??, ??, 1?? (??)
© ?? Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

Bayesian Approach to Neural Network Modelling with Input Uncertainty

W. A. WRIGHT
Andy.Wright@src.bae.co.uk
Sowerby Research Centre
FPC 267, PO Box 5
British Aerospace, Bristol, BS12 7QW
England

G. RAMAGE, D. CORNFORD & I.T. NABNEY
Neural Computing Research Group
Aston University
Aston Triangle
Birmingham, B4 7ET
England

Received ??; Revised ??

Editors: ??

Abstract. It is generally assumed when using Bayesian inference methods for neural networks that the input data contains no noise. For real-world (errors in variable) problems this is clearly an unsafe assumption. This paper presents a Bayesian neural network framework which accounts for input noise provided that a model of the noise process exists. In the limit where the noise process is small and symmetric it is shown, using the Laplace approximation, that this method adds an extra term to the usual Bayesian error bar which depends on the variance of the input noise process. Further, by treating the true (noiseless) input as a hidden variable, and sampling this jointly with the network's weights, using a Markov chain Monte Carlo method, it is demonstrated that it is possible to infer the regression over the *noiseless* input. This leads to the possibility of training an accurate model of a system using less accurate, or more uncertain, data.

.1. Introduction

It can generally be assumed that any data produced in the real world will have some degree of uncertainty (i.e both systematic and random error). It is, therefore, a necessary requirement for any learning system to be able to cope with such uncertainty. A number of techniques have been put forward to deal with this. For instance it is possible to place error bars on the output of certain (e.g the RBF and MLP) neural networks [4]. Such methods allow the predicted distribution of the output, given the model parameters and input data, to be estimated. In the majority of cases these estimates *only* take into account the uncertainty in the target data (i.e. target noise) and in the model parameters. Usually no allowance is made for what is termed "errors in variables" or uncertainty in the input data. However, there is no reason, a priori, why the target variables are more likely to be noisy than the input variables.

A number of researchers have considered the errors in variable problem for neural networks. Tresp et al [15], addressing missing input data, show that for an input with additive Gaussian noise the expectation of the network's output will be biased and the error bar increased. Townsend and Tarassenko [14] have produced a similar result, for an RBF network, by taking a perturbative approach. Their method also allows for the error induced in the weights. They show that, for additive Gaussian noise, the output error bar acquires an extra term which is proportional to the covariance of the input noise process.

Here a Bayesian approach [5] to the calculation of the predictive distribution for a regression network has been taken. It is shown that, given a model of the input noise process, it is possible to obtain an estimate of this posterior distribution on the output which allows for the uncertainty due to the noisy input. In the limit where the input noise is additive Gaussian and small, the result presented agrees with that of Townsend and Tarassenko. Furthermore, it is shown that by sampling, using a Markov chain Monte Carlo (MCMC) method [10] over the (latent) noiseless input variable, it is possible, given an appropriate prior over the noiseless data, to infer the regression given the *noiseless* input. This is demonstrated on a wind vector to radar backscatter forward model for the ERS-1 satellite that is trained using input vectors generated from numerical weather prediction models which are inherently noisy.

.2. Bayesian Regression

This section briefly describes the Bayesian inference methods introduced in [5] together with the calculation of error bars about the expected outcome using the Laplace approximation. Readers already familiar with these techniques may wish to omit this section.

Consider a regression problem with a set of inputs $\boldsymbol{x}^n = \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ (where $\boldsymbol{x}$ is a vector) and a corresponding set of targets $t^n = t_1, \ldots, t_N$. Then $D = \{t^n, \boldsymbol{x}^n\}$ forms a data set from which inference about the relationship between t and $\boldsymbol{x}$ can be made. If the targets are related to the inputs through some deterministic function $f(\boldsymbol{x})$ with additive noise

$$t = f(\boldsymbol{x}) + \boldsymbol{\epsilon},$$

where $\boldsymbol{\epsilon}$ is a Gaussian ($\mathcal{N}(0, \sigma_t^2)$), the probability density of t^* given some new input $\boldsymbol{x}^*$ is:

$$p(t^*|\boldsymbol{x}^*) \propto \exp\left[-\frac{1}{2\sigma_t^2}(f(\boldsymbol{x}^*) - t^*)^2\right]. \tag{1}$$

Given that the regression can be undertaken by a model (e.g. a neural network) with an output, $y(\boldsymbol{x}^*; \boldsymbol{w})$, which depends on the new input and a set of model weights $\boldsymbol{w}$, the conditional distribution (equation 1) can be written as the integral over these parameters:

$$p(t^*|\boldsymbol{x}^*, D) = \int p(t^*|\boldsymbol{x}^*, \boldsymbol{w})p(\boldsymbol{w}|D)\ d\boldsymbol{w}. \tag{2}$$

From equation 1 it then follows that:

$$p(t^*|\boldsymbol{x}^*, \boldsymbol{w}) = \left(\frac{\beta}{2\pi}\right)^{1/2} \exp\left(-\frac{\beta}{2}\left\{y(\boldsymbol{x}^*; \boldsymbol{w}) - t^*\right\}^2\right), \tag{3}$$

where $1/\beta = \sigma_t^2$.

