UNLIMITED

RSRE
MEMORANDUM No. 4453

# ROYAL SIGNALS & RADAR ESTABLISHMENT

## FUNCTIONAL APPROXIMATION BY FEED-FORWARD NETWORKS: A LEAST-SQUARES APPROACH TO GENERALISATION

Author: A R Webb

PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
RSRE MALVERN,
WORCS.

RSRE MEMORANDUM No. 4453

UNLIMITED

**Royal Signals and Radar Establishment**
**Memorandum 4453**

# Functional Approximation by Feed–forward Networks : A Least–squares Approach to Generalisation

*Andrew R. Webb*

11th January 1991.

## Abstract

This paper considers a least–squares approach to function approximation and generalisation. The particular problem addressed is one in which the training data is noiseless (perhaps specified by an assumed model or obtained during some calibration procedure) and the requirement is to define a mapping which approximates the data and which generalises to situations in which data samples are corrupted by noise. The least–squares approach produces a generaliser which is the vector of posterior probabilities and has the form of a Radial Basis Function network for a finite number of training samples. The finite sample approximation is valid provided that the noise on the expected operating conditions is large compared to the sample spacing in the data space. In the other extreme of small noise perturbations, it is shown that better generalisation will occur if the training error criterion (the sum–square error on the training set) is modified by the addition of a specific regularisation term. This is illustrated by an approximator which has a feed–forward architecture and applied to the problem of point–source location using the outputs of an array of receivers in the focal–plane of a lens.

INTENTIONALLY BLANK

# 1 Introduction

Connectionist models based on feed-forward networks (for example, multilayer perceptrons (MLPs) [25] and radial basis function networks [4] (RBFs)) have been used with some success when operating as static pattern classifiers on a wide range of problems. Such networks perform a nonlinear transformation from an $n$-dimensional input space to the $n'$-dimensional output space via a characterisation space defined by the outputs of the (final layer of) hidden units in which a specific feature extraction criterion is maximised [18, 32]. This feature extraction criterion may be viewed as a nonlinear multidimensional generalisation of Fisher's linear discriminant function. Training the network for a pattern classification task consists of presenting data vectors as input, together with class labels at the output of the network, suitably coded, and minimising an error criterion. For a 1-*from*-$n'$ target coding scheme, and the usual sum-square error criterion, the outputs of a trained network approximate the Bayes discriminant vector, the probability of a class given the input to the network [18].

An alternative viewpoint to the pattern classification description on the operation of adaptive, feed-forward layered networks such as the multilayer perceptron is that they perform well for certain tasks by exploiting their modelling flexibility to create an implicit interpolation surface in a high-dimensional space [4, 16]. In fact, it may be shown that multilayer feed-forward networks with a single hidden layer are universal approximators in that an arbitrary function can be approximated arbitrarily well [13, 28]. However, in a practical problem, the mapping we wish to approximate is not known continuously but it is usually defined by a finite set of points in $\mathbb{R}^n \otimes \mathbb{R}^{n'}$ defined by a training set. Specifically, in mapping a finite set of $P$, $n$ dimensional 'training' patterns to the corresponding $n'$ dimensional 'target' patterns, $f : \mathbb{R}^n \rightarrow \mathbb{R}^{n'}$ one may think of this map as being generated by a 'graph' $\Gamma \subset \mathbb{R}^n \otimes \mathbb{R}^{n'}$. The input and target pattern pairs are points on this graph. The learning phase of adaptive network training corresponds to the optimisation of a fitting procedure for $\Gamma$ based on knowledge of the data points. This is curve fitting in the generally high dimensional space $\mathbb{R}^n \otimes \mathbb{R}^{n'}$. Thus *generalisation* becomes synonymous with *interpolation* along the constrained surface which is the 'best' fit to $\Gamma$ [34]. The Radial Basis function network [4] was introduced simply to make this point more explicit, but it also applies to networks such as the multilayer perceptron. It is clear that, by analogy with curve fitting in one or two dimensions, one can create an interpolation surface which is guaranteed to pass through every point in a finite training set provided that the model is of a sufficiently high order (e.g. sufficient numbers of hidden units equivalent to a sufficient number of adjustable parameters). Thus, although we can approximate the mapping arbitrarily accurately (as determined by the sum-squared error on the training set) by increasing the number of hidden units (and consequently the number of adjustable parameters), the error on an unseen test set (the generalisation error) is not guaranteed to decrease. In fact, it is likely to increase as the network progressively begins to model noise in the training data, rather than the structure of the data. This is an incorrect strategy for real data which is confused by extrinsic and intrinsic noise effects and corresponds to overtraining a network. Often a large amount of prior knowledge is required to allow a fitting surface to be produced which is just smooth enough to fit the structure in the data thus allowing good generalisation performance, without being over-complex to permit the fitting of noise on top of the data.

One example of feed-forward networks' being used to provide an interpolation surface is in the context of nonlinear prediction of time series [14, 19]. Indeed, adaptive network techniques have been used recently [4, 5, 8] with some success to predict the behaviour of chaotic time series – deterministic sequences whose second order persistent statistics seem to indicate that they are random. This success stems from the ability of adaptive networks to produce an interpolation surface which approximates the actual nonlinear map which generated the data. Thus, adaptive networks may be applied to time series prediction, as long as the observed time series is generated by an underlying iterative mapping, if the mapping itself is 'smooth' enough to allow an interpolation surface to be constructed.

The objective of many signal processing tasks is to determine some property $f(x)$ of a given data sample, $x$. In a pattern classification task the objective may be to assign $x$ to a particular class or to calculate the loss in assigning $x$ to a given class. In a bearing estimation problem, $x$ would represent the outputs as a function of time of an array of sensors and the problem is to determine the number and the positions of the sources present in the scene. Whatever the problem, the quantity which we wish to evaluate is a function, $f$, of the data sample $x$. The precise form of this function may be known analytically from the physical principles involved (at least for the special case of noiseless data, $x$) or the function $f$ (or a noisy version of $f$) may be known only at sample points, $x_i$, gathered during some training procedure. Whatever way in which $f$ is specified, we wish to derive an approximation to it which is valid at points not necessarily in the training set and which are expected to occur in a practical application (on an unseen test set). This is the problem of *generalisation*. Of course, if the data is noiseless (and the expected data in operation is noiseless) and the function is single–valued, then some form of strict interpolation between training points may provide a satisfactory approximation. However, in most pattern classification tasks, noise will in general be present either in the data used to derive the approximator (the training set) or in the expected operating conditions (the test set). It is important to understand how this noise arises when deriving an approximation to $f$ since the physical conditions should play an important part in the specification of any generaliser. These remarks can be illustrated by a few simple examples.

1. In some pattern classification problems noise may occur on the data points, $x_i$, even though the data is correctly classified. Thus, the training data comprises the set $\{(x_i + n_i, f_i),\ i = 1, \ldots, P\}$ for $P$ training patterns. For example, in the analysis of chemical compounds in which $x_i$ is a chromatogram, and $f_i$ the compound label, $n_i$ is the noise on the measuring equipment. The problem is to construct a function $g(z)$ which, given a chromatogram $z$ not in the training set, estimates the class to which the data belongs. The data may be multi–valued (with several observation values $f$ for a given data sample $z = x + n$ if the noise on the data causes the distributions to overlap) and the problem here is to estimate the expected value for a given data sample, $z$.

2. Alternatively, noise may occur on the observations $f_i$. In a pattern classification problem this would correspond to incorrect labelling of the data. In the example above, if the rôles of $f_i$ and $x_i$ were interchanged (so that the data samples are the class labels and the observations are the measurements) the problem becomes one of estimating the spectral response given the compound type. The data is multi–valued since there are many observations for a given compound. Thus a function could produce the expected value for a given compound. Note that in the classical least–squares problem, there is an underlying assumption that all the errors are confined to the observations.

3. Generally noise will occur on both $x_i$ and $f_i$. For example, in the problem of time series prediction (in which there is noise on the time series, $\{t_p\}$) the data values comprise a time history of, say, $n$ samples, $x_i = (t_{i+1}, t_{i+2}, \ldots, t_{i+n})$ and the observations are the next value in the time series, $f_i = t_{i+n+1}$. For the case of linear prediction, the method of total least squares [11] is a technique which is appropriate when there are errors on both the observations and the data.

4. In this final example, we assume that we have a set of noiseless data $\{(x_i, f_i),\ i = 1, \ldots, P\}$. This may be generated using some known model for the function $f(x)$ or it may be obtained during a calibration procedure of some equipment in which outputs are measured in response to a known input. Integration of the outputs is then performed to give data at a very high signal-to-noise ratio. The problem is to generalise the function to samples which are not in the data set and which are possibly corrupted by noise. For example, given the bandlimited image of an object of finite support, it is possible to recover the object exactly (assuming knowledge of the imaging operator). Thus the object is a single-valued function of the image. This function may be expressed as an summation of prolate spheroidal wave functions. In practice, images will be corrupted by noise, so we wish to generalise to unknown images, assuming knowledge of the noise process.

The problem is to design a function $g(z)$ which for noisy data samples, $z = x + n$ approximates $f(x)$. The data value $z$ may not lie in the space of values $x$ – *i.e.* the noise may perturb the value out of the manifold on which the training data lies. The form of the approximator $g$ will depend on the distribution of $x$ and the noise probability density function of the expected operating conditions. Thus, we can derive several approximators which can be applied in different operating conditions. The advantages of using a model or equivalently noiseless data, together with prior knowledge of the operating conditions is that there is no need to gather training data representative of the different conditions. It is sufficient to have a single training set representative on one operating condition (no noise) and to include the effects of different conditions in the formulation of the approximator. Also, the model used to generate the data may incorporate known physical principles and thus prior knowledge may be included in the data set.

One property of the approximating function $g$ which we require is that for $z = x + n$ then $g(z) \to f(x)$ as $n \to 0$. We n.ay also require that $g$ be an unbiased estimate, *i.e.* $\overline{g(z)} = f(x)$. It may be difficult to satisfy these constraints if $f(x)$ were not known continuously as a function of $x$. Even if it were, there will be many function which satisfy one or both of these conditions, and we may choose one which minimises a cost function. The particular cost function which we shall consider is the least-squares error measure.

The particular problem which we address is that illustrated by the fourth example above, namely that we have a model or theoretical solution for noiseless data and we are required to generalise to situations in which the data will be corrupted by noise. In Section 3 we derive a minimum mean-square error approximation which generalises the training data to data corrupted by noise. For a finite set of training samples, this approximation takes the form of a radial basis function network provided that the noise is large compared to the sample spacing. In the other extreme of small noise, a particular form is assumed for the approximation function. In this case there are two types of error to consider : systematic errors arising from the function's being only an approximation to the true mapping and errors which are due to the sensitivity of the function to noise on the data values. The purpose of the paper is to introduce an approach for training a generaliser to achieve a given functional transformation which minimises the total error (the sum of the systematic and the noise errors) between the observations and the approximations. Thus, the generaliser is trained to make the functional mapping robust to noise on the training data whilst still achieving small systematic errors. This is important in application in which the data values are likely to be corrupted by noise before processing. For example, in some inverse problems in imaging or synthetic aperture radar, the point-spread function of the optics or sensor may be known, but the data is corrupted by measurement noise. To obtain a generaliser which has the desired property of being robust to noise on the data, a regularisation term is derived. This is added to the usual error expression and the combined quantity minimised during training of the network. This regulariser is not based explicitly on a notion of surface fitting by, for example, imposing smoothness conditions on the fitting surface [1, 20] (though of course it is intimately related to this), but is derived by demanding minimal sensitivity of the function to noise on the data whilst still maintaining closeness of fit. The application which we shall use to illustrate the techniques developed in this paper is that of point-source location given the outputs (possibly corrupted by noise) of an array of sensors. Specifically, we consider the outputs of a focal-plane array radar. The particular functional form we consider is a feed-forward network architecture and the network is designed to fit the data and to be robust to noise. A detailed study of the application of feed-forward layered networks to this problem is presented in [30].

Another issue which is important for many potential applications is that of fault tolerance, *i.e.* designing a network so that the functional transformation performed by the network is robust to noise on the *weights* or even to failure of some of the links between nodes or failure of the nodes themselves. This problem has received some attention in the literature [27] but it is not one we address in this paper. However, the basic strategy presented in Section 3 may be developed to apply to the problem of noise on the weights (as opposed to noise on the inputs).

