st Int. Conf. Neural Networks* II, 431-439.

P. Speckman (1985) Spline smoothing and optimal rates of convergence in nonparametric regression models, *Ann. Stat.*, 13 970-983.

C.J. Stone (1982) Optimal global rates of convergence for nonparametric regression, *Ann. Stat.*, 10 1040-1053.

C.J. Stone (1985) Additive regression and other nonparametric models, *Ann. Stat.*, 13 689-705.

R. Tibshirani (1988) Estimating transformations for regression via additivity and variance stabilization, *J. Am. Stat. Assoc.*, 83 394-405.

B. Widrow and M.E. Hoff (1960) Adaptive switching circuits, *1960 IRE WESCON Convention Record*, 96-104.

B. Widrow (1962) Generalization and information storage in networks of Adaline neurons, *Self-Organizing Systems*, M.C. Yovits, G.T. Jacobi, and G.D. Goldstein (ed's), Spartan Books, Washington, D.C., 435-461.

B. Widrow, R.G. Winter, and R.A. Baxter (1987) Learning phenomenona in layered neural networks, *Proc. IEEE 1st Int. Conf. Neural Networks* II, 441-430.

S. Winsberg and J.O. Ramsay (1980) Monotonic transformations to additivity using splines, *Biometrica*, 67 669-674.