Using Bayes' rule, $p(\boldsymbol{w}|D)$ can be written as,

$$\begin{aligned} p(\boldsymbol{w}|D) &= \tfrac{1}{p(D)}\, p(D|\boldsymbol{w})\, p(\boldsymbol{w}) \\ &= \tfrac{1}{Z_S} \exp\left(-\tfrac{\beta}{2}E_D - \tfrac{\alpha}{2}E_W\right) \qquad (4) \\ &= \tfrac{1}{Z_S} \exp\left(-S(\boldsymbol{w})\right), \end{aligned}$$

where Z_S is a normalising constant given by the integral,

$$Z_S = \int \exp\left(-S(\boldsymbol{w})\right)\; d\boldsymbol{w}.$$

The term E_D is the contribution from likelihood $p(D|\boldsymbol{w})$ which assuming that the data is independent can be written as the product

$$\begin{aligned} p(D|\boldsymbol{w}) &= \textstyle\prod_{i=1}^{N} p(t_i|\boldsymbol{x}_i, w) \\ &= \tfrac{1}{Z_D(\beta)} \exp\left(-\tfrac{\beta}{2}\textstyle\sum_{i=1}^{N}\{y(\boldsymbol{x}_i;\boldsymbol{w}) - t_i\}^2\right) \qquad (5) \\ &= \tfrac{1}{Z_D(\beta)} \exp\left(-\tfrac{\beta}{2}E_D\right), \end{aligned}$$

where Z_D is the normalising constant given by the integral of E_D over $\boldsymbol{w}$, and

$$Z_D(\beta) = \left(\frac{2\pi}{\beta}\right)^{N/2}.$$

The second term E_W is the contribution from the prior over the weights

$$p(\boldsymbol{w}) = \frac{1}{Z_W(\alpha)} \exp(-\frac{\alpha}{2}E_W), \qquad (6)$$

where again Z_W is a normalising constant,

$$Z_W(\alpha) = \int \exp(-\frac{\alpha}{2}E_W)\; d\boldsymbol{w}.$$

This prior can be interpreted as giving the regularising term to the integral. Depending on the network model (i.e. RBF or MLP) there are several forms of prior that can be used. For the purposes of the first part of this paper a simple quadratic regularisation function is adopted without loss of generality,

$$E_W = ||\boldsymbol{w}||^2.$$

.2.1. Laplace approximation

Using the Laplace approximation, the posterior distribution $p(\boldsymbol{w}|D)$ is approximated by as a Gaussian fitted to a mode. This is undertaken by maximising the posterior over the weights (which is equivalent to minimising $S(\boldsymbol{w})$ in equation 4) to determine the *most probable* weight vector ($\boldsymbol{w}_{\mathrm{MP}}$) and then Taylor expanding $S(\boldsymbol{w})$ about $\boldsymbol{w}_{\mathrm{MP}}$. Neglecting terms higher than second order leads to the approximation

$$S(\boldsymbol{w}) \simeq S_{\mathrm{MP}} + \Delta\boldsymbol{w}^T \boldsymbol{A}\ \Delta\boldsymbol{w}, \tag{7}$$

where $\Delta\boldsymbol{w} = \boldsymbol{w} - \boldsymbol{w}_{\mathrm{MP}}$ and $S(\boldsymbol{w}_{\mathrm{MP}})$ has been written as S_{MP} and $\boldsymbol{A}$ is the Hessian

$$\boldsymbol{A} = \nabla_w \nabla_w S_{\mathrm{MP}}. \tag{8}$$

Here it is assumed that there is only a single maximum to the distribution. Although this may not always be a valid assumption, for the purpose of this description it is assumed that it is sufficient.

Substituting equations 3 and 7 into equation 2 leads to the relationship,

$$p(t^*|\boldsymbol{x}^*, D) \propto \int \exp\left(-\frac{\beta}{2}\left\{y(x;w) - t\right\}^2\right) \exp\left(-\frac{1}{2}\Delta\boldsymbol{w}^T \boldsymbol{A}\ \Delta\boldsymbol{w}\right) d\boldsymbol{w}. \tag{9}$$

Assuming that the posterior distribution is sufficiently narrow the function $y(\boldsymbol{x}^*;\boldsymbol{w})$ may now be linearly approximated by Taylor expanding about $\boldsymbol{w}_{\mathrm{MP}}$. That is,

$$y(\boldsymbol{x}^*;\boldsymbol{w}) \simeq y(\boldsymbol{x}^*;\boldsymbol{w}_{\mathrm{MP}}) + \boldsymbol{g}^T \Delta\boldsymbol{w} \tag{10}$$

where

$$\boldsymbol{g} = \nabla_w y(\boldsymbol{x}^*;\boldsymbol{w})|_{\boldsymbol{w}=\boldsymbol{w}_{\mathrm{MP}}}.$$

Substituting into equation 9 and evaluating the integral over $\boldsymbol{w}$ gives,

$$p(t^*|\boldsymbol{x}^*, D) = \frac{1}{(2\pi\sigma_t^2)^{1/2}} \exp\left(-\frac{\left\{t^* - y(\boldsymbol{x}^*;\boldsymbol{w}_{\mathrm{MP}})\right\}^2}{2\sigma_t^2}\right), \tag{11}$$

where

$$\sigma_t^2 = \frac{1}{\beta} + \boldsymbol{g}^T \boldsymbol{A}^{-1} \boldsymbol{g}. \tag{12}$$

This provides a Gaussian approximation to the predictive distribution. From the standard deviation of this distribution it is possible to obtain an unbiased estimate of the uncertainty (or "error bar") about the predicted mean y_{MP}. This error bar has two components. The first $(1/\beta)$ is the variance in the distribution over the target vectors t. Although here this term is constant it is possible to make it input dependent and allow for what is termed *heteroscedastic* noise [2]. The second term is an estimate of the width of the posterior over the weights and reflects the uncertainty induced in the weights given the finite amount of data available to train the network.

