1

Learning Functional Relationships In High Dimensions: An Assessment of Perceptrons, Neural Networks, and Other Methodologies

Marco Campi
Department of Electrical Engineering for Automation
University of Brescia
via Branze 38
25123 Brescia
Italy

and

P.R. Kumar
Department of Electrical and Computer Engineering, and
Coordinated Science Laboratory
University of Illinois
1308 West Main Street
Urbana, IL 61801
USA

We are currently in the "data" age. As we enter the "information" age the problem of learning functional relationships from data will loom large.

We provide an assessment of results ranging from perceptrons and neural networks to function estimation in general.

Progress has been made on the function theoretic basis of approximation methods. Progress has also been made on the problem of learning from data.

Two main difficulties center around (i) assessment of the function class in which the true function is assumed to exist, and (ii) the existence of a computationally tractable algorithm for assimilating the training data.

We show that there do exist finite algorithms for training; however the required computational complexity can be very high. We also show that the formulation of the training problem as a convex optimization problem presents some difficulties, at least if approached in a straightforward fashion.

Finally almost no work has been done, to our knowledge, on the capability of the function estimation methods to solve high level problems of interest, as was done for example in the case of perceptrons.

2

Introduction

Since the late 1980s neural networks have attracted a lot of attention. This is the second such wave, since they had been the subject of a spurt of attention in the 1950s and 1960s too (see Rosenblatt (1961)). Without getting into the pros and cons of the neural network methodologies proposed, one could first ask the question: Why did they attract so much attention?

The answer, it appears, is that in many fields there is an accumulated storehouse of data, and a pressing need to interpolate or extrapolate between the data points. While it is not yet clear whether we have truly entered a new "information age" it is probably more accurate to say that we are in a "data age". Huge quantities of data are becoming available and one wishes to use the data to make predictions.

For example, in seismic exploration, there may exist a record of historical data on past oil wells sunk, the seismic data obtained prior to drilling the holes, as well as the outcome, i.e., success or failure or volume of oil obtained. Given this historical data base it is necessary to make a decision on whether to drill a hole in a particular proposed location with given seismic data. Similarly, in banking, there may exist a record of credit histories and when a new application for credit may be received, one wishes to use the past data to make a decision on whether to provide credit. Yet another application is in speech recognition. Given past speaker records, one wishes to classify a new waveform.

All these are problems of *function estimation*. We are given pairs (x_1, y_1), $(x_2, y_2), \ldots, (x_N, y_N)$ of data. Given a new x, one wishes to predict the y value associated with it. Typically, the "independent" variable x is high-dimensional.

Clearly, there is a great demand on the user side for a good methodology for function estimation.

One reason for the attention that neural networks have received since the 1980s is that they are addressed at this pressing problem. (We will address in the sequel whether it is a *satisfactory* solution.) Another reason is that neural networks yield a *general purpose* method. By this is meant that they can be applied to general problems without any particular structure. A last reason, perhaps, is the subliminal appeal associated with the word "neural."

For a function estimation method to be satisfactory, it must be statistically sound, and also computationally tractable. By statistical soundness, we mean that the process for making inferences from noisy data is sound. Subsumed here is that the approximation procedure should be function theoretically sound, i.e., that the parameterization of functions is done appropriately. By computational tractability we mean that the amount of computation that needs to be done to obtain the answer is tractable given the high dimensionality of each data point, the number of data points available, and of course the accuracy and confidence sought in the answer.

As noted at the outset, the present phase of interest in neural networks which started in the 1980s and which continues today is but the second phase of interest in the field. The first phase was in the 1950s. In that phase much research was done on the capabilities of *perceptrons*. Since the 1980s the focus has been on multi-layer acyclic feedforward neural networks.

In the first half of this paper we will outline the results obtained on perceptrons. In the second half of the paper we will examine more recent results on function estimation, including, but not limited to, multi-layer

feedforward (or acyclic) networks. We will also provide some results of our own showing that there do exist finite algorithms for training which are guaranteed to converge; however their computational complexity can be very high. Also, we show that the reduction of the function fitting problem to a convex optimization problem cannot be done in a straightforward fashion.

We will examine the following issues:

(i) Representation capability of parameterization used.

(ii) Complexity of representation.

(iii) Number of data samples needed for a given accuracy and confidence in the estimate.

(iv) Convergence of the training procedure.

(v) Complexity of the training or function fitting procedure.

(vi) The power of the method used for solving certain high level problems.

The central thesis of this article is that only by such an examination can we properly assess where we are today and where further progress needs to be made.

3

A Little History: The Mark-1 Perceptron

The Mark-1 Perceptron (see Block (1962)) consisted of three components:

(i) a "retina" R,

(ii) a set of "associator units" $\{\phi_i\}$, and

(iii) a "decision unit,"

as shown in Figure 3.1. The retina R consisted of a 20×20 array of pixels. An *image* x, regarded simply as

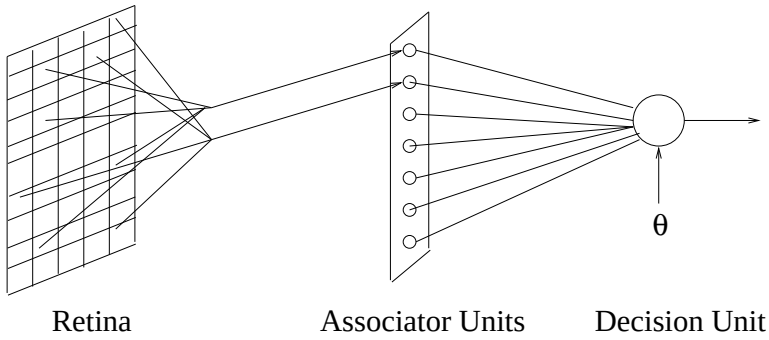

Figure 3.1: The Mark-1 Perceptron.

the subset of pixels which are "on," is impressed on the retina. There were 512 associator units. Each pixel was randomly connected to 40 associator units with fixed, i.e., non-adjustable, weights. Each associator unit ϕ_i has its output multiplied by an *adjustable weight* w_i. The weighted sum $\sum_i w_i \phi_i(x)$ is compared by the decision unit to an *adjustable threshold* θ, and a binary output decision $y(x)$ is thereby made. If $\phi(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_p(x))^T$ and $w = (w_1, w_2, \ldots, w_p(x))^T$, then the decision rule is:

$$\begin{aligned} y(x) &= 1 \text{ if } w^T \phi(x) > \theta \\ &= 0 \text{ if } w^T \phi(x) \leq \theta. \end{aligned}$$

Such a function is called a *linear threshold function.*

4

The Geometric View

Fix any set of weights w. It defines a hyperplane $w^T\phi = \theta$, and all ϕ's on its "right" are classified as 1, and all ϕ's on its "left" are classified as 0. Thus, at the level of the ϕ's, the perceptron can separate disjoint convex sets.

For example, if "$\oplus$" denotes Exclusive OR, i.e., $a_1 \oplus a_2 = 1$ if $(a_1 = 1, a_2 = 0)$ or $(a_1 = 0, a_2 = 1)$, and $a_1 \oplus a_2 = 0$ otherwise, then clearly the perceptron cannot separate the set of ϕ's which have value 1 (i.e., $\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$) from the set of ϕ's which have value 0 (i.e., $\left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}$.