The paper is organised as follows. Section 2 considers the problem of functional approximation. Two specific approaches are described (regularisation theory and parametric modelling) and these are illustrated by some 1-dimensional examples. Section 3 considers a least squares approach to approximation and generalisation. This produces an approximating function which gives the expected observation for a given data sample. Section 4 considers the application to point-source location using a focal-plane array of receivers. Finally, Section 5 concludes with a discussion of the issues raised in this paper and summarises the main results.

## 2 Networks and Functional Interpolation

In this section we present some functional interpolation preliminaries. There are infinitely many surfaces which interpolate a set of data points. Further, if there is noise on the data then strict interpolation is inappropriate. There are two basic approaches to functional interpolation which we consider in this section and we illustrate these with some simple one-dimensional examples. One approach, described in Section 2.1 is to determine the surface which minimises a regularisation term subject to a constraint on the fitting error. This regularisation term may, for example, demand a fitting surface with minimum curvature and its minimisation leads to a particular functional form for the fitting surface. However, the choice of the regulariser requires some prior knowledge concerning the fitting surface. The constraint parameter, which controls the tradeoff between the closeness of fit of the surface to the data points and the smoothness of the surface, may be determined from the data using some form of cross-validation or data-driven smoothing.

The second approach, described in Section 2.2, is to specify the functional form for the fitting surface using, for example, polynomials or sums of Gaussians. The specific form adopted may be chosen using some prior knowledge of the expected form of the 'true' surface or simply for its computational merits. Once the functional form has been decided, the complexity (number of free parameters) must be chosen. This depends on the dimensionality of the data, the number of data points characterising the transformation and the noise on the data and determining the balance between these quantities is a problem which has been studied widely in the pattern recognition literature (for example, [10, 24]). If there are too many free parameters then noise superimposed in the data may be modelled; too few, and the model fails to capture the structure in the data. However, with the functional form and the complexity specified, determination of the values of the parameters of the model ("training") is usually a straightforward task.

## 2.1 Regularisation Theory Approach

Regularisation theory, introduced by Tikhonov ([29], see also [21]) for solving ill-posed problems, is one method for addressing the problem of generalisation. A problem is ill-posed when either the solution does not exist, the solution is not unique or the solution is not stable (does not depend continuously on the initial data). The main idea of regularisation theory is to restrict the class of admissible solutions by introducing suitable *a priori* constraints on the possible solutions. Standard regularisation theory imposes the constraints on the problem by a variational principle with a cost function. It provides a means of solving an equation of the form

$$Az = u \tag{1}$$

for an operator $A$ and data $u$ to give a solution for $z$ which is stable. For example, inverse problems constitute a broad class of problems in which the object or phenomenon in question is characterised by an element $z$ belonging to a set $F$. Usually, $z$ is not observed directly, but rather a quantity $u = Az$ (where $u \in AF$, where $AF$ is the image of the set $F$ under the mapping executed by the operator $A$). The operator $A$ may be an imaging operation and then the problem may be to restore an object from its bandlimited image. Equation (1) has a solution only for those elements $u$ which belong to the set $AF$. However, because of noise on the data, the quantity $u$ is known only approximately and may not belong to the set $AF$. Even if a solution does exist, it may not be stable since the inverse operator $A^{-1}$ may not be continuous.

In such cases as the above, an approximate solution to the problem is sought. This is achieved by the introduction of a regularising term of the form $\alpha\Omega(z)$, where $\alpha$ is the *regularisation parameter* and $\Omega$ is the *stabilising functional* and a solution is sought which minimises the *smoothing functional* $M^{\alpha}(z,u)$ defined by [29]

$$M^{\alpha}(z,u) = \rho^{2}(Az,u) + \alpha\Omega(z) \tag{2}$$

where $\rho$ is a metric on the space to which $Az$ belongs. Thus, we take as an approximate solution of Equation (1), a solution which for $z$ which minimises $M^{\alpha}(z,u)$.

There are two ways of approaching this problem. One approach is to seek a solution which minimises $M^\alpha(z,u)$ and then to determine the regularisation parameter $\alpha$ from supplementary information relating to the particular problem [29]. Alternatively, we may view the problem as a variational problem in which the stabilising functional, $\Omega(z)$ is minimised subject to the constraint that

$$\rho(Az,u) = \delta \tag{3}$$

where the parameter $\delta$ characterises the error in the initial data, $u$. This is termed *Morozov's Discrepancy Principal.* We may use the method of Lagrange multipliers to minimise the functional $M^\alpha(z,u)$ given by Equation (2) and choose $\alpha$ so that the constraint (3) is satisfied.

One application of regularisation theory is to surface fitting in which it is required to find a hypersurface $g(x)$ given the values $f_i \in \mathbb{R}^{n'}$ at a set of points $x_i \in \mathbb{R}^n$, such that $f_i = f(x_i)$. Thus the data points $\{(x_i, f_i), i = 1, \ldots, P\}$ lie in the space $\mathbb{R}^n \otimes \mathbb{R}^{n'}$. This is an ill-posed problem in the sense that there are an infinite number of solutions and therefore some constraint must be applied. This is usually in the form of a smoothness constraint. Thus, in Equation (2), $A$ is the sampling operator, $\rho$ is the Euclidean norm and the stabilising functional is of the form

$$\Omega = ||Pf||^2 \tag{4}$$

where $P$ is usually a linear differential operator. Therefore, Equation (2) gives

$$M^\alpha(g,f) = \sum_{i=1}^{P} ||f_i - g(x_i)||^2 + \alpha ||Pf||^2 \tag{5}$$

where $u$ is the set of function values, $f_i$. Here, $\alpha$ controls the compromise between the smoothness and the fidelity to the data. For example, minimising the stabilising functional, $||Pf||^2$ subject to the constraint that

$$\sum_{i=1}^{P} ||f_i - g(x_i)||^2 = 0 \tag{6}$$

gives the smoothest interpolator to the data (the approximating function $g$ satisfies $g(x_i) = f_i$). In one dimension with $g : \mathbb{R} \rightarrow \mathbb{R}$ and $P$ a linear differential operator with real coefficients, *i.e.*

$$P = \sum_{j=0}^{m} a_j D^j \qquad m > 0, a_m \neq 0 \tag{7}$$

with $D \equiv \frac{d}{dx}$, then the solution which minimises (5) is an $L$-spline [15]. This consists of piecewise solutions of $(2m-1)$ order polynomials with continuity of derivatives up to the $(2m-2)$. The smoothing functional $M^\alpha$ is also equivalent to the $K$-functional introduced in spline theory which is a measure of how well the function $f$ can be approximated by smoother functions while maintaining a control on the size of the $m$th derivative of the approximator [26].

However, an important question is how smooth should the reconstructed function be? The particular choice for $m$ determines the continuity of the solutions and their smoothness

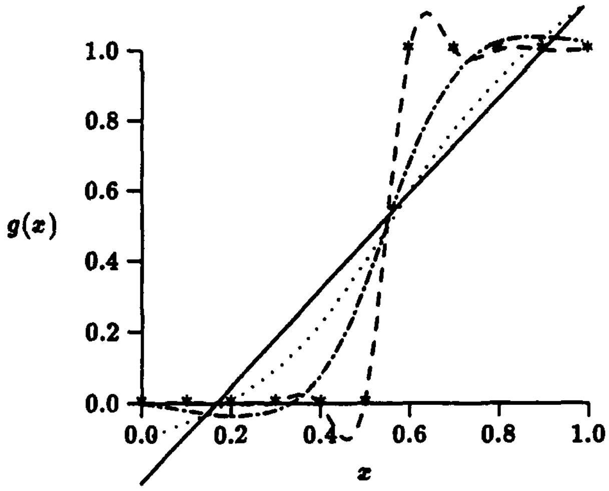

Figure 1: cubic spline approximations for $\alpha = 10.0$ (solid line), $10^{-2}$ (dotted line), $10^{-3}$ (dot–dash) and $10^{-6}$ (dashed)

increases with increasing $m$. For $m = 2$ and $a_0 = a_1 = 0,\ a_2 = 1$, the stabilising functional, $\Omega$, is given by

$$\Omega = \int_a^b \left[\frac{d^2g(x)}{dx^2}\right]^2 \tag{8}$$

This measures the strain energy of bending in a thin flexible beam of infinite extent and the Euler equation is

$$\frac{d^4g}{dx^4} = 0 \tag{9}$$

giving cubic splines as the solution.

In order to illustrate the use of regularisation theory we shall consider fitting a set of data points, $\{(x_i, f_i), i = 1, \ldots, P\}$ in $\mathbb{R} \otimes \mathbb{R}$.

EXAMPLE 1 Figure 1 shows several approximations to a step function, which is characterised by 9 data points (each marked by an asterisk). We have assumed a stabilising functional of the form (8), giving cubic spline solutions, and plotted solutions for several values of $\alpha$. The smallest value of $\alpha$ gives a very good fit to the data and the largest gives the smoothest approximation (a straight line approximation).

EXAMPLE 2 This example and the following one introduce the idea of fitting when there is noise on the data. This can occur in several ways and we illustrate two of them. In both these examples the data points lie on a $\sin(x)/x$ shape curve and are corrupted by noise. In Figure 2 the data points are $(x_i, f_i+n_i)$ where

$$f_i = \frac{\sin(x_i - 0.5)}{(x_i - 0.5)} \qquad x_i = 0.1i,\ i = 1, \ldots, 11 \tag{10}$$

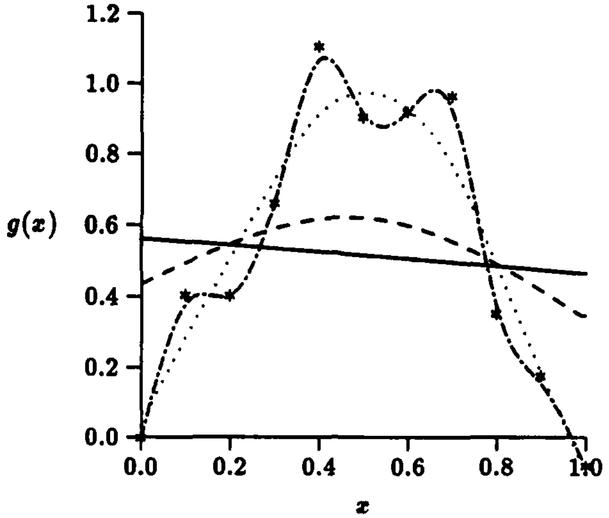

Figure 2: cubic spline approximations for $\alpha = 10.0$ (solid line), $10^{-1}$ (dashed line), $10^{-3}$ (dotted line) and $10^{-5}$ (dot-dash); $\sin(x)/x$ data with added noise of variance of 0.05.

and the noise samples $n_i$ are taken from a normal distribution of zero mean and variance 0.05. Additive noise of this type occurs in many applications in which data points $x_i$ are incorrectly labelled. Figure 2 again plots cubic spline approximations for different values of $\alpha$. We see that as $\alpha$ is decreased, the approximations progressively begin to model noise on the data.

EXAMPLE 3 In Figure 3 the data points are $(x_i + n_i, f_i)$ with $f_i$ given by Equation (10). In this example zero-mean Gaussian noise of variance 0.01 is added to the values $x_i$. Thus, the range of the function remains the same. This corresponds to situations in which the data is correctly labelled (*i.e.* correct values for $f_i$) but experimental procedures or otherwise have caused errors in the measurements $x_i$.

These examples, although very simple, illustrate some of the problems with curve fitting:

- A smoothness constraint may be inappropriate for some data sets. Heuristic techniques for allowing discontinuities have been considered by [2].
- It is important to understand the conditions under which the data set was gathered (and, as we shall see in the following section, the generalisation behaviour which we wish to obtain). These conditions will provide some clue as to the strategy for determining the regularisation parameter, $\alpha$, since noise may appear on the parameter values, $x_i$, or the function values $f_i$, or both.