Unfortunately, for many real problems such an estimate of uncertainty is incomplete. For real problems it is likely that the input will also be uncertain. It would be desirable if this uncertainty could be allowed for in the predictive distribution as is already the case for the uncertainty in the target data. To do this it is necessary to take a look at how noise would enter the inferencing system.

.3. Bayesian Regression with input noise

Suppose that the random vector $\boldsymbol{x}$, the *true* input, is hidden, and so cannot be observed, but samples from another random vector $\boldsymbol{z}$ can be written as a probabilistic function of x,

$$\boldsymbol{z} = p_x(\boldsymbol{x}, \boldsymbol{\gamma}), \tag{13}$$

where $\boldsymbol{\gamma}$ is a random noise vector, independent of $\boldsymbol{x}$, with distribution $p_\gamma(\boldsymbol{\gamma})$. Since only $\boldsymbol{z}$ can be observed data D' consists of targets and noisy inputs,

$$D' = \{t^n, \boldsymbol{z}^n\},$$

where $\boldsymbol{z}^n = \boldsymbol{z}_1, \ldots, \boldsymbol{z}_N$. Tresp et al [15] show that the expected target given the noisy input is

$$E[t^*|\boldsymbol{z}^*] = \int g(\boldsymbol{x}^*) p(\boldsymbol{z}^*|\boldsymbol{x}^*) p(\boldsymbol{x}^*)\ d\boldsymbol{x}^*,$$

where $g(\boldsymbol{x}^*) = E[t^*|\boldsymbol{x}^*]$ is the expectation of t^* given the *noiseless* variable and $p(\boldsymbol{z}^*|\boldsymbol{x}^*)$ is the *generative* distribution of the input noise process. In general, therefore, the expectation given the noisy input is a smoothed, or biased, version of the true expectation.

To illustrate this, consider the regression over data derived from a sine wave $f(x) = \sin(2\pi x)$. If the input noise is additive Gaussian $\mathcal{N}(0, \sigma_x^2)$ and $p(\boldsymbol{x})$ is slowly varying and so is approximated by a uniform distribution then:

$$\begin{aligned} E[t^*|\boldsymbol{z}^*] &= f(z), \\ &= \int_{-\infty}^{\infty} \sin(2\pi \boldsymbol{x}^*) \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp\left[\frac{-1}{2\sigma_x^2}(\boldsymbol{x}^* - \boldsymbol{z}^*)^2\right], \qquad (14) \\ &= \exp\left[-2\pi^2\sigma_x^2\right] \sin(2\pi \boldsymbol{z}^*). \end{aligned}$$

See figure 2. Thus $E[t^*|\boldsymbol{z}^*]$ is also a sine wave with the same frequency, but with an amplitude modulated by the exponent of the variance, and is therefore not equal to $E[t^*|\boldsymbol{x}^*]$.

For a non-linear network where the input is noisy it is not, therefore, sufficient to simply train the network on the input data. To obtain the correct unbiased output it is necessary to allow explicitly for the input noise process.

For this analysis it is assumed that, although the vector $\boldsymbol{x}$ is unknown, a model for the noise process *is* available. This is a significant assumption. However, it is conceivable that although data which relates the true input $\boldsymbol{x}$ to the targets t may not be available, "off line" calibration data or models which relate $\boldsymbol{z}$ to $\boldsymbol{x}$ may be.

If it is assumed that there is some model of the input noise, the inference system can be represented as two separate components.

- A generative component from which the noisy inputs are produced given unknown noiseless inputs (e.g. the calibration model)
- The regression model which takes the noiseless inputs and generates an appropriate output.

Represented graphically the *noiseless* input may be seen as a latent or hidden variable. Figure 1 is a graphical representation of the two cases, with the input noise model excluded and included. Here the shaded nodes represent observed data and the inference over the training data ($\{t^n, \boldsymbol{z}^n\}$) and new data ($\{t^*, \boldsymbol{z}^*\}$) have been separated.

The predictive distribution $p(t^*|\boldsymbol{z}^*, D')$ for the noisy input regression can, therefore, be written in terms of the marginal distribution

$$p(t^*|\boldsymbol{z}^*, D') = \int p(t^*|\boldsymbol{x}^*, D')\ p(\boldsymbol{x}^*|\boldsymbol{z}^*)\ d\boldsymbol{x}^*, \tag{15}$$

where $\boldsymbol{x}^*$ is the *new* (noiseless or latent) input.