5

The Training Algorithm for Perceptrons: Block's Convergence Theorem

Let us suppose that we are given two finite sets of ϕ's, Φ^+ and Φ^-, which are separable by a hyperplane. The sets Φ^+ and Φ^- are *positive* and *negative exemplars*, respectively. The problem of *training* is to find a w and a θ such that

$$\begin{aligned} w^T\phi &> \theta \text{ for all } \phi \in \Phi^+, \\ &\leq \theta \text{ for all } \phi \in \Phi^-. \end{aligned}$$

This is accomplished by a training algorithm whose convergence was proved by Block.

Theorem. *Let $\phi^{(0)}, \phi^{(1)}, \ldots$ be any infinite sequence with repetitions, drawn from $\Phi^+ \cup \Phi^-$. This is the "training" sequence. Let $w^{(0)}$ be any initial set of weights, and $\theta^{(0)}$ be any initial threshold. Consider the following training algorithm. $\phi^{(t)}$ is presented to the perceptron. If the classification by the perceptron is correct, then the weight vector is left unchanged, i.e., $w^{(t+1)} = w^{(t)}$. Otherwise, let $w^{(t+1)} = w^{(t)} + \phi^{(t)}$ if $\phi^{(t)} \in \Phi^+$, or $w^{(t+1)} = w^{(t)} - \phi^{(t)}$ if $\phi^{(t)} \in \Phi^-$. Then, after some finite T, i.e., for all $i \geq T$, the perceptron correctly classifies all future $\phi^{(i)}$'s.*

Equipped with a simple training algorithm which was guaranteed to converge to a correct classifier for the *given data*, if one such existed, the stage was set for applications. Many experiments were performed including classifying letters of the alphabet (for which five decision units are needed since $2^4 < 26 \leq 2^5$), separating circles from squares, separating vertical from horizontal bars, etc.

6

Generalizing from Data

Suppose that we have successfully trained the perceptron so that it correctly classifies the given data points. Does that mean it will perform well on an instance *not* observed in the data (i.e., training) set? That is, does the perceptron *generalize* well?

Here, some powerful results of Vapnik and Chervonenkis (1971) come to the rescue. Suppose there are p associator units. Then ϕ, $w \in R^p$. Consider the set of all half-spaces in R^p. Its Vapnik-Chervonenkis dimension is $p + 1$. Appealing to uniform convergence in the law of large numbers, one can assert the following:

Theorem. *Suppose that there is a vector of weights which can correctly classify all instances of the N data samples. Consider any probability distribution on R^p. Suppose that the N data points have been selected in an iid manner, and the perceptron has been correctly trained on the data set. Suppose that*

$$N \geq Max\left\{\frac{4}{\epsilon}\log\frac{2}{\delta}, \frac{8(p+1)}{\epsilon}\log\frac{8(p+1)}{\epsilon}\right\}.$$

Then if an independent point in R^p is chosen, the probability that the N samples are enough to train the perceptron such that the misclassification error probability is less than ϵ, is greater than $(1-\delta)$.

The above theorem says that the number of samples needed is roughly linear in the dimension p of the problem. Hence the data needed to learn how to separate two linearly separable sets is not too great.

7

A Complexity Theory for Perceptrons

Minsky and Papert (1988) went deeper into the capabilities of perceptrons by examining what high level problems they could solve when their *complexity* was bounded in a certain way.

Let us allow us an associator unit any binary valued function ϕ_i on images impressed on the retina, with the property that it depends only on B pixels, irrespective of the size of the retina. The idea here is to restrict each associator unit to be "simple" in the sense that it looks only at a few pixels. However we will allow an arbitrarily large number of associator units. The issue examined by Minsky and Papert was what could be accomplished by pooling, through a linear threshold function, a possibly large number of predicates each of simple form.

While the particular definition of complexity adopted above is debatable, the thrust of the issue was on examining capability vs. complexity, which we believe to be exactly the right issue to focus on.

Let us say that a predicate on images is of *order* B if it can be realized by a perceptron with each associator unit dependent on no more than B pixels, irrespective of the size of the retina, i.e., as $|R| \to +\infty$. An example is clarifying.

Example: Convexity is an Order 3 Predicate

Consider an image X on a retina. Let $\psi_{\text{CONVEX}}(X)$ be the convexity predicate, i.e., $\psi_{\text{CONVEX}}(X) = 1$ if X is convex, and $= 0$ otherwise. (Note that since the retina is discretized we need to define what we mean by a "convex" set. However we will ignore this issue here; see Minsky and Papert (1988) for an elaboration of this issue). If the image is not convex, then there exists some set of three collinear points such that the outer two points are in the image but the middle one is not. Hence $\psi_{\text{CONVEX}}(X)$ is an order 3 predicate.

8

Parity and Connectedness

One example of a high level problem is that of recognizing connectedness in images. In defining connectedness in a discretized image we will not recognize corner contacts between pixels.

One of the central results of Minsky and Papert (1988) is that $\psi_{\text{CONNECTED}}$, the predicate denoting connectedness, is *not* of bounded order. That is, as $|R| \to +\infty$, it is indeed necessary to use associator units which depend on an unbounded number of pixels.

In the proof of this result an important role is played by the *parity predicate* ψ_{PARITY} which is 1 if and only if the number of pixels in the image is odd.

Consider the two retinas shown in Figure 8.1. There are 3 pixels in the retina on the left, denoted x_1, x_2 and

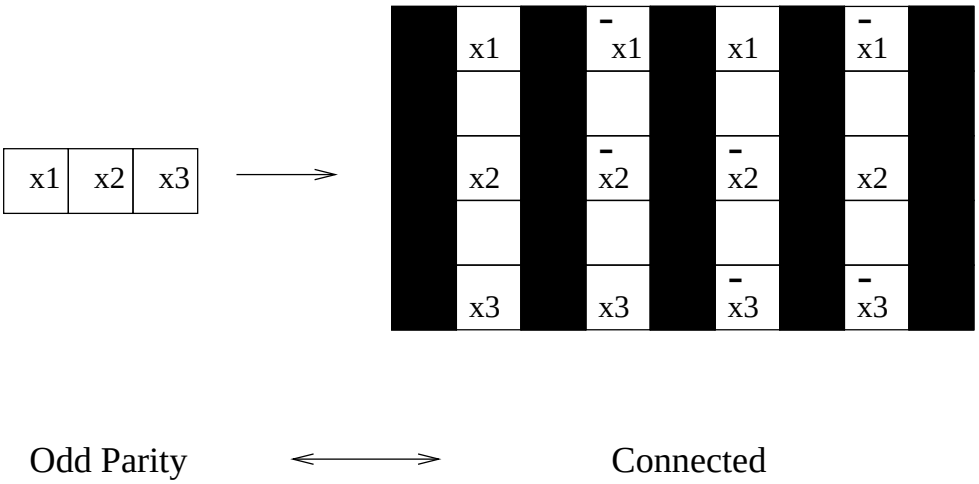

Figure 8.1: Transforming parity into connectedness.

x_3. (A value of $x_i = 1$ denotes that the pixel is black, and a value of 0 denotes that it is white). The retina on the right is a 5×9 array of pixels. Given any image on the retina on the left, an image on the retina on the right is produced as follows: Every odd column is always colored black. All the other pixels are either x_i or $\bar{x}_i$ (the complement of x_i) and colored according to what their values are in the retina on the left.

It is easy to check that the image produced on the retina in the right is connected if and only if the image on the retina on the left is of odd parity.