Also, the choice of functional $\Omega$ is prompted by the nature of the problem. For example, several different regularisation principles for low-level vision are given in [23]. Many of the

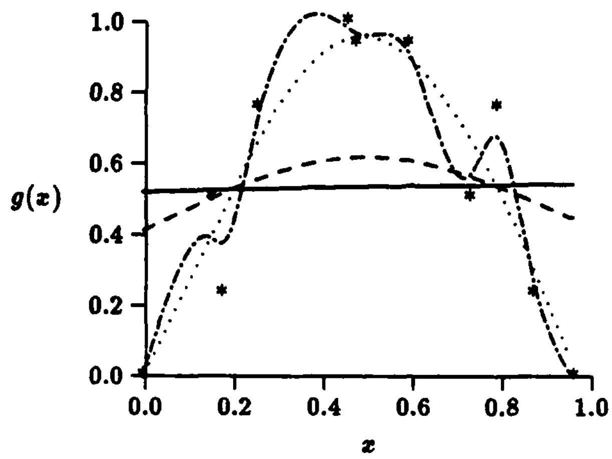

Figure 3: cubic spline approximations for $\alpha = 10.0$ (solid line), $10^{-1}$ (dashed line), $10^{-3}$ (dotted line) and $10^{-5}$ (dot-dash); $\sin(x)/x$ data with noise of variance 0.01 added to data values, $x_i$.

stabilising functionals are of the Tikhonov type. Tikhonov stabilisers are integrals of linear combinations of the squares of the derivatives of the desired solution, $z$, and are given by ([29], page 70)

$$\Omega(z) = \int_a^b \sum_{r=0}^{p} q_r(x) \left(\frac{d^r z}{dx^r}\right)^2 dx, \tag{11}$$

where $q_r \geq 0, r = 0, 1, \ldots, p-1$ and $q_p(x) > 0$. They have been widely used in inverse problems.

One of the main problems in standard regularisation theory is in the choice of the smoothing functional, $\Omega$, and the degree of smoothness required for the function to be recovered. The value of $\alpha$ controls the compromise between the degree of regularisation of the solution and its closeness to the data and standard regularisation theory provides techniques for determining the best $\alpha$. However, standard regularisation methods impose the constraints on the problem by a variational principle, such as the cost functional of Equation (2). The cost that is minimised should reflect physical constraints about what represents a good solution : it has to be both close to the data and regular by making the quantity $\Omega(z)$ small but this is often chosen in an *ad hoc* manner.

## 2.2 Specifying the Functional Form

The second approach to curve fitting which we shall consider is that of specifying the functional form of the fitting surface. Thus, the form of the interpolating (or approximating) surface is imposed (using some from of prior knowledge, physical intuition, guessing) rather than being determined as a consequence of an applied constraint. The functions may be defined globally (as in the case of some feed-forward networks) or locally (so that the parameters are valid only over a small region of the input space). The parameters of the function are chosen to minimise an error criterion, for example, the sum squared error between the function values, $f_i$ and the fitting surface values at the data points, $x_i$. Of course, if the function depends nonlinearly on the parameters, there is no guarantee of obtaining a global minimum of the fitting error (if indeed there is a unique global minimum). Also, if there are too many parameters then the solution may be very sensitive to noise on the data. The following examples give some one–dimensional illustrations using polynomials as the fitting functions.

EXAMPLE 4 In this example the fitting surface is a polynomial of order $d$ of the form

$$g(x) = \sum_{j=0}^{d} a_0 x^j \tag{12}$$

and depends linearly on the unknown parameters, $a_j$. Given a set of data points, $\{(x_i, f_i), i = 1, \ldots, P\}$, the parameters $a_j$ are chosen to minimise the error

$$\sum_{i=1}^{P} (f_i - g(x_i))^2 \tag{13}$$

and the solution can be obtained using a pseudo–inverse technique. Figure 4 shows polynomial fits to the step function used in Example 1 for different values of the degree, $d$. As the order of the polynomial is increased, the fitting error decreases, but the 'generalisation' becomes poorer. For example, there are greater oscillations for degree 8 than the cubic spline and the fit at the data points is poorer. The main reason for this is that in this example we are attempting a *global* fit to the data whereas the cubic spline approximation comprises functions which are defined *locally*.

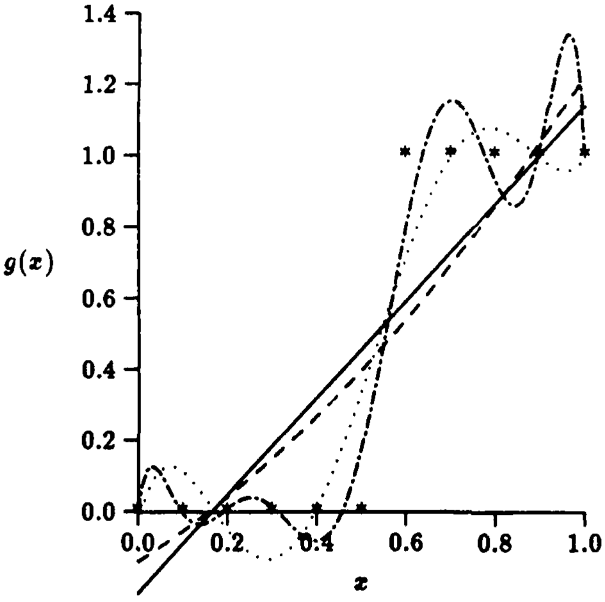

Figure 4: Polynomial approximations for $d = 1$ (solid line), 2 (dashed line), 5 (dotted) and 8 (dot–dash).

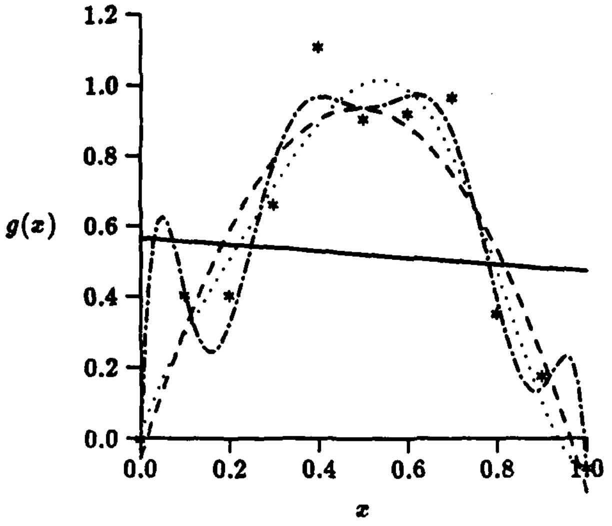

Figure 5: Polynomial approximations for $d = 1$ (solid line), 2 (dashed line), 5 (dotted) and 8 (dot–dash); $\sin(x)/x$ curve with added noise of variance = 0.05.

EXAMPLE 5 Figure 5 plots polynomial fits to the noisy data of Example 2 (a $\sin(x)/x$ curve with noise added to the function values, $f_i$)

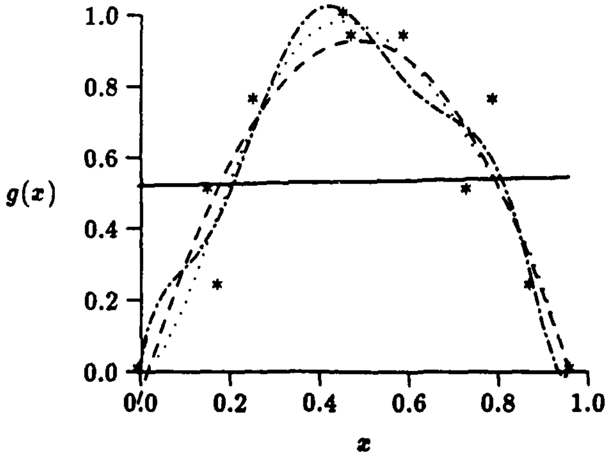

Figure 6: Polynomial approximations for $d = 1$ (solid line), 2 (dashed line), 5 (dotted line) and 8 (dot–dash); $\sin(x)/x$ curve with noise of variance = 0.01 added to the data values.

EXAMPLE 6 Figure 6 plots polynomial fits to the noisy data of Example 3 (a $\sin(x)/x$ curve with noise added to the parameter values $x$).

We see from the above examples that choosing a model with too few parameters may give a poor fit to the data, whilst if there are too many parameters then noise on the data is modelled. However, the sensitivity of the fitting surface to noise, for large parameter models, may be reduced if a regularisation term is introduced. That is, we minimise

$$\sum_{i=1}^{P}(f_i - g(x_i))^2 + \alpha \int_{x_1}^{x_P} \left(\frac{d^2g}{dx^2}\right)^2 dx \tag{14}$$

Thus, we are specifying the functional form *and* using a regularisation term. In this situation, we cannot approach the problem as one in variational calculus in the manner of Example 1. That is, we are not allowed to minimise the stabilising functional subject to a prescribed constraint on the fitting error since the parametric form for the fitting surface may not allow the constraint to be satisfied. Thus, we use a prescribed value for $\alpha$ and minimise the total term (the smoothing functional).

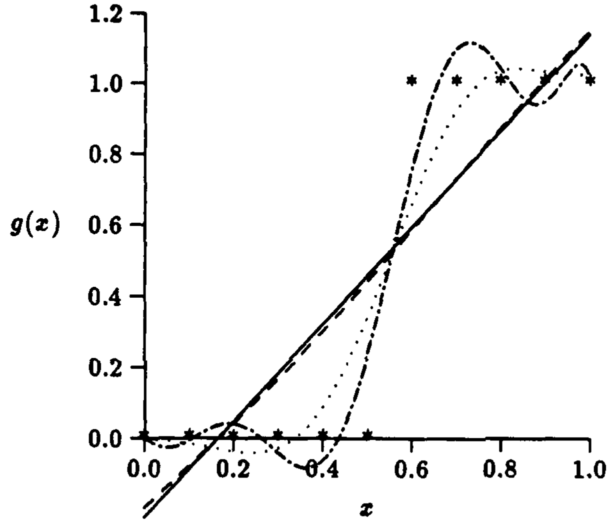

Figure 7: Polynomial approximations for $d = 8$ and $\alpha = 10.0$ (solid line), $10^{-1}$ (dashed line), $10^{-3}$ (dotted line) and $10^{-5}$ (dot–dash)

EXAMPLE 7 Figure 7 plots step function approximations for a polynomial of degree 8 and several values of $\alpha$. Comparing this figure with the $d = 8$ curve of figure 4, we see that the addition of the regularisation term has resulted in smoother approximations to the data. Thus the high oscillatory effects resulting from choosing a parametric form with too high an order or complexity (too many free parameters) can be offset by the use of a regularisation term. Therefore the choice of order is not so critical, though there still remains the problem of choosing an appropriate regularisation parameter.

The discussion in this section has related to the problem of curve fitting using prescribed functional forms. Specifically, it is not limited to feed–forward network structures, but the problems of surface–fitting and potential mechanisms of solution do apply to network architectures since, as discussed in Section 1, feed–forward networks may be viewed as performing a functional mapping from an input space to an output space. Therefore it is to be expected that for networks containing more adjustable parameters (in terms of the weights of the network) than are required to solve a given task, the solutions obtained by minimisation of an output error will not necessarily be stable against perturbations of the data and may give poor generalisation properties. This has been overcome by the introduction of an additional term in the error expression and is analogous to the regularisation term in (14). This has been termed deterministic weight decay [6] and effectively defines a new surface, $E' = E(w) + E_2(w)$, where $w$ is the weight vector, $E(w)$ is the error criterion at the output of the network (often a sum-square error measure) and $E_2(w)$ is the additional term which is large when $|w|$ is large. If $E_2$ is too large then the solutions do not correspond to solutions of the original task. If $E_2$ is extremely large, then there is only one solution, $w = o$. On the other hand, if $E_2$ is very small, it does not affect the solution very much at all. Sometimes a solution is to use a term $E_2(w, t)$ which is a decreasing function of time. This has been termed 'simulated ironing' [6] and is very similar to the 'graduated non-convexity' approach for function minimisation [2, 3]. Also, a specific form for $E_2$ (as one of Tikhonov's regularisation functions) in neural network applications have been considered by Farhat [7] for target shape estimation using radar imaging data. However, it should be emphasised that the choice of the form for $E_2$ should depend on knowledge of the data, the solution and the real world task for which the approximation is being constructed.

Characteristics of generalisation should be motivated by high-level concepts of what the surfaces created by the generalisers should be like when the generalisers are operating in the real world [36]. Physical plausibility of the solution, rather than its uniqueness, should be the most important concern in regularisation analysis and a physical analysis of the problem should play the main rôle. It is important to choose a penalty term or regularisation term, expressing an estimate of the *a priori* implausibility of each possible solution, which is appropriate to the task.