The first term of the right hand side of equation 15 is the posterior over the model. As in equation 2 this can be expanded in terms of the model variables $\boldsymbol{w}$. Allowing for the independence of $\boldsymbol{w}$ on $\boldsymbol{x}^*$ this gives,

$$p(t^*|\boldsymbol{x}^*, D') = \int p(t^*|\boldsymbol{w}, \boldsymbol{x}^*)p(\boldsymbol{w}|D')\ d\boldsymbol{w}. \tag{16}$$

Now consider the $p(\boldsymbol{w}|D')$ term. Expanding $D' = \{t^n, \boldsymbol{z}^n\}$ and marginalising over $\boldsymbol{x}^n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, the latent (noiseless) input data points, gives

$$p(\boldsymbol{w}|D') = \int p(\boldsymbol{w}, \boldsymbol{x}^n|t^n, \boldsymbol{z}^n)\ d\boldsymbol{x}^n. \tag{17}$$

Using Bayes' rule and the conditional independence of t^n on $\boldsymbol{z}^n$ given $\boldsymbol{x}^n$ equation 17 can be written as,

$$p(\boldsymbol{w}|D') = \frac{p(\boldsymbol{w})}{p(t^n|\boldsymbol{z}^n)} \int p(t^n|\boldsymbol{w}, \boldsymbol{x}^n)p(\boldsymbol{x}^n|\boldsymbol{z}^n)\ d\boldsymbol{x}^n. \tag{18}$$

Finally using equations 15, 16 and 18 and Bayes' rule to express the conditional distribution over $\boldsymbol{x}$ and $\boldsymbol{z}$ in terms of the generative noise distribution gives

$$\begin{aligned} p(t^*|\boldsymbol{z}^*, D') &= \tfrac{1}{Z_{tz}} \textstyle\int p(t^*|\boldsymbol{w}, \boldsymbol{x}^*)p(\boldsymbol{z}^*|\boldsymbol{x}^*)p(\boldsymbol{x}^*) \\ &\qquad p(t^n|\boldsymbol{w}, \boldsymbol{x}^n)p(\boldsymbol{z}^n|\boldsymbol{x}^n)p(\boldsymbol{x}^n)p(\boldsymbol{w})\ d\boldsymbol{w}\ d\boldsymbol{x}^n d\boldsymbol{x}^*, \end{aligned} \tag{19}$$

where $Z_{tz} = p(\boldsymbol{z}^*)p(t^n, \boldsymbol{z}^n)$ is a normalising constant.

Thus the posterior of the output of the model conditioned on the noisy input $\boldsymbol{z}^*$ can be expressed as the integral over the model parameters $\boldsymbol{w}$, the perfect (but hidden) input $\boldsymbol{x}^*$ and perfect (but hidden) input data $\boldsymbol{x}^n$.

.3.1. *Interpretation of the Predictive Distribution*

Comparing the different components of equation 19 with equations 2, 3 and 4, it is possible to obtain an understanding of how these components would contribute to the posterior and so to the uncertainty of the expectation $E[t^*|\boldsymbol{z}^*, D']$. The integral over $\boldsymbol{w}$ is similar to that in equation 2. This integral may, therefore, be interpreted as providing contributions to the posterior which allow for the uncertainty in the target vectors and the density of the training data upon which the inference is based. The integration over $\boldsymbol{x}^n$ represents the contribution to the posterior from training the model on uncertain inputs while the integration over $\boldsymbol{x}^*$ gives a contribution to the predictive distribution which allows for the uncertainty in the new input.

.3.2. *Laplace approximation of the predictive distribution*

Generally the integral in equation 19 can only be effectively estimated by using MCMC methods. However, it is instructive to evaluate the error bars analytically using the Laplace approximation. To do this it is necessary to make certain assumptions to make the analysis more tractable.

First consider the integrals over $\boldsymbol{x}^n$ and $\boldsymbol{x}^*$ in equation 19. These are of the form

$$I_x = \int p(t|\boldsymbol{w}, \boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{x})\ d\boldsymbol{x}. \tag{20}$$

If it is assumed that the noise process is additive Gaussian $\mathcal{N}(0, \sigma_x^2)$ then the integral simplifies to

$$I_x = \int \exp\left(-\frac{\beta}{2}\left\{t - y(\boldsymbol{x}; \boldsymbol{w})\right\}^2 - \frac{1}{2\sigma_x^2}\left\{\boldsymbol{x} - \boldsymbol{z}\right\}^T\left\{\boldsymbol{x} - \boldsymbol{z}\right\}\right) d\boldsymbol{x}. \tag{21}$$

Here it has been assumed that the prior $p(\boldsymbol{x})$ is a uniform distribution. This may be a good assumption in circumstances where $p(\boldsymbol{x})$ varies much more slowly than the other distributions. In the sine wave example used in this paper $p(\boldsymbol{x})$ is constant over the training data and so this is not an unreasonable assumption. However, in general this will not be the case and it will be necessary to explicitly allow for the prior over the true data. The generality of this result therefore is limited by the imposition of this prior.