This mapping between parity and connectedness can be extended to larger sized retinas. From that we can deduce that:

$$\psi_{\text{CONNECTED}} \text{is of bounded order only if } \psi_{\text{PARITY}} \text{is of bounded order.}$$

Motivated by this, let us examine the order of ψ_{PARITY}. Unfortunately:

Theorem. *ψ_{PARITY} is of unbounded order.*

Outline of Proof. A predicate of the form $\phi_A(X) = 1$ if $A \subseteq X$, and $= 0$ otherwise, will be called the *mask* of A. By using the disjunctive normal form it follows that boundedness of the order of a predicate can be equally well determined even if we restrict the associator units to be masks (depending on no more than B pixels).

Let M_p be the set of all masks of order p, and let G be the group of all permutations on R. Note that permutations leave parity unchanged.

Now suppose that ψ_{PARITY} can be realized by using masks of order no more than $B < +\infty$. Then

$$\begin{aligned}
\psi_{\text{PARITY}}(X) &= \psi_{\text{PARITY}}(g^{-1}X) \text{(for every permutation } g \in G) \\
&= I\left[\sum_{\ell=0}^{B}\sum_{\phi\in M_\ell} w(\phi)(\phi g^{-1}(x)) > \theta\right] \\
&= I\left[\sum_{\ell=0}^{B}\sum_{\phi\in M_\ell} w(\phi g)\phi g g^{-1}(x) > \theta\right] \\
&\qquad \text{(since } M_\ell = M_\ell g\text{)} \\
&= I\left[\sum_{\ell=0}^{B}\sum_{\phi\in M_\ell} w(\phi g)\phi(x) > \theta\right] \text{ for every } g \in G \\
&= I\left[\sum_{\ell=0}^{B}\sum_{\phi\in M_\ell}\sum_{g\in G} w(\phi g)\phi(x) > \theta\right] \text{ (summing over } g \in G\text{)} \\
&= I\left[\sum_{\ell=0}^{B}\sum_{\phi\in M_\ell} \rho(\phi)\phi(x) > \theta\right] \text{ (defining } \rho(\phi) := \sum_{g\in G} w(\phi g)\text{)} \\
&= I\left[\sum_{\ell=0}^{B}\gamma(\ell)\sum_{\phi\in M_\ell} \phi(x) > \theta\right] \text{ for some } \gamma(\ell) \\
&\qquad \text{(noting } \rho(\phi) = \rho(\phi') \text{ for } \phi, \phi' \in M_\ell\text{)} \\
&= I\left[\sum_{\ell=0}^{B}\gamma(\ell)\binom{|X|}{\ell} > \theta\right] \\
&= I\left[\text{polynomial of degree } B \text{ in } |X| > \theta\right] \\
&=: I\left[q_B(|X|) > \theta\right].
\end{aligned}$$

However, for parity, we will need

$$\begin{aligned}
q_B(0) &\le \theta \\
q_B(1) &> \theta \\
&\vdots \\
q_B(R) &\le \theta \text{ if } R \text{ is even, or } > \theta \text{ otherwise.}
\end{aligned}$$

This requires the degree B of the polynomial to equal R. This is unbounded as $|R| \to +\infty$, showing that ψ_{PARITY} is of unbounded order.

From this we deduce the striking result:

Theorem. *$\psi_{CONNECTED}$ is of unbounded order.*

Minsky and Papert (1988) went further to examine which topologically invariant predicates are of bounded order. By topological invariance, we mean that the value of the predicate is unchanged by smooth deformations of the image. The deep result is that:

Theorem. *The only topologically invariant predicates which are of bounded order are those which are purely a function of the Euler number.*

Recall that

Euler Number $(X) :=$ # of connected components in X - # of holes in X.

This means that we cannot distinguish between the two images in Figure 8.2 when smooth deformations are

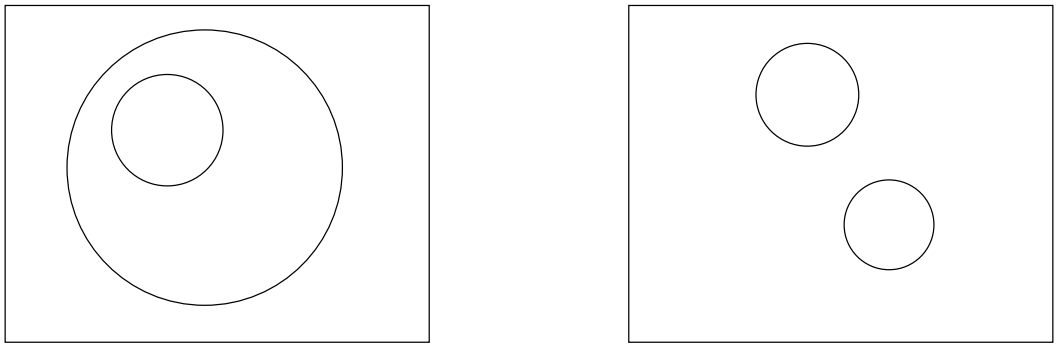

Figure 8.2: The images on the left and the right have the same Euler number.

allowed. A doughnut within a doughnut (on the left) has Euler number 0. So does the image on the right consisting of two separate doughnuts.

9

A Brief Summary of Perceptrons

This study of perceptrons illuminates their high level capabilities. To summarize, perceptrons have the following properties:

1. They can separate two disjoint convex sets.

2. They have a training algorithm which is guaranteed to produce a set of weights which separates any two finite data sets, provided they can be separated by a hyperplane.

3. They require a number of data points approximately linear in the dimension of the data. Then one can guarantee that with high precision and confidence the classification of an unseen data point will be accurate.

4. Certain high level problems (e.g., connectedness) cannot be solved by perceptrons unless we allow predicates which look at unbounded numbers of pixels.

What then can we say about neural networks with many layers? Or other function approximation methods? How do the results compare with the results known about the capabilities of perceptrons? We address these issues in the second half of the paper.

10

The General Function Estimation Problem

Let us now turn to the more general problem of function estimation from data samples. Given two compact sets $X \subseteq R^d$ and $Y \subseteq R$, consider a function F mapping X into Y. Function F is not known, and its value can only be measured with noise at N sample points,

$$y_k = F(x_k) + e_k, \quad k = 1, \ldots, N.$$

The points $\{x_k\}$ are assumed to be independently chosen according to some unknown probability measure μ. The problem we will examine is that of finding an approximation $\widehat{F}_n$ of function F, of the form

$$\widehat{F}_n = \sum_{i=1}^{n} c_i h(\theta_i), \tag{10.1}$$

where c_i's are real numbers and $h(\theta_i)$'s are continuous functions chosen from a set of basis functions

$$\{h(\theta) : X \to Y, \theta \in \Theta, \Theta \subseteq R^p \text{ compact set}\}.$$

The discrepancy between F and $\widehat{F}_n$ is measured by the L_2-norm

$$\|F - \widehat{F}_n\|^2 := \int (F(x) - \widehat{F}_n(x))^2 \mu(dx), \tag{10.2}$$

which can interpreted as the expected squared error we make if we use $\widehat{F}_n$ to predict the value of F for unseen x-samples.

11

Choice of Basis Functions

The first issue to confront is the choice of basis functions.