The approach considered in the following section is to derive a fitting surface which is robust to noise on the data points $x$. This is relevant to real-world problems where a generaliser is required to operate on noise-corrupted data.

## 3 Minimum Mean-square Approximations

In this section we consider a least squares approach to generalisation. We wish to determine some property, $\boldsymbol{f}$, of a data sample[1] $\boldsymbol{x}$. We assume that the training set is noiseless and we wish to generalise to data samples corrupted by noise (and consequently not in the training set). For example

- we may be given a portion of a sinusoid signal and be asked to determine the frequency.
- in image restoration problems with a measured or modelled point-spread function (including, for example, synthetic aperture radar with known imaging operator) the data, $\boldsymbol{x}$, may be the diffraction-limited image of an object of finite support and the problem is to reconstruct the object, $\boldsymbol{f}(\boldsymbol{x})$.

[1] We now use blodface variables to denote vector quantities

- in the problem of point–source location using either focal–plane or aperture plane arrays in which the array manifold is assumed known – either measured during a calibration procedure or modelled using idealised beamshapes – the data $\mathbf{x}$ is an array of receiver outputs and $f(\mathbf{x})$ is the position of a source in the scene.

We assume that, in the absence of noise, $f(\mathbf{x})$ is known. For example, in each of the illustrations above, we can determine the frequency of the sinusoid, recover the object unambiguously from its image and determine the position of the source exactly from the data sample $\mathbf{x}$. Thus, we know the mapping, $f(\mathbf{x})$ perhaps continuously as a function of $\mathbf{x}$, perhaps at discrete sample points $\mathbf{x}_i$. This is not such a severe restriction since in some situations we may generate $f(\mathbf{x})$ from a theoretical model. In other cases, $f(\mathbf{x})$ may be obtained by a calibration procedure in which data samples (the set $\{(\mathbf{x}_i, f_i), \; i = 1, P\}$) are collected in a controlled environment, perhaps at a very high signal–to–noise ratio so that the data is, in effect, noiseless.

The use of a noiseless data set to characterise a generaliser is the problem addressed by Wolpert [35, 36]. The *HERBIE* (*HEuRistic BInary Engine*) generalisation models of Wolpert are essentially local models which are constructed to ensure the correct response to the training data. Thus the models perform strict interpolation and are only valid for situations in which the test set is itself noiseless also. The range of application of such models is rather limited. In addition, the models assume that the data is single–valued, *i.e.* for a given data sample, $\mathbf{x}$, there is a unique observation, $f(\mathbf{x})$. For many pattern classification tasks this is an unrealistic assumption (for example, see the medical illustration in [17]) as distributions will invariably overlap. In the *NETtalk* example considered by Wolpert [36] to illustrate his theory, there is a unique class for a given data pattern and therefore it is possible to construct a surface which strictly interpolates the data points.

In practice, the data $z$ which is to be analysed will be corrupted by noise, so that $z = \mathbf{x} + \mathbf{n}$ where, in the three examples above, $\mathbf{n}$ would be a noisy one–dimensional signal, a two–dimensional image or a vector of noise samples. Given $z$, we require an estimate of the unknown physical quantity $f(\mathbf{x})$. One obvious candidate for the estimate is $f(z)$. However, this may not be suitable for two main reasons:

1. $z$ may not lie in the domain of $f$. In the image restoration example above, the noise may comprise out–of band components, so that $z$ does not lie in the domain of band–limited images. Thus, the function, $f$, is not defined for $z$.

2. Even if $f(z)$ is defined, it may not yield an 'optimal' estimate for $f(\mathbf{x})$, the uncorrupted value, possibly giving large errors for small perturbations to $\mathbf{x}$.

Therefore, we seek an approximation, $g$, to the function $f$ which is

1. defined for all perturbations, $\mathbf{n}$, to $\mathbf{x}$.

2. minimises the sensitivity of the estimate with respect to the noise perturbations.

3. $g(z) \rightarrow f(\mathbf{x})$ as $\mathbf{n} \rightarrow 0$.

It differs from the strategy of Wolpert in that we are not attempting to reproduce the training set exactly, but to minimise the expected square error between the function $f$ and its approximation $g$ when there is noise on the data points. In many practical situations, data vectors are likely to be corrupted by noise and it is important that an estimator properly takes account of the data on which it is operating. The degree of noise corruption, or the signal–to–noise ratio, will depend on many factors relating, for example, to measuring equipment, signal levels and the experimental procedure. Although the noise statistics and the signal–to–noise ratio for the operating environment may be known or measured, they may differ appreciably from those used to collect the training data for the estimator. For example, in a signal processing task, the network may be required to operate over a range of values of signal–to–noise ratio. In order to choose appropriate parameter values for the estimator, it would be necessary to collect different sets of training data representative of each signal–to–noise region and to train the estimator on each set of data. This would give values for the parameters of the estimator for the different operating régimes. In some practical situations, data may be expensive or time–consuming to collect and therefore it would be desirable to use one training set and to train the estimator to operate at different expected signal–to–noise ratios. In the following subsection we derive the analytic form for a minimum variance approximation to a given transformation and consider approximations to it when the transformation, $f(\boldsymbol{x})$, is defined by a finite set of points $\{(\boldsymbol{x}_i, f_i), i = 1, \ldots, P\}$. In subsection 3.2 we derive a high signal–to–noise ratio approximation for the variance and apply this to feed–forward networks in 3.3. Finally, we conclude this section with a summary of the main results.

### 3.1 Minimum Mean–square Estimate

Suppose that we wish to approximate a transformation $f$ from $\mathbb{R}^n$ to $\mathbb{R}^{n'}$. Let the approximation be given by $g$ which is chosen so that the quantity $V$, defined by

$$V = \int\int |f(\boldsymbol{x}) - g(\boldsymbol{x} + \boldsymbol{\xi})|^2 p_n(\boldsymbol{\xi}) p(\boldsymbol{x}) d\boldsymbol{x} d\boldsymbol{\xi} \tag{15}$$

is a minimum, where $p_n(\boldsymbol{\xi})$ is the probability density function of a noise distribution in the space $\mathbb{R}^n$ and $p(\boldsymbol{x})$ defines the distribution of data points $\boldsymbol{x}$ in the space $\mathbb{R}^n$. Equation (15) defines the expected square error in the approximation when the data points in the domain of $f$ are corrupted by additive noise, and may be written (for $z = \boldsymbol{x} + \boldsymbol{\xi}$) as

$$V = \int\int (f(\boldsymbol{x}) - g(z))^2 p_n(z - \boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} dz. \tag{16}$$

Minimising with respect to the function $g$ gives the solution for $g$ as

$$g(z) = \frac{\int f(\boldsymbol{x}) p_n(z - \boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}}{\int p_n(z - \boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}} \tag{17}$$

This is the approximation to the function $f$ for which the expected square error in the functional value, integrated over the domain of $f$, is a minimum and generalises $f$ to points $z$ outside the distribution of the data points $\boldsymbol{x}$. It is shown in Appendix A that a more general form for $g(z)$ is

$$g(z) = E\{f(\boldsymbol{x})|z\} \triangleq \int f(\boldsymbol{x}) p(\boldsymbol{x}|z) d\boldsymbol{x}, \tag{18}$$

the expected value of $f(x)$ given the data sample $z$, and for the case of additive noise, the conditional density $p(x|z)$ is given by

$$p(x|z) = \frac{p_n(z-x)p(x)}{\int p_n(z-x)p(x)dx}. \tag{19}$$

For a function $f$ defined by a finite set of points $\{(x_i, f_i), i = 1, \ldots, P\}$ in $\mathbb{R}^n \otimes \mathbb{R}^{n'}$, then provided that the integrands in Equation (17) are sufficiently smooth, the solution $g$ may be approximated by $\hat{g}$ given by

$$\hat{g}(z) = \frac{\sum_{i=1}^{P} f_i \, p_n(z-x_i)}{\sum_{i=1}^{P} p_n(z-x_i)} \tag{20}$$

or

$$\hat{g}(z) = \sum_{i=1}^{P} f_i \, \tilde{p}_n(z-x_i) \tag{21}$$

where $\tilde{p}_n(z-x_i)$ is defined by

$$\tilde{p}_n(z-x_i) = \frac{p_n(z-x_i)}{\sum_{i=1}^{P} p_n(z-x_i)} \tag{22}$$

This equation is identical in form to radial basis function approximations [4] in that the approximating functional is a linear combination of (specified) nonlinear functions of the difference between a data point, $z$ and a 'centre'. In this case the nonlinear basis functions are determined by the noise probability density function, the centres by the data points $x_i$, and the weights are the function values, $f_i$ at the centres. Thus a radial basis function network structure arises as a natural consequence of the minimum variance solution. For example, for a Gaussian noise model with diagonal covariance matrix with equal diagonal elements $\sigma^2$,

$$\hat{g}(z) = \frac{\sum_{i=1}^{P} f_i \exp[-\frac{1}{2\sigma^2}|z-x_i|^2]}{\sum_{i=1}^{P} \exp[-\frac{1}{2\sigma^2}|z-x_i|^2]}. \tag{23}$$

Note that in order to derive the function $g$ which approximates $f$ and generalises to unseen data, we have not assumed a specific functional form, nor a smoothness condition. We have assumed that we know how to perform the mapping if there were no noise (noiseless training data) and assumed a minimum mean square error measure. A consequence of this is the radial basis function nature of the solution. However, we do need to know the noise distribution. If we were to assume that it is Gaussian with diagonal covariance matrix with equal elements, then we would need to specify the noise variance on the test data.

The function $\hat{g}$ will provide a good approximation to the exact minimum mean-square solution, $g$, if the standard deviation of the noise is large compared to the distance between sample points, $x_i$. Poggio [22] has also derived a radial basis function approximation to $f(x)$ as a consequence of specifying a regularisation term in which the operator $P$ has radial symmetry.

Note that the function $g(z)$ may be defined over the whole space $\mathbb{R}^n$, whereas the data points $x$ may lie on a reduced dimension manifold, $X$, in $\mathbb{R}^n$ (as specified by the probability

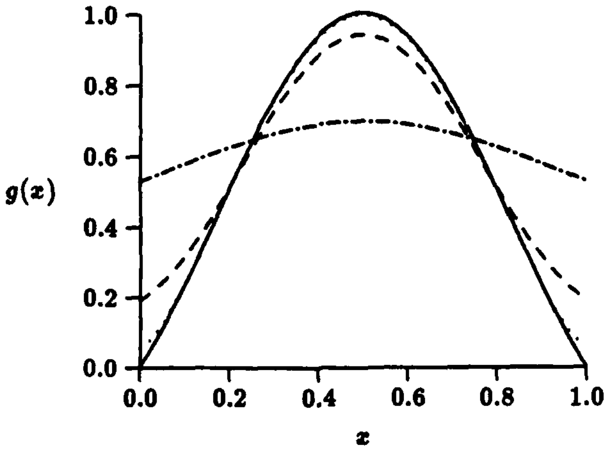

Figure 8: Approximations to $f(x)$ given by Equation (24) (solid line) for $\sigma^2 = 0.1$ (dot–dash line), 0.01 (dashed line) and 0.001 (dotted line)

density function, $p(\boldsymbol{x})$). Thus, the approximation to $\boldsymbol{f}$, $\boldsymbol{g}(\boldsymbol{z})$, is defined for values of $\boldsymbol{z}$ which do not necessarily lie on the manifold, $X$. This is important in many applications in which noise will corrupt data points, $\boldsymbol{x}$, to give values $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{\xi}$ which lie outside the domain of $\boldsymbol{f}$. In these situations it is not sufficient to interpolate the training set $\{(\boldsymbol{x}_i, \boldsymbol{f}_i), i = 1, \dots P\}$ without due regard to defining the mapping for points outside the manifold.

EXAMPLE 8 In this example the minimum mean–square approximations to the function

$$f(x) = \frac{\sin(x - 0.5)}{x - 0.5} \tag{24}$$

(solid line) are calculated for a Gaussian noise model and plotted in Figure 8 for values of the variance, $\sigma^2$, of $10^{-1}$ (dot–dash), $10^{-2}$ (dashed) and $10^{-3}$ (dotted). Equation (17) was evaluated using Simpson's rule assuming a uniform distribution for $x$ on $[0, 1]$.