Given the above assumptions if it is *also* assumed that the noise process is small it is possible to linearise $y(\boldsymbol{x};\boldsymbol{w})$ around $\boldsymbol{z}$. Neglecting second order terms this gives,

$$y(\boldsymbol{x};\boldsymbol{w}) = y(\boldsymbol{z};\boldsymbol{w}) + \boldsymbol{h}^T \delta \boldsymbol{z}$$

where

$$\boldsymbol{h} = \nabla_x y(\boldsymbol{x};\boldsymbol{w})|_{\boldsymbol{x}=\boldsymbol{z}}$$

and $\delta\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{z}$. The integral I_x now becomes,

$$\begin{aligned} I_x &= \int \exp\left(-\tfrac{\beta}{2}\left\{t - y(\boldsymbol{z};\boldsymbol{w}) - \boldsymbol{h}^T\delta\boldsymbol{z}\right\}^2 - \tfrac{1}{2\sigma_x^2}\delta\boldsymbol{z}^T\delta\boldsymbol{z}\right) d\,\delta\boldsymbol{z} \\ &= \tfrac{1}{Z_z}\exp\left(-\tfrac{\beta'}{2}\left\{t - y(\boldsymbol{z};\boldsymbol{w})\right\}^2\right), \end{aligned} \tag{22}$$

where Z_z is a normalising constant and

$$\frac{1}{\beta'} = \frac{1}{\beta} + \sigma_x^2\ \boldsymbol{h}^T\ \boldsymbol{h}. \tag{23}$$

This result can now be used to compute the integral over $\boldsymbol{x}^*$ and $\boldsymbol{x}^n$ in equation 19. If the prior is modelled using a quadratic regularisation term of the form $E_D = ||\boldsymbol{w}||^2$ (see equation 6) the predictive distribution can be written as

$$\begin{aligned} p(t^*|\boldsymbol{z}^*, D') &= \tfrac{1}{Z_x}\int \exp\left(-\tfrac{\beta'}{2}\left\{t^* - y(\boldsymbol{z}^*;\boldsymbol{w})\right\}^2\right) \\ &\qquad \prod_{i=1}^N \exp\left(-\tfrac{\beta'}{2}\left\{t_i - y(\boldsymbol{z}_i;\boldsymbol{w})\right\}^2 - \tfrac{\alpha}{2}||\boldsymbol{w}||^2\right) d\boldsymbol{w}, \end{aligned} \tag{24}$$

where Z_x is a normalising constant.

Using the Laplace approximation this leads to:

$$p(t^*|\boldsymbol{z}^*, D') \propto \int \exp\left(-\frac{\beta'}{2}\{t^* - y(\boldsymbol{z}^*; \boldsymbol{w})\}^2\right) \exp\left(-\frac{1}{2}\Delta\boldsymbol{w}^T \boldsymbol{A}\, \Delta\boldsymbol{w}\right) d\boldsymbol{w}, \tag{25}$$

where again $\boldsymbol{A}$ is the Hessian matrix

$$\boldsymbol{A} = \nabla_w \nabla_w S_{\mathrm{MP}},$$

$\Delta\boldsymbol{w} = \boldsymbol{w} - \boldsymbol{w}_{\mathrm{MP}}$ and $S(\boldsymbol{w}_{MP})$ has been written as S_{MP}.

As before taking a linear expansion of $y(\boldsymbol{z}^*; \boldsymbol{w})$ about $\boldsymbol{w}_{\mathrm{MP}}$ the integral over $\boldsymbol{w}$ can be calculated and a Gaussian approximation of the predictive distribution obtained. That is,

$$p(t^*|\boldsymbol{z}^*, D') = \frac{1}{(2\pi\sigma_t^2)^{1/2}} \exp\left(-\frac{\{t^* - y(\boldsymbol{z}^*; \boldsymbol{w}_{\mathrm{MP}})\}^2}{2\sigma_t^2}\right). \tag{26}$$

However, now

$$\begin{aligned} \sigma_t^2 &= \tfrac{1}{\beta'} + \boldsymbol{g}^T \boldsymbol{A}^{-1} \boldsymbol{g} \\ &= \tfrac{1}{\beta} + \sigma_x^2\, \boldsymbol{h}^T\, \boldsymbol{h} + \boldsymbol{g}^T \boldsymbol{A}^{-1} \boldsymbol{g}. \end{aligned} \tag{27}$$

Compared to the previous error bar given in equation 12 this estimate of uncertainty contains an additional term. The term is related to the variance of the input noise process multiplied by the square derivative of the regression function $y(\boldsymbol{x}; \boldsymbol{w})$.

For a neural network this gives the not too surprising result that the contribution to the uncertainty of the output due to noise on the input is proportional to the variance of the noise multiplied by the square of the derivative of the output given the input. This agrees with the error bar calculated by Townsend and Tarassenko [14] who, using a variational approach, looked at the effect of perturbing the input to a RBF network.

.4. Noisy sine wave example

.4.1. Introduction

As a simple example of allowing for input noise in a network used for regression, consider the regression noisy sine wave problem illustrated in figures 3 and 4. These shows the regression over data generated from a simple sine wave $y = \sin(2\pi x), x = [0, 1]$. 20 data points are generated to which Gaussian noise $\mathcal{N}(0, \sigma_t^2)$ ($\sigma_t = 0.1$) was added to y to generate the noisy targets t^n (figure 3) and in addition to this $\mathcal{N}(0, \sigma_x^2)$ ($\sigma_x = 0.1$) added to the input values $\boldsymbol{x}$ (figure 4).