Example 1: Ridge Functions

A *ridge* function is the composition of a 1-dimensional function with a linear function. An example is

$$h(x;(v,\alpha)) := \text{ sat } (v^T x + \alpha),$$

where $v \in R^d$, $\alpha \in R$, and the saturation function sat $(\cdot)$ is defined as

$$sat(\zeta) = \begin{cases} 0, & \zeta \leq 0 \\ \zeta, & 0 \leq \zeta \leq 1 \\ 1, & \zeta \geq 1. \end{cases}$$

The sat$(\cdot)$ function is a special case of the *sigmoidal* function used in neural networks. Let $\sigma(\cdot)$ denote the sigmoidal function. It is a continuous approximation of the sign function, i.e., a continuous monotone increasing function satisfying:

$$\lim_{\zeta \to -\infty} \sigma(\zeta) = 0$$
$$\lim_{\zeta \to +\infty} \sigma(\zeta) = +1.$$

Using sigmoidal functions, the output of a neural network with one hidden layer can be written as

$$\hat{F}_n(x) = \sum_{i=1}^{n} c_i \sigma(v_i^T x + \theta),$$

where v_i is the vector of weights for the i-th node, and θ_i is its threshold.

It has been shown by Cybenko (1989) and Hornik *et al.* (1989) that the above sums can be used to approximate arbitrarily closely any continuous function on a compact set in the L_∞-norm. (Note that this also implies good approximation in the L_2-norm). Thus neural networks with one hidden layer can approximate target functions as closely as desired, by choosing a sufficiently large number of nodes in the hidden layer.

Example 2: Hinge Functions

A *hinge* function (see Breiman (1993)) consists of two hyperplanes continuously joined together. It is defined as either

$$h(x;(v_-,\alpha_-,v_+,\alpha_+)) := \min\{v_-^T x + \alpha_-, v_+^T x + \alpha_+\}$$

or

$$h(x; (v_-, \alpha_-, v_+, \alpha_+)) := \max\{v_-^T x + \alpha_-, v_+^T x + \alpha_+\}.$$

It can be shown that even this choice of basis functions allows one to approximate target functions as closely as desired; see Breiman (1993).

12

Complexity of Representation

Above we have given two examples of basis functions which can both be used to approximate functions. The next issue to address is: How many basis functions do we need to approximate to within a certain accuracy? Note that in the case of neural networks with one hidden layer, the number of basis functions is the same as the number of nodes, and hence the question can equivalently be posed as: What is the *complexity* of the network in terms of the number of nodes needed to approximate a target function to within a certain accuracy?

We will use the L_2-norm to measure the accuracy of the representation. The following result of Jones (1992) is fundamental.

Theorem. *Suppose*

$$\|h(\theta)\| \leq b \text{ for all } \theta \in \Theta, \text{ and} \tag{12.1}$$

$$F \text{ lies in the closure of the convex hull of } \{h(\theta), \theta \in \Theta\}. \tag{12.2}$$

Then there exists a sequence of functions $\widehat{F}_n$ of the form (10.1) such that

$$\|F - \widehat{F}_n\|^2 \leq \frac{c(b)}{n}, \tag{12.3}$$

where the constant $c(b)$ depends only on the bound b on the norm of the basis functions, and does not *depend on the dimension d of the space.*

Idea of the Proof. The proof actually provides a constructive approximation procedure. Let $\hat{F}_{n-1}$ be an approximation using $n-1$ terms. Since F belongs to the closure of the convex hull of the basis functions, it is not possible for *all* of the basis functions to lie on the same side of F as $\hat{F}_{n-1}$; see Figure 12.1. Hence there is at least one basis function which is on the other side; call it $g \in \{h(\theta) : \theta \in \Theta\}$. Then join $\hat{F}_{n-1}$ and g by a straight line, and consider the nearest point on it to F. Let $\hat{F}_n$ denote this new approximation. As illustrated in the Figure 12.1, the error $\|F - \hat{F}_n\|$ is small compared to the error $\|F - \hat{F}_{n-1}\|$. This can be quantified using Pythagoras' Theorem, and the bound b on the norms of $h(\theta)$ can be used to show the result.

We summarize Jones' algorithm for producing a sequence of approximations converging at the rate $O(\frac{1}{\sqrt{n}})$:

Jones' Algorithm

Initialize the sequence at any basis function:

$$\widehat{F}_0 = h(\theta_0).$$

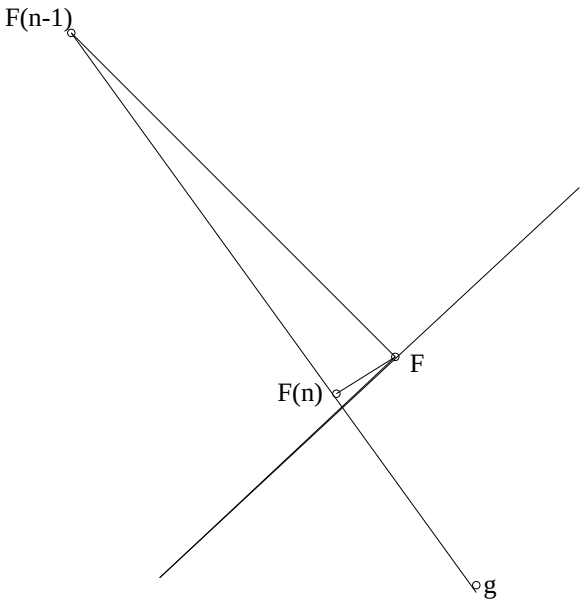

Figure 12.1: The approximation procedure of Jones.

Once $\widehat{F}_{n-1}$ has been determined, determine $\widehat{F}_n$ as follows:

$$(\alpha_n, \theta_n) = \mathrm{argmin}_{\substack{0 \le \alpha \le 1 \\ \theta \in \Theta}} \|\mathrm{F} - (1-\alpha)\widehat{\mathrm{F}}_{\mathrm{n-1}} - \alpha \mathrm{h}(\theta)\|^2;$$

$$\widehat{F}_n = (1-\alpha_n)\widehat{F}_{n-1} + \alpha_n h(\theta_n).$$

Some comments about the above approximation procedure are in order.

(i) First, the procedure does not just *add* a scaled basis function at each stage. Rather it *also* scales the earlier approximation. In fact, at each stage it chooses a convex combination of a new basis function and the old approximation.

(ii) Jones' algorithm can be viewed as a "greedy" approach. At each stage it opportunistically chooses a new basis function. In contrast if the sequence of basis functions is *pre-fixed* then the convergence rate can be as poor as $O(\frac{1}{n^{\frac{1}{d}}})$; see Barron (1993).

(iii) It is vital that the error criterion is the L_2-norm. It is possible to show that if the L_2-norm is replaced by the L_1-norm, then no greedy algorithm exists.

Example: An Application of Jones' Theorem to Fourier Series Approximation

Suppose that a function $F(x)$ and its Fourier Transform $\breve{F}(\omega)$ are both in L_1. Above $x \in R^d$, and $\omega \in R^d$. Then

$$F(x) = \frac{1}{(2\pi)^d} \int \breve{F}(\omega) e^{i\omega^T x} d\omega.$$

By multiplying and dividing by $\frac{|\breve{F}(\omega)|}{\|\breve{F}\|_1}$ this can be rewritten as

$$F(x) = \int \frac{|\breve{F}(\omega)|}{\|\breve{F}\|_1} Re\left[\frac{\breve{F}(\omega)\|\breve{F}\|_1}{(2\pi)^d|\breve{F}(\omega)|}e^{i\omega^T x}\right] d\omega.$$

Thus $F(x)$ belongs to the closure of the convex hull of $Re\left[\frac{\breve{F}(\omega)\|\breve{F}\|_1}{(2\pi)^d|\breve{F}(\omega)|}e^{i\omega^T x}\right]$.