The minimum mean–square approximation derived above provides a biased estimate, in that for a data point, $\boldsymbol{x}_0$, the mean of the estimate (the average over all perturbations $\boldsymbol{\xi}$ to $\boldsymbol{x}_0$) is not necessarily equal to the functional value $\boldsymbol{f}(\boldsymbol{x}_0)$, *i.e.*

$$\int \boldsymbol{g}(\boldsymbol{z}) p(\boldsymbol{z}|\boldsymbol{x}_0) d\boldsymbol{z} \neq \boldsymbol{f}(\boldsymbol{x}_0) \tag{25}$$

where $\boldsymbol{z} = \boldsymbol{x}_0 + \boldsymbol{\xi}$. In some practical situations it may be advantageous to have an *unbiased* estimate so that integration may be performed after the functional transformation, *i.e.* we need to produce an approximation $\boldsymbol{g}(\boldsymbol{z})$ which is defined for all noise perturbations and which, for inputs $\boldsymbol{z} = \boldsymbol{x}_0 + \boldsymbol{\xi}$, if averaged will tend to $\boldsymbol{f}(\boldsymbol{x}_0)$, the true value in the absence of noise.

This may be achieved by demanding a minimum variance estimate which satisfies the constraint

$$\int g(z)p(z|\boldsymbol{x}_0)dz \;=\; f(\boldsymbol{x}_0) \tag{26}$$

*i.e.* minimise

$$V \;=\; \int\!\!\int(f(\boldsymbol{x}) - g(\boldsymbol{x}+\boldsymbol{\xi}))^2 p_n(\boldsymbol{\xi})p(\boldsymbol{x})d\boldsymbol{x}d\boldsymbol{\xi} - \int\lambda(\boldsymbol{x})\int(f(\boldsymbol{x}) - g(\boldsymbol{x}+\boldsymbol{\xi}))p_n(\boldsymbol{\xi})p(\boldsymbol{x})d\boldsymbol{x}d\boldsymbol{\xi} \tag{27}$$

where $\lambda(z)$ is a Lagrange multiplier. The solution for $g(z)$ is

$$g(z) \;=\; \frac{\int(f(\boldsymbol{x}) - \frac{1}{2}\lambda(\boldsymbol{x}))p_n(z-\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}}{\int p_n(z-\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}} \tag{28}$$

and $\lambda(\boldsymbol{x})$ satisfies the integral equation

$$\frac{1}{2}\int\lambda(\boldsymbol{x})Q(\boldsymbol{x},\boldsymbol{x}_0)p(\boldsymbol{x})d\boldsymbol{x} \;=\; \int f(\boldsymbol{x})Q(\boldsymbol{x},\boldsymbol{x}_0)p(\boldsymbol{x})d\boldsymbol{x} - f(\boldsymbol{x}_0) \tag{29}$$

where $Q(\boldsymbol{x},\boldsymbol{x}_0)$ is defined by

$$Q(\boldsymbol{x},\boldsymbol{x}_0) \;=\; \int\left[\frac{p_n(\boldsymbol{\xi})p_n(\boldsymbol{\xi}+\boldsymbol{x}_0-\boldsymbol{x})}{\int p_n(\boldsymbol{\xi}+\boldsymbol{x}_0-\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x})}\right]d\boldsymbol{\xi} \tag{30}$$

For a finite number of data points, the integral in Equation (28) is again of a radial basis function form, with this time the weights being the function values offset by the Lagrange multiplier terms.

## 3.2 Perturbation Analysis for High Signal–to–Noise Ratios

In the previous subsection a solution for the minimum variance approximation to a known function, $f : \mathbb{R}^n \rightarrow \mathbb{R}^{n'}$ was derived. When the functional transformation is specified only by points in $\mathbb{R}^n \otimes \mathbb{R}^{n'}$, then this minimum variance solution may be approximated by a summation which takes the form of a radial basis function network with nonlinear functions being (normalised) noise probability density functions. This summation will be a good approximation to the minimum variance solution provided that the standard deviation of the noise distribution is large compared to the spacing between samples, $\boldsymbol{x}_i$. In a low noise situation (where the standard deviation of the noise distribution is small compared to the distance between sample points), the approximation $\hat{g}(z)$ to $g(z)$ will be accurate only in the region of the sample points and at intermediate values will give a very poor approximation. Therefore, we need to specify a model for the approximation to $f(\boldsymbol{x})$ or a constraint in the form of a regularisation term in order to describe how the function varies between sample points.

Let us assume that we have a parameterised model for the approximation to $f$. In the following section, we shall consider a specific model (namely a feed–forward network), but at the moment there is no restriction to its form other than it is a continuous function, $g$, of the data $z$ with continuous first derivatives. First of all we shall calculate the perturbation to the error between the actual values, $\boldsymbol{f}_i$ and the approximate values due to noise on the data points.

Let $\{(\boldsymbol{x}_i, \boldsymbol{f}_i,\ i = 1, \ldots, P\}$ denote the set of points describing the mapping $\boldsymbol{f} : \mathbb{R}^n \rightarrow \mathbb{R}^{n'}$. For a given data value, $\boldsymbol{x}_p$, let $E_p \equiv E(\boldsymbol{x}_p)$ be the error between the approximation to $\boldsymbol{f}(\boldsymbol{x})$ and the desired value, $\boldsymbol{f}_p$ for the $p$th pattern, $\boldsymbol{x}_p$. Often, the total error, is given by

$$E_T = \frac{1}{P}\sum_{p=1}^{P} E_p = \frac{1}{P}\sum_{p=1}^{P} E(\boldsymbol{x}_p), \tag{31}$$

the sum–square error between the approximations and the desired values (termed the 'target' values in a feed–forward network framework), though in the analysis which follows we impose no such restriction. Let the pattern $\boldsymbol{x}_p$ be corrupted by additive noise, $\boldsymbol{n}$, so that the error for pattern $\boldsymbol{x}_p$ is

$$E_p = E(\boldsymbol{x}_p + \boldsymbol{n}) = E(\boldsymbol{x}_p) + (\boldsymbol{n}^*\nabla)E|_{\boldsymbol{x}_p} + (\boldsymbol{n}^*\nabla)^2 E\Big|_{\boldsymbol{x}_p} \tag{32}$$

expanding by Taylor's theorem and assuming that $\boldsymbol{n}$ is small so that terms $\mathcal{O}(|\boldsymbol{n}|^3)$ may be neglected. For $\langle \boldsymbol{n} \rangle = 0$, the expected error (average over all noise vectors) is

$$\langle E_p \rangle = E(\boldsymbol{x}_p) + \frac{1}{2}\langle \boldsymbol{n}^* \boldsymbol{H}^p \boldsymbol{n} \rangle \tag{33}$$

where $E(\boldsymbol{x}_p)$ is the error in the absence of noise and $\frac{1}{2}\langle \boldsymbol{n}^* \boldsymbol{H}^p \boldsymbol{n} \rangle$ is an additional error term where $\boldsymbol{H}^p$ is the Hessian with respect to the data space components, evaluated for the $p$th pattern

$$H_{ij}^p = \frac{\partial^2 E}{\partial x_i \partial x_j}\bigg|_{\boldsymbol{x}_p} . \tag{34}$$

For $\langle n_i n_j \rangle = \sigma^2 \delta_{ij}$, the additional error term may be written

$$\frac{1}{2}\langle \boldsymbol{n}^* \boldsymbol{H}^p \boldsymbol{n} \rangle = \frac{\sigma^2}{2} Tr(\boldsymbol{H}^p), \tag{35}$$

where $Tr$ is the matrix trace operation. Averaging over all data patterns gives the mean expected error

$$\langle E_T \rangle = \frac{1}{P}\sum_{p=1}^{P} E(\boldsymbol{x}_p) + \frac{\sigma^2}{2P}\sum_{p=1}^{P} Tr(\boldsymbol{H}^p). \tag{36}$$

Equation (36) gives the mean expected error in the approximation and consists of two terms. The first is the error in the approximation when there is no noise on the data. The second term is a second derivative quantity proportional to the noise variance $\sigma^2$. For $\sigma^2 = 0$, $\langle E_T \rangle$ reduces to the usual error term in the absence of noise. Thus, if we have a mapping $\boldsymbol{f} : \mathbb{R}^n \rightarrow \mathbb{R}^{n'}$ defined by points in $\mathbb{R}^n \otimes \mathbb{R}^{n'}$ in which the data points in $\mathbb{R}^n$ are corrupted by additive noise with zero mean and variance $\sigma^2$ (sufficiently small so that the higher order terms in the Taylor expansion may be neglected), then minimising the error over all patterns and over the noise distribution with respect to the parameters of the approximating function, $\boldsymbol{g}$, is equivalent to minimising a modified error term defined on the

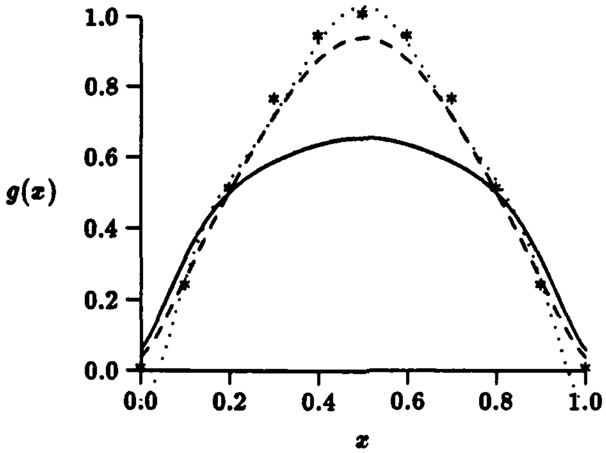

Figure 9: Polynomial approximations for $d = 8$ and $\sigma^2 = 0.05$ (solid line), 0.01 (dashed line) and 0.001 (dotted line)

patterns in the absence of noise. A different approximation to $f$ can be derived for different values of the noise variance, $\sigma^2$. Equation (36) shows that the effects of noise on the test data can be compensated for by training an approximator with a modified error criterion.

EXAMPLE 9 In this example we are again using a one–dimensional illustration and fitting the sampled $\sin(x)/x$ data with a polynomial of degree 8 (see Figure 9). The regularisation term is the second derivatives of the error and the coefficients of the polynomial are determined by minimising the total error as given by Equation (36).

The two terms in Equation (36) may be regarded as the usual error metric plus a regularisation or stabilising term with regularisation parameter $\sigma^2$, the variance of the noise on the inputs. The expression for the error given in Equation (36) is a function of the parameters of the approximation whose precise form depends on the particular error criterion and the model. For the usual sum–square error criterion, the quantity, $\langle E_T \rangle$ is the square error between the observation values and the approximator, averaged over the noise distribution and the data distribution and is equivalent to $V$ given by (15) for the general case of arbitrary noise distribution. Thus, minimising the modified error gives an approximation to the vector of posterior probabilities.

For the sum–squared error criterion, the second term in Equation (36) may be written as

$$\frac{\sigma^2}{P} \sum_{p=1}^{P} \left( \|J^p\|^2 - (f_p - g(x_p))^* q^p \right) \tag{37}$$

where the $n' \times n$ matrix $J^p$ is the Jacobian

$$J^p_{ij} = \left. \frac{\partial g_i}{\partial x_j} \right|_{x_p} \tag{38}$$

representing the derivative of the $i$th component of the approximation with respect to the $j$th input, evaluated for pattern $\boldsymbol{x}_p$. The quantity $\sigma^2||J^p||^2$ is the trace of the covariance matrix of the approximator, *i.e.* the sum of the variances (about the mean output values). The vector $q^p = (q_1^p, q_2^p, \ldots, q_n^p)^*$ is a vector of second derivative terms, with $k$th component

$$q_k^p = \sum_{i=1}^{n} \left.\frac{\partial^2 g_k}{\partial x_i^2}\right|_{\boldsymbol{x}_p} \tag{39}$$

evaluated for the $p$th pattern.

### 3.3 Feed–forward Network Approximations

We now consider a particular parametric form for the generaliser when the noise perturbations are small, namely a feed–forward layered network architecture. The behaviour of continuous feed–forward networks in noisy environments has also been studied in the situation when there is additive noise on the inputs and/or the outputs of the network [12]. The approach adopted in this paper differs from [12] in that we assume a set of input–output pairs which characterise the mapping exactly, but train the network for operation in a noisy environment.

The basic network structure we shall consider is the feed–forward type with an input layer of $n$ nodes, an output layer of $n'$ nodes and an intermediate hidden layer of $n_0$ nodes. The input to each node in the hidden layer is a combination of the input vector and a weight vector. The structure of the standard layered network model considered in this paper is depicted in Figure 10.