.4.2. Laplace approximation

Using MacKay's evidence approach [6] the mean and variance of the output of an MLP with five hidden units and a linear output activation unit was generated. From figure 5 it can be seen that the regression allowing for the input and target noise has a much larger variance away from the peaks of the sine wave. This is to be expected since the effect of the input noise will be to broaden the data along the x-axis which will have greatest effect where the gradient of the curve is the steepest.

It has to be remembered that this result comes from the linearisation where the noise process is assumed to be Gaussian and small and further that the prior over the true input data is uniform. In general, therefore, the uncertainty in the output will be more complicated and can only be determined by estimating the integral in equation 19.

.4.3. Monte Carlo Simulation

Rather than using a Laplace approximation it may be more accurate to perform the integrations in equation 19 using a Markov chain Monte Carlo (MCMC) approximation. Using Bayes' rule equation 19 can be rewritten in terms of the joint distribution of $\boldsymbol{w}$ and $\boldsymbol{x}^n$ given D':

$$p(t^*|\boldsymbol{z}^*, D') = \int p(t^*|\boldsymbol{w}, \boldsymbol{x}^*)p(\boldsymbol{x}^*|\boldsymbol{z}^*)p(\boldsymbol{x}^n, \boldsymbol{w}|D')\ d\boldsymbol{x}^*\ d\boldsymbol{w}\ d\boldsymbol{x}^n. \tag{28}$$

Similar integrals are discussed by Dellaportas and Stephens [3]. However, they use a Gibbs sampler to undertake the integration, a method which often has the disadvantage of slow convergence. Here the integral over the joint variables $\boldsymbol{w}$ and $\boldsymbol{x}^n$ was approximated using a Metropolis method [8]. This leaves the convolution over $\boldsymbol{x}^*$ which can be calculated numerically, as a line integral, without need for further sampling.

To demonstrate this method, a five hidden unit MLP with a linear activation output unit was again used on the noisy sine wave problem. Using the Metropolis algorithm samples of both $\boldsymbol{x}^n$ and $\boldsymbol{w}$ were taken every 100 iterations over a run of 150000 iterations, which used the standard practice of discarding the first third of the iterations as "burn-in". The hyper-parameters α and β for the regression model and $\gamma = 1/\sigma_x^2$ for the noise model were re-estimated during this process by sampling the hyper-parameters using a gamma distribution for their hyper-prior as described by Neal [10].

Convergence of the method was aided by using separate Gaussian proposal distributions for both the weights ($\sigma = 0.01$) and hidden input variables ($\sigma = 0.05$). This led to a rejection rate of approximately 50% in all the results presented.

For a regular grid of values of $\boldsymbol{z}^*$ the expectations $E[t^*|\boldsymbol{z}^*, D']$ and $E[t^{*2}|\boldsymbol{z}^*, D']$ were then evaluated by using this Metropolis algorithm to first approximate the expectations $E[t^*|\boldsymbol{x}^*, D']$ and $E[t^{*2}|\boldsymbol{x}^*, D']$ for 100 regularly spaced values of $\boldsymbol{x}^*$, three standard deviations either side of a desired value of $\boldsymbol{z}^*$, and then performing the numerical integral

$$E[f(t^*)|\boldsymbol{z}^*, D'] = \frac{1}{Z_x}\int_{-3\sigma_x}^{3\sigma_x} E[f(t^*)|\boldsymbol{x}^*, D'] \exp\left(\frac{1}{2\sigma_x^2}(\boldsymbol{x}^* - \boldsymbol{z}^*)^T(\boldsymbol{x}^* - \boldsymbol{z}^*)\right) d\ \boldsymbol{x}^*, \tag{29}$$

where the normalising constant

$$Z_x = \int_{-3\sigma_x}^{3\sigma_x} \exp\left(\frac{1}{2\sigma_x^2}(\boldsymbol{x}^* - \boldsymbol{z}^*)^T(\boldsymbol{x}^* - \boldsymbol{z}^*)\right) d\ \boldsymbol{x}^*. \tag{30}$$

As with the Laplace approximation, since $p(\boldsymbol{x})$ is uniform over the unit interval from which the data was generated a uniform prior distribution was again adopted.

It can be seen from Figures 7 and 6 that allowing for the input noise process has a marked effect on the prediction of the regression function and the estimate of the error bars. The error bars in figure 7 grossly underestimate the uncertainty. In both cases poor estimates of the true regression are obtained. This is to be expected since, as is shown in equation 14, the convolution of a sine wave with a Gaussian is a sine wave of the same frequency but with an amplitude inversely proportional to the exponent of the variance. From figure 2, which shows this smoothed function, it can be seen that the results shown in figure 6 is in reasonably good agreement.