Now we can apply Jones' result to conclude that a greedy trigonometric approximation with n terms has error no more than $\frac{c}{\sqrt{n}}$.

It should be noted that there are some severe difficulties in obtaining a greedy approximation. At each step we will have to scan over all harmonics and choose a good one. This may pose an impossible computational burden. In contrast traditional Fourier approximation proceeds by moving through the harmonics in order. It however cannot have an error rate as good as a greedy approach. In fact, as noted earlier, its convergence rate can be $O(\frac{1}{n^{\frac{1}{d}}})$; see Barron (1993).

For sigmoidal functions, Barron (1993) has shown that if

$$\int_{R^d} |\omega||\breve{F}(\omega)|d\omega \leq C, \tag{12.4}$$

then the approximation error with a single layer neural network consisting of n nodes is $O\left(\frac{1}{\sqrt{n}}\right)$.

A similar result is also available for Breiman's hinge approximations. The relevant condition is

$$\int_{R^d} |\omega|^2|F(\omega)|d\omega \leq C. \tag{12.5}$$

Assumptions (12.1,12.2), (12.4), or (12.5) are all ways of restricting the function class. Moreover, even though this is not explicit from the statements of the assumptions, the restriction can become more and more severe as d increases. Clearly, the fundamental question regarding these assumptions is: Are these kinds of restrictions on the functions satisfactory? A deep answer to this question would require the study of the structure of sets of functions defined by the assumptions, and the verification of how well this structure matches real-life approximation problems. As far as we know, no satisfactory answer of this kind is so far available.

13

Approximating the True Function by Minimizing the Empirical Error

Above we have seen that if one chooses the approximation consisting of n basis functions that minimizes the (squared) *expected error*,

$$\int_X (F(x) - \widehat{F}_n(x))^2 \mu(dx), \tag{13.1}$$

then the expected error is of the order $O(\frac{1}{\sqrt{n}})$. However, we do not know the expected error (since we neither know $F(x)$ nor $\mu(dx)$) and therefore cannot minimize it. In its place we only have the *training set*, which is chosen randomly and moreover may also contain noisy observations.

The expected error and the error of the approximation on the training set, called the *empirical error*, can be related as follows.

Suppose that the noise sequence e_i is an i.i.d. sequence, drawn from a zero mean distribution ν which, for ease of explanation, has bounded support. Clearly

$$\int_X (F(x) - \widehat{F}_n(x))^2 \nu(dx) = \int_{X\times Y} (y - \widehat{F}_n(x))^2 \nu(dy - F(x))\mu(dx) - V, \tag{13.2}$$

where $V = \int_Y y^2 \nu(dy)$ is the noise variance (assuming that Y is large enough). Thus, minimizing (13.1) is equivalent to minimizing

$$\int_{X\times Y} (y - \widehat{F}_n(x))^2 \nu(dy - F(x))\mu(dx). \tag{13.3}$$

This quantity is also unknown. However, an *estimate* of it is

$$\frac{1}{N}\sum_{k=1}^{N} (y_k - \widehat{F}_n(x_k))^2. \tag{13.4}$$

This is the (squared) *empirical error* on the training set.

The next question that arises is whether minimizing the empirical error (13.4) will lead to an approximation that nearly minimizes the true expected fitting error (13.1).

The following proposition is obvious:

Proposition. Suppose that the family $\{\text{EmpiricalError}_\theta(\cdot), \theta \in \Theta\}$ approximates uniformly well the family $\{\text{ExpectedError}_\theta(\cdot), \theta \in \Theta\}$ in the sense that for *every* $\theta \in \theta$,

$$|\text{ExpectedError}_\theta - \text{EmpiricalError}_\theta| \leq \epsilon \text{ for all } \theta \in \Theta.$$

Then if θ^* is an ϵ minimizer of $|\text{EmpiricalError}_\theta|$, i.e.,

$$|\text{EmpiricalError}_{\theta^*}| \leq |\text{EmpiricalError}_\theta| + \epsilon \text{ for all } \theta \in \Theta,$$

then θ^* is a 3ϵ-minimizer of $|\text{ExpectedError}_\theta|$, i.e.,

$$|\text{ExpectedError}_{\theta^*}| \leq |\text{ExpectedError}_\theta| + 3\epsilon \text{ for all } \theta \in \Theta.$$

This suggests that if the empirical error approximates the true error *uniformly* well over all $\theta \in \Theta$, then by minimizing the empirical error we can nearly minimize the true error.

The question then becomes: Does the empirical error nearly equal the true error uniformly over the particular class of minimizers sought? This cannot happen with probability one; we seek to determine whether it happens with a high probability exceeding $(1-\delta)$. This probability $(1-\delta)$ is called the *confidence* level.

To resolve whether uniform convergence occurs, we need to compute the probability of a "bad multi-sample," namely a multi-sample such that (13.4) and (13.3) are far apart by more than ϵ for some $\widehat{F}_n$:

$$q(N,n,\epsilon) := \mu^N \Big\{ (x_1, \ldots, x_N) : \sup(\alpha_i, \theta_i, i = 1, \ldots, n) \mid \frac{1}{N} \sum_{k=1}^{N} (y_k - \widehat{F}_n(x_k))^2 - \int_{X \times Y} (y - \widehat{F}_n(x))^2 \nu(dy - F(x)\nu(dx) \mid > \epsilon \Big\}. \tag{13.5}$$

This *statistical problem* has been well dealt with in the theory of learning, due to the pioneering work of Vapnik and Chervonenkis (1971). The net result is that when the class of functions over which uniformity is sought satisfies a certain property, related to the notion of Vapnik–Chervonenkis dimension, then with probability exceeding $(1-\delta)$ the empirical error is within ϵ of the true error uniformly over the class, when the number of samples is large enough, as given by an expression related to the Vapnik-Chervonenkis dimension, ϵ, and δ; see Vapnik (1982).

To cast the statistical problem as a problem of uniform convergence, let

$$g(x, y, \alpha_i, \theta_i, i = 1, \ldots, n) := (y - \widehat{F}_n(x))^2,$$

and

$$P(dx, dy) = \nu(dy - F(x))^2 \mu(dx).$$

Then the statistical issue can be reformulated as one of empirically estimating

$$\int_{X \times Y} g(x, y, \alpha_i, \theta_i, i = 1, \ldots, n) P(dx, dy) \tag{13.6}$$

uniformly well with respect to $\alpha_i, \theta_i, i = 1, \ldots, n$.

The interesting fact is that the unknown function F no longer appears explicitly in (13.6), it is hidden in the measure P. Thus, the problem can be addressed by resorting to tools from the theory of distribution free uniform convergence of empirical means, see Vapnik (1982).

The minimization of criterion (10.2) is now performed by resorting to its empirical version (13.4).

Assume now that the function class is such that we have an estimation method with a guaranteed level of accuracy ϵ and a level of confidence exceeding $(1-\delta)$.