For a multilayer perceptron with a single hidden layer and the sum–square error criterion, the regularisation term may be written in terms of the weights as

$$\frac{\sigma^2}{P}\sum_{p=1}^{P}\left[||\Lambda \mathbf{G}^p M||^2 - \sum_{j=1}^{n_0}\left(\sum_{i=1}^{n'}(t_i^p - o_i^p)\lambda_{ij}h_j^p(1-h_j^p)(1-2h_j^p)\right)\sum_{k=1}^{n}\mu_{jk}^2\right] \tag{40}$$

where $h_j^p$ is the output of the $j$th hidden node for input pattern $\boldsymbol{x}_p$ and $G^p$ is an $n_0 \times n_0$ diagonal matrix with $(i,i)$ component $h_i^p(1-h_i^p)$. The scalar quantities $\lambda_{ij}$ and $\mu_{jk}$ are the weights between the $i$th output node and the $j$th hidden node, and between the $j$th hidden node and the $k$th input node respectively. This extra error term is a function of the final layer weights $\Lambda$ (the $n' \times n_0$ matrix of weights $\lambda_{ij}$ excluding the biases) and the first layer weights $M$ (with components $\mu_{ij}$) (note that $G^p$, $h_i^p$ and $o_i^p$ also are functions of these weights).

The regularisation term involves second derivatives of the outputs as a function of the inputs and is given above for the multilayer perceptron and the sum–square error criterion. In order to minimise the expected total error using any of the schemes described in the previous section, an evaluation of the derivatives of the above error term is required. This leads to third derivative terms (derivatives of the Hessian are required). Although in principle it is possible to do this at each iteration of the optimisation procedure, it would lead to considerably more computation being required to obtain a minimum of the error. However, we may use the knowledge that the noise perturbation to the input patterns is

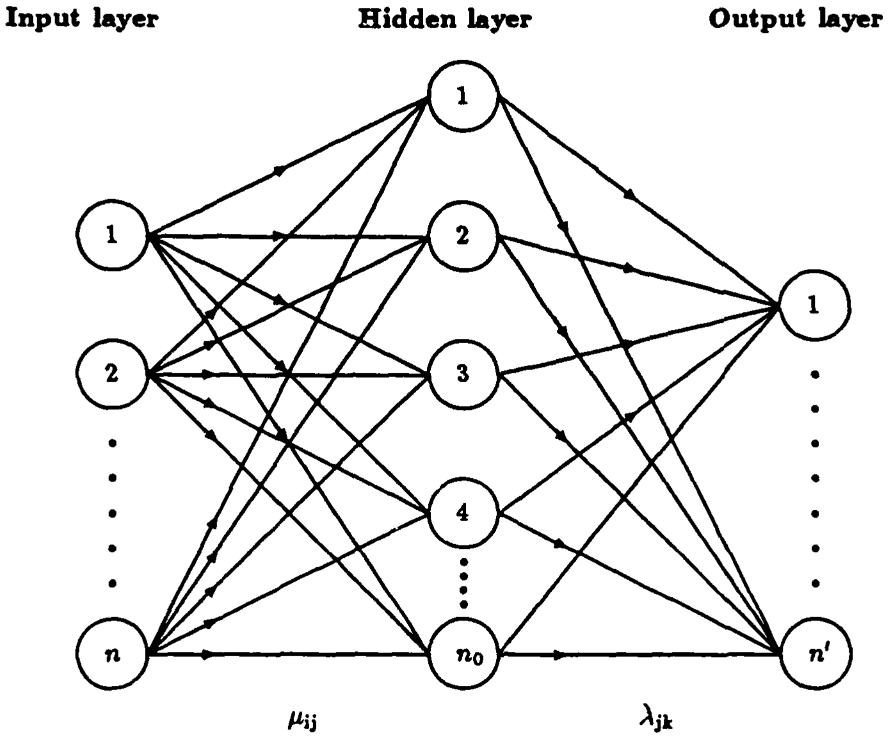

Figure 10: A schematic diagram of the standard feed forward adaptive layered network geometry considered in this paper.

small (so that the Taylor series expansion is valid) to obtain the solution for the parameters of the model which minimise the total error in terms of the solution which minimises the error when the regularisation term is absent. This is given in Appendix B.

## 3.4 Summary of Main Results

This section has described a minimum mean square error approach to function approximation and generalisation. The main results can be summarised as follows.

1. The minimum mean square estimate is the expected vector of the *a posteriori* density.
2. Approximating the integral over the training set by a summation gives a radial basis function solution with nonlinear functions being determined by the noise probability density function. This approximation is valid provided that the standard deviation of the noise is large compared to the sample spacing of the data points. This radial basis function form is a consequence of the sum–square error condition and is not imposed *a priori*.
3. If the expected noise on the test data is small compared to the sample spacing then the radial basis function approximation is invalid. A generaliser can still be constructed by

specifying the functional form and modifying the sum-squared error on the training set by an additional error criterion or regularisation term. The regularisation parameter is determined by the noise variance and the form of the regularisation term by the functional form assumed for the approximation.

# 4 A Radar Point Source Location Problem

In this section, we illustrate the results of the previous sections by considering an application of feed-forward adaptive networks to point source location using focal-plane arrays. The method may be applied to any array of sensors where the image response function may be characterised by an array manifold. However, in order to be specific, we have confined our study to the focal-plane situation and one idealised array in particular, namely a 5 × 1 array of elements, each with a $\sin(x)/x$ shape point-spread function (Equation (41) below). Thus, the array manifold consists of a set of real-valued vectors. A theoretical and practical study of a maximum likelihood approach to point-source location using millimetre-wave staring array sensors is given in [31] and a detailed study of the use of feed-forward networks is presented in [30].

For a linear array (with a narrow slit as the aperture) the imaging equation is one-dimensional and the response $h_i(\xi)$ is giver by

$$h_i(\xi) = \frac{sin[\Omega(x_i - \xi)]}{\pi(x_i - \xi)} \qquad (41)$$

Figure 11 illustrates the response of each receiving element to a point source in the far field for the linear array. The distance between the peak of a response and the first null is termed the "beamwidth" and is equal to unity for these examples ($\Omega = 1$). The distance between adjacent receivers in the focal-plane is taken to be unity, giving samples of the image at Nyquist rate.

The problem is to determine the position of a source in the scene given the outputs, possibly corrupted by noise, of the array of receivers. A training set was generated which consisted of a set of normalised images of single point sources together with the corresponding source positions. For the linear array, the images of a single source are calculated using Equation (41) at 101 different positions, equally spaced across the field of view of the array from −2.5 to 2.5 (at a spacing of 1/20). The image vectors are normalised to remove the effects of source amplitude and these form the training set. For the test data, the normalised images of a single source at 200 positions chosen randomly between −2.5 and 2.5 are taken as input with the source position as target.

The training data lies on a one-dimensional manifold (the position on the manifold is characterised by source position) within the 5-dimensional space of array outputs. However, in noisy operating conditions, data vectors will not necessarily lie on the manifold. There we seek a function of the data which approximates the position and generalises to points off the manifold. In the previous section it was shown that provided that the noise variance is sufficiently large, a generaliser which minimises the mean square error is of the form of a radial basis function network with centres at the data points and nonlinear functions

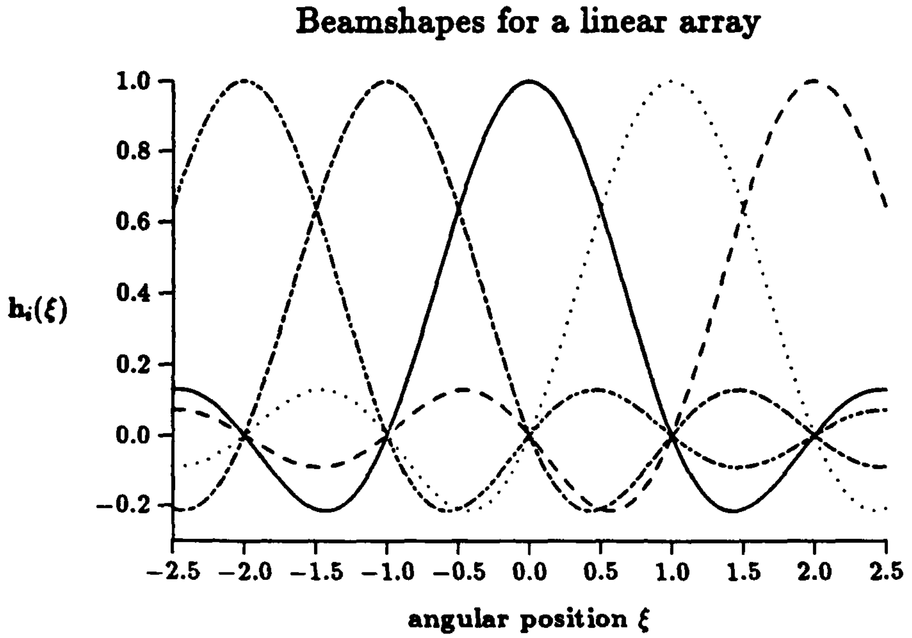

Figure 11: Response of each receiving element to a point source in the far-field for a 5 × 1 linear array of receivers in the focal-plane of an imaging system.

determined by the assumed noise probability density function.[2] If the expected noise is small, then some functional form must be assumed for the approximator. In the example below, we consider the case of small noise perturbations and consider the use of a feed-forward network as the approximator.

The networks, which we have chosen to apply to this problem, have a single hidden layer with input in the form of a hyperplane and a nonlinear transfer function

$$\phi(x) = \frac{1}{1+e^{-x}}. \tag{42}$$

This is a rather arbitrary choice, and there may be other nonlinearities more suitable to this problem. The output nodes are linear functions of the input, *i.e.* $\Phi(x) = x$.

The focal-plane array illustration described is highly idealised. In general, the array manifold, and the image vectors, would be complex vectors and some method of incorporating complex vectors into a feed-forward network would have to be considered. This is not a difficult task, but for our purposes we shall restrict the example to considering real vector inputs only.

For a given value of $n_0$, the number of hidden units, and a given set of training data, a multilayer perceptron network was trained to find the weights which minimised the error at the output. Initially, the values of the weights were chosen randomly from a uniform distribution on $(-1.0, 1.0)$. Then the BFGS (Broyden-Fletcher-Goldfarb-Shanno) nonlin-

[2] Since the data is normalised to remove the effects of source amplitude, the form of the radial basis function network differs in detail from that given in Equation (23). See [30] for further details

ear optimisation strategy[3] was used to find the solution for the weights for which the error at the output of the network is a minimum. The network was tested using the test data generated and the normalised error on test calculated. For the linear array, the experiment was run for 100 different random start configurations for the weights. The solution for the weights which gave the lowest normalised error on test over the 100 experiments was chosen as the one which best describes the mapping from image space to position space for the particular network under consideration.

Figure 12 plots the bias in the position estimate (true position minus predicted position, evaluated in the absence of noise on the input) for a network with 5 hidden units trained to minimise the sum-squared error on the test set. The figure shows that in the absence of noise, a network can predict the position very accurately. Generally, the error is less than $5.0 \times 10^{-4}$ of a beamwidth across the field of view. Figure 13 plots the root of the mean square error at the output as a function of position for noise of variance $\sigma^2 = 10^{-3}$ on the inputs to the network. The data for this figure was generated as a result of a Monte-Carlo simulation in which noise was added to an image vector and the position of the source estimated using the network. The errors were averaged over 10000 sample images for each position. The figures show that the noise error dominates the bias error at the outputs.

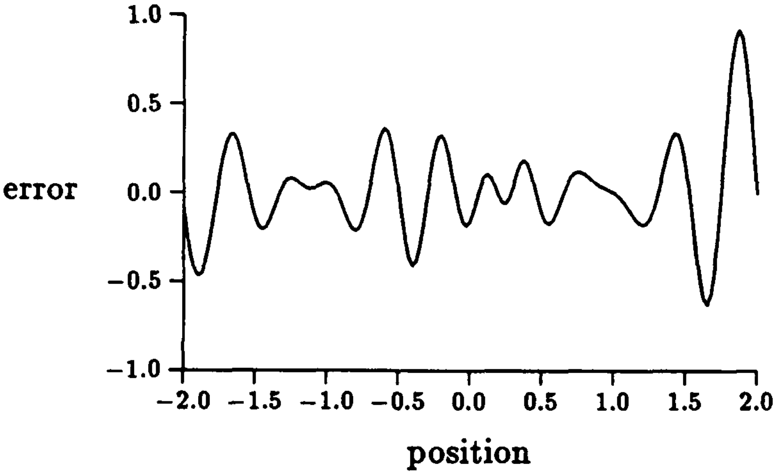

Figure 12: Bias (×1000) as a function of position for a linear array and a network with 5 hidden units, trained on 101 patterns.