.4.4. Reconstruction of the true regression

So far only the regression over the noisy input has been considered. However, it is possible to reconstruct the regression over the *true* noiseless input. Taking the right hand-side of equation 28 and considering the joint integral over $\boldsymbol{w}$ and $\boldsymbol{x}^n$ gives the relationship,

$$p(t^*|\boldsymbol{x}^*, D') = \int p(t^*|\boldsymbol{w}, \boldsymbol{x}^*) p(\boldsymbol{x}^n, \boldsymbol{w}|D') \; d\boldsymbol{w} \; d\boldsymbol{x}^n \tag{31}$$

Taking the expectation of the predicted target t^* gives:

$$\begin{aligned} \textstyle\int t^* p(t^*|\boldsymbol{x}^*, D') \; dt^* &= \textstyle\int t^* p(t^*|\boldsymbol{w}, \boldsymbol{x}^*) p(\boldsymbol{x}^n, \boldsymbol{w}|D') \; d\boldsymbol{w} \; d\boldsymbol{x}^n \; dt^* \\ &\equiv E[t^*|\boldsymbol{x}^*, D']. \end{aligned} \tag{32}$$

Thus it is possible to calculate the expectation of the noiseless regression given the input corrupted data D'. Figure 8 shows the reconstructed regression over the noiseless hidden input calculated using the Metropolis method. Here the error bars reflect only the uncertainty in the target noise and that induced in the weights. This can be seen by comparing this result with figure 3 which shows an MCMC construction of the regression for the same sine wave data but where there is no input noise present. The results are in close agreement although there is some deviation near the limits of the data which may be due to the approximation for the prior $p(\boldsymbol{x})$ breaking down.

.5. Scatterometer forward model

.5.1. Introduction

The ERS-1 satellite was launched in 1991 by the European Space Agency. It carries a scatterometer which measures the return radar power from three antennae that form 500 km wide a swathe to the right side of the satellite ground track. Over the ocean the surface wind field can be inferred from the scatterometer measurements by inverting, using Bayes' rule, a probabilistic forward model $p(\boldsymbol{\sigma}^o|u, v, \theta)$ where $\boldsymbol{\sigma}^o$ is the backscatter triplet, (u, v) are the wind vector components and θ the angle of beam incidence [9].

The forward model which is used operationally is the model CMOD4 [11]. CMOD4 assumes that the 3 antennae are equivalent and have a functional form defined by:

$$\sigma^o_{\mathrm{lin}} = B_0(1 + B_1\cos(\chi) + B_3\tanh(B_2)\cos(2\chi))^{1.6}, \tag{33}$$

where B_0, B_1, B_2, B_3 are complex functions of the wind speed (s) and the beam incidence angle (θ), χ is the wind direction relative to the beam look angle and σ^o_{lin} is the backscatter measured on a linear scale [13]. Unfortunately, this model is unreliable for high $(> 16ms_{-1})$ and low $(< 4ms_{-1})$ wind speeds and it has been found necessary to obtain an alternative forward model that allows for the non-linear transfer function and measurement uncertainties. However, difficulty arises when the accuracy of the sensors is compared to the accuracy of the data available to train a non-linear model. The uncertainty inherent in the satellite measurements is significantly smaller than the uncertainty in the available wind vector data. The training data can only be derived from numerical weather prediction (NWP) models which are designed for general weather prediction and give poor estimates of the surface wind vector, at the spatial resolution observed by the satellite.

The uncertainty in the training data can be seen by considering figure 9. This shows 10,000 $\boldsymbol{\sigma}^o$ triplets plotted in log space $\sigma^o = \log(\sigma^o_{\text{lin}})$ projected onto the plane $\sigma^o = 0$ for a fixed $\theta = 34.9^o$.

The mid-beam measurements depend predominantly on two geophysical variables (wind speed and direction) and it can be shown [12] that they lie close to a well defined manifold. Consequently the data may be labelled using wind vectors obtained from NWP models. For NWP wind speeds, $9 \leq s \leq 10\ ms^{-1}$, this gives the distribution of points shown in figure 10. The operational forward model, CMOD4, plotted over the same range of wind speeds gives a solid surface which shows where the points *should* lie in absence of input noise. It can be seen that the spread of the points is very large (around 5 dB) especially when compared with the uncertainty expected from the satellite system ($\sim$ 0.2 dB) which is mostly due to instrumentation noise.

The level of uncertainty in the NWP predictions is, therefore, noticeably greater than any uncertainty that would derive from the satellite system itself. As has been shown in earlier work training a neural forward model on this data without allowing for this uncertainty results in biased predictions. This is graphically illustrated in figures 11 and 12 which shows a neural network model (details of which are described later) trained using NWP data *without* allowing for the input uncertainty. Here $\boldsymbol{\sigma}^o$ has been predicted for all wind velocities and it can be seen that the fit of this surface to the data is poor.

.5.2. Noise input network solution

To obtain an un-biased forward model it is necessary to train the network allowing for the input noise and this requires a model of the noise process. The uncertainty in the data produced by the NWP models is a complex function of wind speed and direction [13]. Indeed, the speed component has a complicated skewed distribution at low wind speeds. However, when the wind is expressed in Cartesian coordinates the noise distribution *can* be approximated by a spherical Gaussian distribution but with a much smaller variance. The noise variances are set using results from [13].

Although $p(\boldsymbol{\sigma}^o|u, v, \theta)$ is required the wind vectors need not be presented to the network as vector components. At constant speed and fixed incidence angle, σ^o varies roughly as $\cos(2\chi)$ (see [7]). Thus $\cos(2\chi)$ forms an input to the network. All variables are normalised to zero mean with a common standard deviation around $0.5 - 0.7$.