Recall the definition of q in (13.5). Denote by

$$\epsilon(N, n, \delta)$$

the solution of the equation

$$q(N, n, \epsilon) = \delta$$

with respect to ϵ. Utilizing the approximation rate results alluded to earlier (assuming they apply) leads to the conclusion that the minimizer $\widehat{F}_n$ of (13.4) is such that

$$\|F - \widehat{F}_n\|^2 \leq O\left(\frac{1}{n}\right) + 2\epsilon(N, n, \delta)$$

with confidence exceeding $(1 - \delta)$. The number of basis terms n in the function approximation can be optimized by minimizing the right hand side.

The end result is that we finally obtain a bound for the error of the form

$$\|F - \widehat{F}_n\|^2 = \gamma(N, \delta)$$

where $\gamma(N, \delta)$ is interpreted as the level of accuracy guaranteed with probability exceeding $(1 - \delta)$.

Since we need to minimize (13.4) the issue again is how to obtain a greedy approximation; this time for the empirical error.

14

A Greedy Approximation Useful in Minimizing the Empirical Error

In the sequel it will be of interest to note that the greedy algorithm of Jones can also be applied to the empirical error and, in fact, can be further simplified by reducing the optimization step to finding an approximation of the mismatch $F - \widehat{F}_{n-1}$. The rate of convergence remains the same.

A Greedy Approach for Minimizing the Empirical Error

Initialize the sequence with:

$$\widehat{F}_0 = 0$$

Once $\widehat{F}_{n-1}$ has been determined, determine $\widehat{F}_n$ as follows:

$$(\alpha_n, \theta_n) = \text{argmin}_{\substack{0 \le \alpha \le 1 \\ \theta \in \Theta}} \|(\mathrm{F} - \widehat{\mathrm{F}}_{\mathrm{n-1}}) - \alpha \mathrm{h}(\theta)\|^2;$$

$$\widehat{F}'_n = \widehat{F}_{n-1} + \alpha_n h(\theta_n);$$

$$\widehat{F}_n = \frac{(F, \widehat{F}'_n)}{\|\widehat{F}'_n\|^2} \widehat{F}'_n \qquad (\text{ so that } F - \widehat{F}_n \perp \widehat{F}_n). \tag{14.1}$$

Proposition.

$$\|F - \widehat{F}_n\|^2 = O\left(\frac{1}{n}\right).$$

Proof. Fix $\gamma > 0$.

Since F belongs to the closure of the convex hull of $\{h(\theta), \theta \in \Theta\}$, there exists a suitable approximation $\sum_{i=1}^m c_i h(\theta_i)$ of F such that

$$(F - \widehat{F}_{n-1}, F - \sum_{i=1}^m c_i h(\theta_i)) < \gamma,$$

with $c_i \ge 0$, $\sum_{i=1}^m c_i = 1$. Observing that

$$(F - \widehat{F}_{n-1}, F - \sum_{i=1}^m c_i h(\theta_i)) = \sum_{i=1}^m c_i (F - \widehat{F}_{n-1}, F - h(\theta_i)),$$

we see that there exists an $\bar{i}$ in $[1, m]$ such that

$$(F - F_{n-1}, F - h(\theta_{\bar{i}})) < \gamma.$$

From this, taking into account the orthogonality condition (14.1) we obtain

$$\begin{aligned}(F - \widehat{F}_{n-1}, h(\theta_{\bar{i}})) = (F - \widehat{F}_{n-1}, h(\theta_{\bar{i}}) - F) + (F - \widehat{F}_{n-1}, F - \widehat{F}_{n-1}) \\ +(F - \widehat{F}_{n-1}, \widehat{F}_{n-1}) \\ > -\gamma + \|F - \widehat{F}_{n-1}\|^2.\end{aligned} \tag{14.2}$$

Set

$$\alpha_{\bar{i}} := \frac{(F - \widehat{F}_{n-1}, h(\theta_{\bar{i}}))}{\|h(\theta_{\bar{i}})\|^2}.$$

The following chain of inequalities follows from the orthogonality condition, the arbitrariness of γ and the boundedness of h:

$$\begin{aligned}\|F - \widehat{F}_n\|^2 &\le \|F - \widehat{F}'_n\|^2 \\ &\le \|(F - \widehat{F}_{n-1}) - \alpha_{\bar{i}} h(\theta_{\bar{i}})\|^2 \\ &\le \|F - \widehat{F}_{n-1}\|^2 - \frac{(F - \widehat{F}_{n-1} h(\theta_{\bar{i}}))^2}{\|h(\theta_{\bar{i}})\|^2} \\ &\le \|F - \widehat{F}_{n-1}\|^2 - \frac{(\|F - \widehat{F}_{n-1}\|^2 - \gamma)^2}{\|h(\theta_{\bar{i}})\|^2} \\ &\le \|F - \widehat{F}_{n-1}\|^2 - \frac{(\|F - \widehat{F}_{n-1}\|^2 - \gamma)^2}{b^2} \\ &\le \|F - \widehat{F}_{n-1}\|^2 \frac{b^2}{b^2 + \|F - \widehat{F}_{n-1}\|^2}.\end{aligned}$$

Thus

$$\frac{1}{\|F - \widehat{F}_n\|^2} \ge \frac{1}{\|F - \widehat{F}_{n-1}\|^2} + \frac{1}{b^2}$$

from which the result follows.

15

Function Estimation by Neural Networks

What we have seen is that if we have a number of data points which is sufficient for the function class, and if we fit the data well, then we nearly minimize the true error of using the estimate.

Utilizing this, Barron (1991) and Barron (1994) have shown that with N data points and n nodes, the error is:

$$\|F - \widehat{F}_n\|^2 = O(\frac{C_d^2}{n}) + O(\frac{nd}{N} \log N).$$

In particular, using $n \approx \sqrt{\frac{N}{d \log N}}$ nodes gives an L_2-error of $O(C_d \sqrt{\frac{d \log N}{N}})$. Note though that the constant C_d may be exponentially large in d.

16

Local Polynomial Methods

It appears that all existing methods consist of restricting the function class so as to satisfy a condition of the type needed for Jones' result to apply. To obtain an understanding of how broad or narrow this function class is, let us consider the best results obtainable with another function class, the set of functions of bounded second derivative. For this latter class not only is the worst case error known, but also an asymptotically optimal procedure is known, the *local polynomial method*; see Fan and Gijbels (1996).

Let us begin by considering functions of a single variable. The model for the observations is:

$$y_i = F(x_i) + \sigma(x_i)e_i.$$

The assumptions are that

(i) $|F''(x)| \leq C$.

(ii) x_i are iid with density $p(x)$.

(iii) e_i are iid with mean 0 and variance 1.

(iv) $p(x)$ and $\sigma(x) > 0$ are continuous.

The main assumption to focus on is the boundedness of $|F''(x)|$.

The problem is to estimate $F(x)$ for a specified x, given the observations (x_i, y_i) for $1 \leq i \leq n$. (The results carry over to the L_2-norm).

Let us consider the class of linear estimators L of the form:

$$\widehat{F}(x) = \sum_{j=1}^{N} w_j(x; x_1, x_2, \ldots, x_N) y_j.$$

Let us define the *linear minimax risk*

$$R = \inf_{\widehat{F} \in L} \sup_{F} E\left[(\hat{F}(x) - F(x))^2 | x_1, \ldots, x_N\right].$$

Fan and Gijbels (1996) have shown the following result:

Theorem. As the number of observations N tends to infinity,

$$R = \frac{3}{4} 15^{-1/5} \left[\frac{\sqrt{C}\sigma^2(x)}{Np(x)}\right]^{\frac{4}{5}} (1 + o(1)).$$

There is also a procedure which sharply attains this bound; it is the local polynomial method.