Figure 14 plots the bias error in the absence of noise on the inputs as a function of position for a network trained to minimise the modified error criterion (the sum-squared error over the training set plus a regularisation term[4] with regularisation parameter with a value of $10^{-3}$). The bias error is considerably poorer, increased by a factor of about 100. However, the root of the mean square error at the outputs for noise on the inputs is reduced as shown in Figure 15 (generated in a similar way to Figure 13, using a Monte-Carlo simulation)

[3] See [33] for a comparison of different nonlinear optimisation strategies on the point-source location problem

[4] In fact, the additional term comprised the square of the norm of the Jacobian matrix only. The second derivative term was neglected

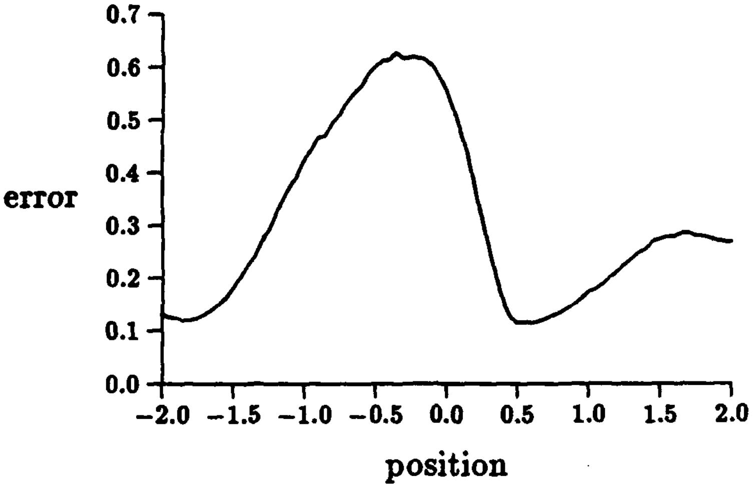

Figure 13: The root of the total square error as a function of position for the linear array and a network with 5 hidden units

## 5 Discussion.

The intention of this paper has been to consider the problem of approximating a set of data points in $\mathbb{R}^n \otimes \mathbb{R}^{n'}$ by a functional mapping $\mathbb{R}^n \rightarrow \mathbb{R}^{n'}$. There are several ways in which this task may be approached and the particular strategy adopted should depend on the practical problem being addressed, the form of the training data and the expected test data or operating conditions. In some cases, strict interpolation of the training data may be appropriate, but if there is noise on this data or there is noise on the expected operational data, then this is not the correct strategy. Further, in a classification task, strict interpolation may not be possible due to overlapping classes (a given data sample in $\mathbb{R}^n$ may have different observations in $\mathbb{R}^{n'}$).

The class of problems considered in this paper is one in which the training data is noiseless (perhaps generated from a model or obtained during some calibration procedure) but there is noise on the expected test data. Thus, there is one set of training data and different operating (test) conditions are modelled by assuming a form for the noise probability density function. A minimum mean square error approach has been employed and this leads to an approximation which generalises from the training set to noisy conditions and which gives the vector of posterior probabilities. For a finite number of training samples, this generaliser may be approximated by a radial basis function network, which is a good approximation to the generaliser provided that the noise is large compared to the spacing of samples in the data space, $\mathbb{R}^n$.

If the expected perturbations to the training data are small, then a functional form may be assumed for the generaliser. It has been shown that, for small noise perturbations, minimising a modified error criterion on the training set (a sum-squared error plus a regularisation term dependent on the functional form and a regularisation parameter dependent on the noise variance) gives an approximation to the posterior probabilities.

A problem in point-source location using a focal plane array has been used to illustrate the analysis. It was shown that training a multilayer perceptron using the modified

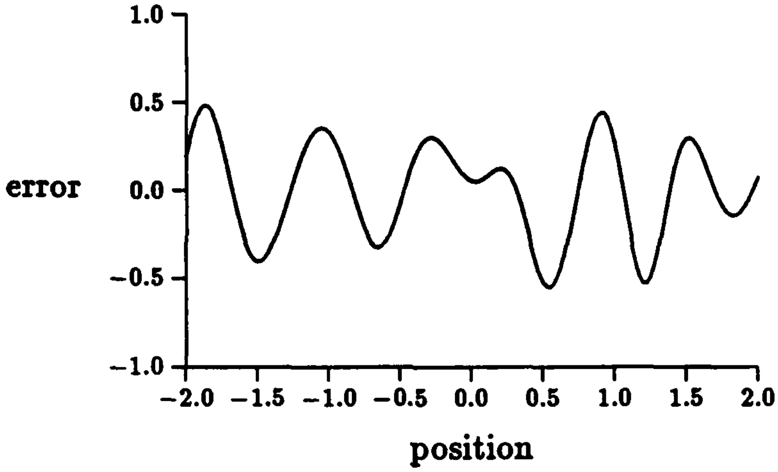

Figure 14: Bias (×10) as a function of position for a linear array and a network with 5 hidden units, trained on 101 patterns, and with a value of $\sigma^2$ of $10^{-3}$.

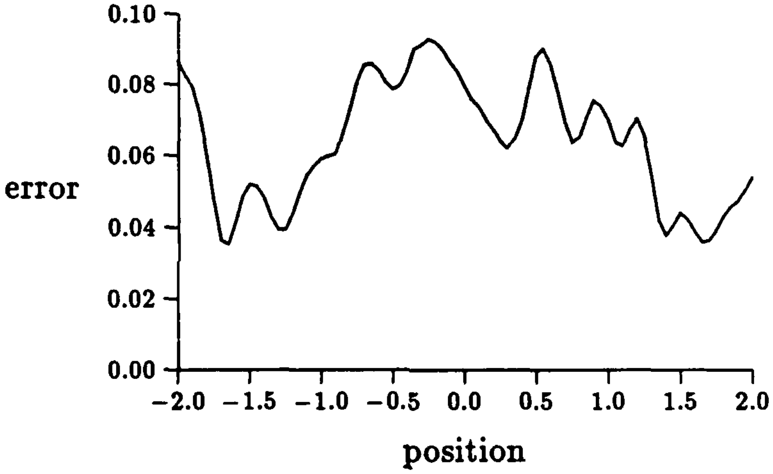

Figure 15: The root of the total square error as a function of position for the linear array and a network with 5 hidden units

error criterion gave better performance on noisy data (that is, smaller error in the position estimate) than the usual sum–squared error.

A practical difficulty with the approach is that the probability density function of the noise must be known to construct the generaliser. For many problems this will not be known *a priori*. Even if a particular form could be assumed (for example, Gaussian additive noise), then it is unlikely that the overall signal–to–noise ratio would be known. Therefore it would be necessary to have several generalisers appropriate for different operating conditions or a generaliser which adapts to different noise levels.

Also, for some problems, the training data will inevitably be noisy. If this data is representative of the operating conditions then it is not necessary to model the noise. However, if we wish to generalise to different operating conditions, then a model must be assumed for the noise process. It is also important to know where the noise occurs in the training set. In a classification problem it may occur on the class labels (incorrectly classified data) and the method of training should take this into account. These are avenues for possible further work.

In conclusion, this paper has shown how a generaliser can be constructed. A feed–forward network approximates the posterior probabilities provided that the network is trained to minimise a sum–squared error augmented with a regularisation term. An illustration of a radar point–source location problem has been given and is considered in detail elsewhere [30].

## Acknowledgements

I would like to thank John Bridle, David Lowe and Stephen Luttrell for their comments and advice during the course of this work.

# Appendix A Minimum Mean–square Error Solution

Let $\boldsymbol{f}(\mathbf{x})$ be a transformation from $\mathbb{R}^n$ to $\mathbb{R}^{n'}$ and $p(\mathbf{x})$ denote the distribution of data points, $\mathbf{x}$, in $\mathbb{R}^n$. Let $\boldsymbol{g}(\boldsymbol{z})$ be the approximation to $\boldsymbol{f}(\mathbf{x})$ and the conditional distribution of $\boldsymbol{z}$ on $\mathbf{x}$ be $p(\boldsymbol{z}|\mathbf{x})$. The square error, $V$, between $\boldsymbol{f}$ and $\boldsymbol{g}$ integrated over all $\boldsymbol{z}$ and $\boldsymbol{x}$ is given by

$$\begin{aligned} V &= \int\int |\boldsymbol{f}(\mathbf{x}) - \boldsymbol{g}(\boldsymbol{z})|^2 p(\boldsymbol{z},\mathbf{x}) d\mathbf{x} d\boldsymbol{z} \\ &= \int\int |\boldsymbol{f}(\mathbf{x}) - \boldsymbol{g}(\boldsymbol{z})|^2 p(\boldsymbol{z}|\mathbf{x}) p(\mathbf{x}) d\mathbf{x} d\boldsymbol{z} \end{aligned} \tag{43}$$

Minimising with respect to $\boldsymbol{g}$ gives

$$\begin{aligned} \boldsymbol{g}(\boldsymbol{z}) &= \frac{\int \boldsymbol{f}(\mathbf{x}) p(\boldsymbol{z}|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}}{\int p(\boldsymbol{z}|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}} \\ &= \int \boldsymbol{f}(\mathbf{x}) p(\mathbf{x}|\boldsymbol{z}) d\mathbf{x} \end{aligned} \tag{44}$$

the expected value of $\boldsymbol{f}(\mathbf{x})$ given $\boldsymbol{z}$, Thus the minimum mean–square estimate is the expected vector of the *a posteriori* density ([9], Chapter 5).

For the special case where $\boldsymbol{z}$ represents additive Gaussian noise so that (for Gaussian noise of variance $\sigma^2$)

$$p(\boldsymbol{z}|\mathbf{x}) = p_n(\boldsymbol{z} - \mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp(-\frac{1}{2\sigma^2}|\boldsymbol{z} - \mathbf{x}|^2) \tag{45}$$

then, for a finite number of training samples, $\boldsymbol{g}(\boldsymbol{z})$ may be approximated by

$$\hat{\boldsymbol{g}}(\boldsymbol{z}) = \frac{\sum_{i=1}^{P} \boldsymbol{f}_i \exp[-\frac{|\boldsymbol{z}-\mathbf{x}_i|^2}{2\sigma^2}]}{\sum_{i=1}^{P} \exp[-\frac{|\boldsymbol{z}-\mathbf{x}_i|^2}{2\sigma^2}]}. \tag{46}$$

Alternatively, in a pattern classification problem in which there are $M$ classes, if $\mathbf{x}_i$ represents a class label so that $p(\boldsymbol{z}|\mathbf{x}_i)$ is the class conditional distribution, then

$$\boldsymbol{g}(\boldsymbol{z}) = \sum \boldsymbol{f}(\mathbf{x}_i) p(\mathbf{x}_i|\boldsymbol{z}) \tag{47}$$

and for $\boldsymbol{f}(\mathbf{x}_i)$ being a vector with a '1' in the $i$th position and '0' elsewhere, then $\boldsymbol{g}(\boldsymbol{z})$ is the vector of posterior probabilities.

## Appendix B Perturbation Analysis Solution for the Weights

**Let $\boldsymbol{w}_0$ be the solution for the weights (the matrices $\mathbf{A}$ and $M$ and the vectors of biases $\mu_0$ and $\boldsymbol{\lambda}_0$) which minimises the sum–squared error when noise is absent. Let the solution for a noise level of $\sigma^2$ be $\boldsymbol{w}_0 + \boldsymbol{w}'$. The derivative of the total error at $\boldsymbol{w}_0 + \boldsymbol{w}'$ is equal to zero**

$$\sum_{p=1}^{P} \frac{\partial E(\boldsymbol{x}_p)}{\partial \boldsymbol{w}}\bigg|_{\boldsymbol{w}_0+\boldsymbol{w}'} + \frac{\sigma^2}{2}\sum_{p=1}^{P}\sum_{k=1}^{n} \frac{\partial}{\partial \boldsymbol{w}}\frac{\partial^2 E}{\partial x_k^2}\bigg|_{\boldsymbol{w}_0+\boldsymbol{w}',\boldsymbol{x}_p} = 0 \qquad (48)$$

**and expanding about the 'noiseless' solution $\boldsymbol{w}_0$ to first order gives**

$$\sum_{p=1}^{P} \frac{\partial E(\boldsymbol{x}_p)}{\partial \boldsymbol{w}}\bigg|_{\boldsymbol{w}_0} + \left(\boldsymbol{w}'^{*}\frac{\partial}{\partial \boldsymbol{w}}\right)\sum_{p=1}^{P} \frac{\partial E(\boldsymbol{x}_p)}{\partial \boldsymbol{w}}\bigg|_{\boldsymbol{w}_0} + \frac{\sigma^2}{2}\sum_{p=1}^{P}\sum_{k=1}^{n} \frac{\partial}{\partial \boldsymbol{w}}\frac{\partial^2 E}{\partial x_k^2}\bigg|_{\boldsymbol{w}_0,\boldsymbol{x}_p} = 0 \qquad (49)$$

where $\partial/\partial\boldsymbol{w} = (\partial/\partial w_1, \ldots, \partial/\partial w_N)^*$, and $N$ is the total number of weights. Now, since the error in the absence of noise is a minimum at $\boldsymbol{w}_0$, *i.e.*

$$\sum_{p=1}^{P} \frac{\partial E(\boldsymbol{x}_p)}{\partial \boldsymbol{w}}\bigg|_{\boldsymbol{w}_0} = 0, \qquad (50)$$

then Equation (49) may be written

$$\sum_{p=1}^{P}\sum_{i=1}^{N} w_i' \frac{\partial}{\partial w_i}\frac{\partial E(\boldsymbol{x}_p)}{\partial w_j}\bigg|_{\boldsymbol{w}_0} = -\frac{\sigma^2}{2}\sum_{p=1}^{P}\sum_{k=1}^{n} \frac{\partial^3 E}{\partial w_j \partial x_k^2}\bigg|_{\boldsymbol{w}_0,\boldsymbol{x}_p} \qquad (51)$$

or

$$\boldsymbol{H}_w \boldsymbol{w}' = -\frac{\sigma^2}{2}\frac{\partial}{\partial \boldsymbol{w}}\{Tr(\boldsymbol{H}_x)\} \qquad (52)$$

where the $N \times N$ matrix $\boldsymbol{H}_w$ and the $n \times n$ matrix $\boldsymbol{H}_x$ are defined by

$$\begin{aligned} [\boldsymbol{H}_w]_{ij} &= \sum_{p=1}^{P} \frac{\partial^2 E(\boldsymbol{x}_p}{)}\partial w_i \partial w_j \\ [\boldsymbol{H}_x]_{ij} &= \sum_{p=1}^{P} \frac{\partial^2 E}{\partial x_i \partial x_j}\bigg|_{\boldsymbol{x}_p} \end{aligned} \qquad (53)$$

**Provided that the inverse of the matrix $\boldsymbol{H}_w$ exists, then the solution for the perturbation to $\boldsymbol{w}_0$ is**

$$\boldsymbol{w}' = -\frac{\sigma^2}{2}[\boldsymbol{H}_w]^{-1}\frac{\partial}{\partial \boldsymbol{w}}\{Tr(\boldsymbol{H}_x)\} \qquad (54)$$

## References

[1] C.M. Bishop. Curvature-driven Smoothing in Backpropagation Neural Networks. In *International Neural Network Conference*, volume 2, pages 749–752, Paris, 1990. Kluwer Academic Publishers.

[2] A. Blake and A. Zisserman. Using Weak Continuity Constraints. Internal Report CSR-186-85, University of Edinburgh, Department of Computer Science, James Clerk Maxwell Building, The King's Buildings, Mayfield Road, Edinburgh, EH9 3JZ, 1985.

[3] A. Blake and A. Zisserman. Invariant Surface Reconstruction Using Weak Continuity Constraints. In *IEEE Computer Vision and Pattern Recognition Conference*, pages 62–67. IEEE, 1986.

[4] D.S. Broomhead and D. Lowe. Multi-variable Functional Interpolation and Adaptive Networks. *Complex Systems*, 2(3):269–303, 1988.

[5] M. Casdagli. Nonlinear Prediction of Chaotic Time Series. *Phisica D*, 35:335–356, 1989.

[6] J. Denker, D. Schwartz, B. Wittner, S. Solla, R. Howard, and L. Jackel. Large Automatic Learning, Rule Extraction, and Generalization. *Complex Systems*, 1:877–922, 1987.

[7] N.H. Farhat and B. Bai. Echo Inversion and Target Shape Estimation by Neuromorphic Processing. *Neural Networks*, 2(2):117–125, 1989.

[8] J.D. Farmer and J.J. Sidorowich. Prediction Chaotic Time Series. *Physical Review Letters*, 59(8):845–848, 1987.

[9] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Inc., London, 1972.

[10] K. Fukunaga and R.R. Hayes. Effects of Sample Size in Classifier Design. *IEEE Trans. Pattern Analysis Machine Intelligence*, 11(8):873–885, August 1989.

[11] G.H. Golub and C.F. Van Loan. An Analysis of the Total Least Squares Problem. *SIAM J Numer. Anal.*, 17(6):883–893, December 1980.

[12] T.J. Guillerm and N.E. Cotter. Neural Networks in Noisy Environment: A Simple Temporal Higher Order Learning for Feed-forward Networks. In *IEEE Int. Joint Conf. Neural Networks*, volume III, pages 105–112, San Diego, 1990.

[13] K. Hornik, M. Stinchcombe, and H. White. Multilayer Feedforward Networks are Universal Approximators. *Neural Networks*, 2(5):359–366, 1989.

[14] R.D. Jones, Y.C. Lee, C.W. Barnes, G.W. Flake, K. Lee, P.S. Lewis, and S. Qian. Function Approximation and Time Series Prediction with Neural Networks. In *IEEE Int. Joint Conf. Neural Networks*, volume I, pages 649–665, San Diego, 1990.

[15] G.S. Kimeldorf and G. Wahba. A Correspondence Between Bayesian Estimation on Stochastic Processes and Smoothing Splines. *The Annals of Mathematical Statistics*, 41(2):495–502, 1970.

[16] A. Lapedes and R. Farber. How Neural Nets Work. In D.Z. Anderson, editor, *Neural Information Processing Systems*, pages 442–456. AIP, Denver, 1987.

[17] D. Lowe and A.R. Webb. Incorporating Prior Probabilities and Misclassification Costs into Network Training : An Example from Medical Diagnosis. *Network*, 1(3):299–323, July 1990.

[18] D. Lowe and A.R. Webb. Optimised Feature Extraction and the Bayes Decision in Feed-forward Classifier Networks. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1990. in press.

[19] D. Lowe and A.R. Webb. Time Series Prediction by Adaptive Networks: A Dynamical Systems Perspective. *IEE Proceedings Part F*, 138(1):17–24, February 1991.

[20] K. Matsuoka. An Approach to Generalization Problem in Back-propagation Learning. In *International Neural Network Conference*, volume 2, pages 765–768, Paris, 1990. Kluwer Academic Publishers.

[21] V.A. Morozov. *Methods for Solving Incorrectly Posed Problems*. Springer Verlag, New York, 1984.

[22] T. Poggio and F. Girosi. Regularization Algorithms for Learning That Are Equivalent to Multilayer Networks. *Science*, 247:978–982, February, 23 1990.

[23] T. Poggio, V. Torre, and C. Koch. Computational Vision and Regularization Theory. *Nature*, 317:314–319, September 1985.

[24] S. Raudys and V. Pikelis. On Dimensionality, Sample Size, Classification Error, and Complexity of Classification Algorithm in Pattern Recognition. *IEEE Trans. Pattern Analysis Machine Intelligence*, 2(3):242–253, May 1980.

[25] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learing Internal Representations by Error Propagation. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing : Exploration in the Microstructure of Cognition*, pages 318–362. Cambridge : MIT Press, 1986.

[26] L.L. Schumaker. *Spline Functions: Basic Theory*. John Wiley and Sons, Inc., 1981.

[27] C.H. Sequin and R.D. Clay. Fault Tolerance in Artificial Neural Networks. In *IEEE Int. Joint Conf. Neural Networks*, volume I, pages 703–708, San Diego, 1990.

[28] M. Stinchcombe and H. White. Approximating and Learning Unknown Mappings Using Multilayer Feedforward Networks with Bounded Weights. In *IEEE Int. Joint Conf. Neural Networks*, volume III, pages 7–16, San Diego, 1990.

[29] A.N. Tikhonov and V.Y. Arsenin. *Solutions of Ill-posed Problems*. V.H. Winston and Sons, Washington D.C., 1977.

[30] A.R. Webb. Application of Feed-forward Networks to Point-source Location Using a Millimetre-wave Focal-plane Array Radar. RSRE Memo, R.S.R.E., St Andrews Road, Malvern, Worcs., WR14 3PS, 1990. to be submitted to *IEEE Aerospace and Electronic Systems*.

[31] A.R. Webb. Point-source Location Using a Millimetre-wave Focal-plane Array Radar. *IEE Proceedings Part F*, 1990. to appear.

[32] A.R. Webb and D. Lowe. The Optimised Internal Representation of Mulitlayer Classifier Networks Performs Nonlinear Discriminant Analysis. *Neural Networks*, 3(4):367–375, July/August 1990.

[33] A.R. Webb, D. Lowe, and M.D. Bedworth. A Comparison of Nonlinear Optimisation Strategies for Feed–forward Adaptive Layered Networks. RSRE Memo 4157, R.S.R.E., St Andrews Road, Malvern, Worcs., WR14 3PS, 1988.

[34] A. Wieland and R. Leighton. Geometric Analysis of Neural Network Capabilities. In *IEEE First Int. Conf. Neural Networks*, volume III, pages 385–392, San Diego, 1987.

[35] D.H. Wolpert. A Benchmark for How Well Neural Nets Generalize. *Biol. Cybern.*, 61:303–313, 1989.

[36] D.H. Wolpert. Constructing a Generalizer Superior to NETtalk via a Mathematical Theory of Generalization. *Neural Networks*, 3(4):445–452, 1990.

INTENTIONALLY BLANK

# REPORT DOCUMENTATION PAGE

DRIC Reference Number (if known) ....................................

Overall security classification of sheet ................................................UNCLASSIFIED...............................................................................

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the field concerned must be marked to indicate the classification eg (R), (C) or (S).

| | | |
|---|---|---|
| Originators Reference/Report No.<br>MEMO 4453 | Month<br>JANUARY | Year<br>1991 |

Originators Name and Location

RSRE, St Andrews Road
Malvern, Worcs WR14 3PS

Monitoring Agency Name and Location

Title

FUNCTIONAL APPROXIMATION BY FEED-FORWARD NETWORKS:
A LEAST-SQUARES APPROACH TO GENERALISATION

| | |
|---|---|
| Report Security Classification<br>UNCLASSIFIED | Title Classification (U, R, C or S)<br>U |

Foreign Language Title (in the case of translations)

Conference Details

| | |
|---|---|
| Agency Reference | Contract Number and Period |
| Project Number | Other References |

| | |
|---|---|
| Authors<br>WEBB, A R | Pagination and Ref<br>33 |

Abstract

This paper considers a least-squares approach to function approximation and generalisation. The particular problem addressed is one in which the training data is noiseless (perhaps specified by an assumed model or obtained during some calibration procedure) and the requirement is to define a mapping which approximates the data and which generalises to situations in which data samples are corrupted by noise. The least-squares approach produces a generaliser which is the vector of posterior probabilities and has the form of a Radial Basis Function network for a finite number of training samples. The finite sample approximation is valid provided that the noise on the expected operating conditions is large compared to the sample spacing in the data space. In the other extreme of small noise perturbations, it is shown that better generalisation will occur if the training error criterion (the sum-square error on the training set) is modified by the addition of a specific regularisation term. This is illustrated by an approximator which has a feed-forward architecture and applied to the problem of point-source location using the outputs of an array of receivers in the focal-plane of a lens.

Abstract Classification (U,R,C or S)
U

Descriptors

Distribution Statement (Enter any limitations on the distribution of the document)

UNLIMITED

580/48

INTENTIONALLY BLANK