Let h be the "bandwidth." Let

$$K(u) = \frac{3}{4}(1-u^2)^+$$

denote the Epanechnikov kernel. Given x, we locally fit a linear function by minimizing over a, b the weighted squared error criterion:

$$\sum_{i=1}^{N}(y_i - a - b(x_i - x))^2 K\left(\frac{|x_i - x|}{h}\right).$$

The estimate at x is then simply a. This estimate is asymptotically optimal as $N \to \infty$ when h is chosen as follows:

$$h = \left[\frac{15\sigma^2(x)}{Np(x)}C^2\right]^{\frac{1}{5}}.$$

This shows precisely the asymptotic order of the the minimax risk when the estimate is linear.

The linearity restriction is not severe since even in the nonlinear class

$$R_{\text{nonlinear}} \geq (0.894)^2 R.$$

The above result generalizes to high dimensions also. The result is

$$R_{\text{nonlinear}} \geq C_d \left[\frac{1}{N^{\frac{4}{d+4}}}\right].$$

This shows that the best convergence rate is very poor in high dimensions, when the only information we have is that the second derivative of the function F is bounded.

This suggests that perhaps the assumption $\int_{R^d} |\omega||F(\omega)|d\omega < C$ on the function class could require excessive smoothness as the dimension d is increased. Some evidence for this is the sufficient condition for it given by Murata (1996):

$$f \in C_0^{d+1}(R^d) \cap L_\infty(R^d).$$

Anther way to view these results is as follows. The number of data points needed to densely fill the input space X grows exponentially with the input dimension (the "curse of dimensionality"). Hence, in high dimensional spaces (say, $d \geq 10$), it is unrealistic to assume that data are dense, and this immediately restricts the possibility of using local approximation methods. To deal with this difficulty, the only way out may be to suitably restrict the function class to which F belongs so that even data points x_i far away from x can contribute to forming a suitable estimate of $F(x)$.

17

The Training Algorithm Required in Function Estimation

Note that up till now we have been assuming that we can obtain a minimizer or near minimizer of the empirical error. For this we need a minimization algorithm. We need to assess how difficult or easy this task is.

Note that the algorithm for minimizing the empirical error is at the heart of the training procedure. Everything else dealt with so far is just theoretical support for concluding that once we have obtained a minimizer (or near minimizer) of the error on the training set, then we can be highly confident that we have fitted the true unknown function well.

We therefore turn to the algorithmic aspects of the problem of function estimation, i.e., an algorithm to obtain a function which nearly minimizes the empirical error (13.4).

For neural networks, the popular method advocated seems to be to obtain the derivatives by the well-known back propagation algorithm, and then to use it in an iterative optimization procedure. One such optimization procedure is the gradient method, another is the Levenberg-Marquardt method. The difficulty is that the function to be minimized is parameterized in such a way that it has several local minima; see the cover of Bertsekas (1995). Thus there is no guarantee of obtaining a global minimum. Even the convergence of the training algorithm cannot be assured.

Without a solution to the minimization problem, which is at the heart of the training procedure, no satisfactory theory can exist. Our view, therefore, is that finding a solution to this problem is today the main issue in function estimation. In the rest of the paper, we concentrate on this algorithmic problem and discuss some difficulties connected with it.

It is here that the Greedy algorithm for minimizing the empirical error is useful. We can use it to solve the problem of fitting a function to reduce the empirical error problem of minimizing the cost

$$\frac{1}{N}\sum_{k=1}^{N}(F(x_k) - \widehat{F}_{n-1}(x_k) - \alpha h(x_k, \theta))^2 \tag{17.1}$$

that only depends on $p+1$ parameters (where $\theta \in R^p$).

Note that the Greedy approach has delivered a powerful and surprising result. Since $\|F - \widehat{F}_n\|^2$ tends to zero as fast as $\frac{c}{n}$, a restricted (and dimension independent!) number of terms is needed to achieve a good approximation. Since the number of parameters to be estimated in each term is $p+1$, the total number of parameters is $n(p+1)$. It can therefore be expected from learning theory that a reasonable number of data points will achieve a good estimation result.

18

Difficulties in Minimizing $\frac{1}{N}\sum_{k=1}^{N}(F(x_k) - \widehat{F}_{n-1}(x_k) - \alpha h(x_k,\theta))^2$

We now show that the problem (17.1) is in general a nonconvex optimization problem and discuss the issue of formulating convex optimization problems equivalent to it. Let

$$V := \{(\alpha h(x_1,\theta),\ldots,\alpha h(x_N,\theta))^T, \qquad 0 \le \alpha \le 1, \qquad \theta \in \Theta\}.$$

V is a closed and, in general, nonconvex set.

The optimization problem (17.1) can be rewritten as

$$\min_{v\in V} \|e - v\|_E^2, \tag{18.1}$$

where $e := [(F(x_1) - \widehat{F}_{n-1}(x_1)),\cdots,(F(x_N) - \widehat{F}_{n-1}(x_N))]^T$ and $\|\cdot\|_E$ is the empirical 2-norm. We want to show the general result that not only is problem (18.1) nonconvex, but also that no continuous convex program exists which is equivalent to it. A continuous convex program is a program problem of the form

$$\min_{\alpha,\theta,w} l(\alpha,\theta,w,e) \tag{18.2}$$

subject to

$$g(\alpha,\theta,w,e) \le 0,$$

where l and g are continuous functions, convex in (α,θ,w) for any given e.

Problem (18.2) is said to be *equivalent* to (18.1) if the (α,θ) coordinate of every optimal solution for (18.2) is also optimal for (18.1).

Proposition. *For some e, the set of minimizers over V of $\|e-v\|_E^2$ of problem (18.1) has cardinality greater than one.*

Proof. Consider v_1, $v_2 \in V$ and $\lambda \in (0,1)$ such that $\lambda v_1 + (1-\lambda)v_2 \notin V$, (such v_1, v_2, and λ exist because V is nonconvex). Clearly, there exists a closed ball $\overline{B}$ of radius $\rho > 0$ and centered at $\lambda v_1 + (1-\lambda)v_2$ which is essentially disjoint with V.

Define:

$$\mathcal{B} := \{B = \text{ closed ball such that } \overline{B} \subseteq B \text{ and which is essentially disjoint with } V\}.$$

The following three facts can be established:

i) There exists a constant c such that the radius of all balls $B \in \mathcal{B}$ is upper bounded by c;

ii) Set $R := \sup\{r(B), B \in \mathcal{B}\}$, where $r(B)$ is the radius of ball B. Then, there exists a ball $\widetilde{B} \in \mathcal{B}$ with radius R;

iii) $\widetilde{B} \cap V$ has cardinality at least 2.

Let e = center of $\widetilde{B}$. Then the set of minimizers of the problem (18.1) has cardinality greater than 1. This completes the proof.

Fix now a vector $\overline{e}$ such that the set of minimizers of the problem (18.1) has cardinality greater than 1. To proceed we make the following additional assumption: For $\overline{e}$, the set of minimizers contains two disconnected components V_1^* and V_2^*.

Proposition. *No continuous convex program of the form (18.2) exists that is equivalent to problem (18.1).*

Proof. Define

$$A_i := \{(\alpha,\theta), 0 \le \alpha \le 1, \theta \in \Theta \text{ subject to } [\alpha h(x_1,\theta) \cdots \alpha h(x_N,\theta)]^T \in V_i^*\}, \text{ for } i = 1,2.$$

Clearly, A_1 and A_2 are disconnected. Consider now a program of the form (18.2) and suppose that it is equivalent to problem (18.1). Since the program (18.2) is convex, the set of its minimizers is convex and, hence connected. Project this set along its (α,θ) coordinates and name the resulting set $A(e)$ (the argument e indicates that the set $A(e)$ depends on the vector e in program (18.2)). Clearly, $A(e)$ is a connected set.

Consider in particular the set $A(\overline{e})$. Since $A(\overline{e})$ is connected and sets A_1 and A_2 are disconnected, either $A(\overline{e}) \cap A_1 = \phi$ or else $A(\overline{e}) \cap A_2 = \phi$. Suppose $A(\overline{e}) \cap A_1 = \phi$. This implies that

$$\inf_{\substack{(\alpha,\theta)\in A_1 \\ w:\, g(\alpha,\theta,w,\overline{e})\le 0}} l(\alpha,\theta,w,\overline{e}) > \inf_{w:\, g(\overline{\alpha},\overline{\theta},w,\overline{e})\le 0} l(\overline{\alpha},\overline{\theta},w,\overline{e}), \tag{18.3}$$

where $(\overline{\alpha},\overline{\theta}) \in A(\overline{e})$.

Fix now a vector $v_1^* \in V_1^*$ and set

$$\tilde{e} = (1-\epsilon)\overline{e} + \epsilon v_1^*.$$

It is easy to see that for any $\epsilon \in (0,1)$, the program (18.2) admits v_1^* as its unique minimizer and so the corresponding set of optimal solutions in the (α,θ) space is contained in A_1. On the other hand, due to the continuity of function ℓ, equation (18.3) entails that for ϵ small enough all points in A_1 will be strictly suboptimal for program (18.2). From this we see that program (18.2) is in fact not equivalent to (18.1) and this leads to a contradiction.

Note that the above result implies that problem (18.1) cannot be reformulated as a continuous convex program. One could however conceive formulating an array of optimization problems and then taking the best solution among them. A side result of the proof is that the minimal number of optimization problems needed equals the number of disconnected optimal sets $V_1^*, \ldots, V_m^*$.

Example

The number of disconnected optimal sets $V_1^*, \ldots, V_m^*$ and, therefore the number of optimization problems, may in general be very large indeed. For ease of explanation, assume $d = 2$ and that the $h(\theta)$'s are ridge functions with an abrupt saturation function:

$$sat(\zeta) = \begin{cases} 0, & \zeta \le 0 \\ 1, & \zeta > 0. \end{cases}$$

Then, it can be shown that the number of disconnected sets is about $\binom{N/2}{2}$. In the d-dimensional case: it is about $\binom{N/2}{d}$. One can extend this to consider continuous, steep saturation functions.

19

A Finite Algorithm for Optimization

We now show that there exists a finite algorithm for solving the optimization problem (18.1). The method consists of reformulating it as a large number of quadratic programming programs. Thus there exists a training algorithm guaranteed to converge. However, its complexity can be very high.

Let us consider hinge functions. Given N points $x_1, \ldots, x_N \in X \subseteq R^d$, there are at most $\binom{N}{d}$ ways to separate them with hyperplanes (in fact, there are exactly $\binom{N}{d}$ ways if no $d+1$ points exist belonging to the same plane). For each possible division of the data points $\{x_k\}$ into two sets S_- and S_+, one can define the following convex quadratic program:

$$\min \sum_{x(k) \in S_-} (e(k) - v_-^T x(k) - \alpha_-)^2 + \sum_{x_k \in S_+} (e(k) - v_+^T x(k) - \alpha_+)^2$$

subject to

$$\begin{aligned} v_-^T x(k) + \alpha_- &\geq v_+^T x(k) + \alpha_+, \qquad \forall x(k) \in S_- \\ v_-^T x(k) + \alpha_- &\leq v_+^T x(k) + \alpha_+, \qquad \forall x(k) \in S_+, \end{aligned}$$

for finding the best "max" hinge function, and a similar quadratic program for the "min" hinge function. The optimal solution is then obtained by a comparison of the solutions of all the $\binom{N}{d}$ quadratic programs.

While this is in fact a finite procedure guaranteed to terminate in an optimal solution, its complexity is very high due to the large number $\binom{N}{d}$ of quadratic programs that need to be solved.

20

Concluding Remarks

In our view, the major open problem currently is that of finding a tractable optimization algorithm for training the function estimator on the data. We have shown that there do exist finite algorithms for fitting the empirical error, i.e., for training. They are thus guaranteed to converge; however their computational complexity can be very high. We have also shown that one cannot reduce the function fitting problem to a convex program in a straightforward way.

Another issue needing investigation is the nature of restrictions on the function class, both with regard to theoretical understanding of their restrictiveness, as well as practical adequacy.

Finally, we are not aware of any work which explores the connection between function estimators and their ability to solve high level problems of interest, as was done for example in case of perceptrons. This is a very interesting area for future research.

Acknowledgments

This material is based upon work partially supported by the National Science Foundation under Contract No. SBC-MIT-5710000330-NSF. Any opinions, findings, and conclusions are those of the authors and do not necessarily reflect the views of the above agencies.

Barron, A. R. (1991): "Complexity regularization with applications to artificial neural networks." In Roussas, Ed., *Nonparametric Functional Estimation and Related Topics*, pp. 561–576. Kluwer Academic, The Netherlands.

Barron, A. R. (1993): "Universal approximation bounds for superpositions of a sigmoidal function." *IEEE Transactions on Information Theory*, **39:3**, pp. 930–945.

Barron, A. R. (1994): "Approximation and Estimation Bounds for Artificial Neural Networks." *Machine Learning*, **14**, pp. 115–133.

Bertsekas, D. P. (1995): *Nonlinear Programming*. Athena Scientific Press, Belmont.

Block, H. D. (1962): "The Perceptron: A model for brain functioning." *Reviews of Modern Physics*, **34**, pp. 123–135.

Breiman, L. (1993): "Hinging hyperplanes for regression, classification and function approximation." *IEEE Transactions on Information Theory*, **39:3**, pp. 999–1013.

Cybenko, G. (1989): "Approximation by superpositions of a sigmoidal function." *Mathematics of Control, Signals, and Systems*, **2**, pp. 303–314.

Fan, J. and J. Gijbels (1996): *Local Polynomial Modelling and Its Applications*. Chapman & Hall, Madras.

HORNIK, K., M. STINCHCOMBE, and H. WHITE (1989): "Multilayer feedforward networks are universal approximators." *Neural Networks*, **2**, pp. 359–366.

JONES, L. K. (1992): "A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training." *The Annals of Statistics*, **20:1**, pp. 608–613.

MINSKY, M. and S. PAPERT (1988): *Perceptrons: An Introduction to Computational Geometry*. MIT Press. Expanded Edition.

MURATA, N. (1996): "An integral representation of functions using three-layer networks and their approximation bounds." *Neural Networks*, **9:6**, pp. 947–966.

ROSENBLATT, I. (1961): *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, Washington, DC.

VAPNIK, V. N. (1982): *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, New York, NY.

VAPNIK, V. N. and A. CHERVONENKIS (1971): "On the uniform convergence of relative frequencies of events to their probabilities." *Theory of Prob. and its Appl.*, **16:2**, pp. 264–280.