The complexity of this problem precludes performing the integration of the hidden variables by sampling over the posterior. As a less computationally expensive alternative the maximum *a posteriori* value of $p(\boldsymbol{w}|D')$ is computed with a non-linear optimisation algorithm. This is undertaken be considering the four errors $E_i = -\ln(p_i)$, obtained from the expansion of equation 31.

$$p(t^*|\boldsymbol{x}^*, D') \propto \int_{\boldsymbol{w}} p(t^*|\boldsymbol{x}^*, \boldsymbol{w}) \int_{\boldsymbol{x}^n} \prod_n p_1(t^n|\boldsymbol{x}^n, \boldsymbol{w})\; p_2(\boldsymbol{z}^n|\boldsymbol{x}^n)\; p_3(\boldsymbol{x}^n)\; p_4(\boldsymbol{w})\; d\boldsymbol{x}^n\; d\boldsymbol{w}. \tag{34}$$

These are:

- $E_1 = -\ln(\prod_n p_1(\boldsymbol{\sigma}^o|s_s, \chi_s, \theta, \boldsymbol{w})$. The error of the model, calculated for the observed satellite measurements and for sampled wind vectors (s_s, χ_s) which tend to the noise free values during training.
- $E_2 = -\ln\prod_n p_2(s_s, \chi_s|s, \chi)$. The error due to the sampled wind vectors differing from their associated noisy wind vectors.
- $E_3 = -\ln(p_3(s, \chi)) = -\ln(p(s))$. The prior distribution of the true wind speeds in the training set. This is sampled approximately to be uniform in the relative direction and so depends on speed only.
- $E_4 = -\ln(p_4(\boldsymbol{w}))$. The prior over the weights which regularises the neural network [1].

Here p_1 is assumed to be spherically Gaussian in the target space, thus:

$$E_1 = \sum_{i=1}^{3} (\sigma^o_{s,i} - \sigma^o_i)^2/(2\sigma^2_{\sigma^o}), \tag{35}$$

where the sum is over the three σ^o values and the patterns in the training set, σ_{σ^o} is the standard deviation of the errors in $\boldsymbol{\sigma}^o$ measurements and σ^o_s is the output obtained propagating the sampled inputs (s_s, χ_s)

through a multi-layer perceptron (MLP) with M hidden units. This can be written:

$$\tilde{\sigma}^o = \sum_{j,k1}^{M} w_{k,j} \tanh(w_{j,1}\tilde{\theta} + w_{j,2}\sin(\chi) + w_{j,3}\cos(\chi) + w_{j,4}\cos(2\chi) + w_{j,5}\tilde{s} + w_{j,0}) + w_{k,0}, \tag{36}$$

where $\tilde{.}$ denotes the associated normalised quantities. The output is then transformed into real σ^o space by inverting the normalisation.

The distribution p_2 is assumed to be spherically Gaussian in vector components of the wind with standard deviation σ_u, so that:

$$E_2 = \sum \left((u_s - u)^2 + (v_s - v)^2\right) / (2\sigma_u^2). \tag{37}$$

The wind speed distribution, p_3, is represented by a uniform distribution between 4 and 28 ms^{-1}, similar to that in the dataset, with smooth Gaussian decrease at the ends:

$$\begin{aligned} E_3 &= (4 - s_s)^2/(2\sigma_u^2) \qquad s_s < 4\ ms^{-1} \\ E_3 &= (28 - s_s)^2/(2\sigma_u^2) \qquad s_s > 28\ ms^{-1}. \end{aligned} \tag{38}$$

Finally, p_4 corresponds to the weight decay prior:

$$E_4 = \sum_{\boldsymbol{w}} \boldsymbol{w}^2/(2\sigma_{\boldsymbol{w}}^2) \tag{39}$$

To compute the MAP value of $p(\boldsymbol{w}|D')$, a non-linear optimisation is performed using gradient information. The following derivatives are computed analytically:

$$\frac{\partial E_i}{\partial \tilde{\chi}}, \quad \frac{\partial E_i}{\partial \tilde{s}}, \quad \frac{\partial E_i}{\partial \boldsymbol{w}}, \qquad i = 1, \ldots, 4. \tag{40}$$

A training set of 10,000 patterns is used and thus there are more than 20,000 derivatives at each step in the optimisation.

.5.3. *Results*

Assessing the quality of a model is difficult and graphical representations are used to present the results. Figures 11 and 12 shows the neural network based model trained *without* objective measurements and accounting for input noise respectively. The surface defined by the model accounting for the input noise fits the target data well in $\boldsymbol{\sigma}^o$ space, while the model trained without accounting for input noise lies largely within the interior of the manifold defined by the observations. It is noted that the model trained accounting for the input noise can be seen to fit the observed $\boldsymbol{\sigma}^o$ values poorly at low wind speeds. However, CMOD4 (figure 13) which has a restricted functional form also fits the observations poorly at both high and low wind speeds.

In figure ?? it can be seen that the evolution of the wind vectors during the optimisation of the model accounting for the input noise ($\sigma^2_{\boldsymbol{w}} = 10$, $\sigma^2_u = 1.5\ ms^{-1}$, $\sigma^2_{\sigma^o} = 0.2\ dB$, 2500 iterations of the scaled conjugate gradient algorithm). The lines are radial, which suggests the vector adjustments are correlated and should not happen as the wind vectors, in the training set, are selected so that their errors are uncorrelated.

The change in direction is slightly directional dependent. Absolute wind direction has no simple geophysical meaning as it is relative to each satellite beam. Therefore, this dependency is probably due to some misfit in the model, rather than systematic errors in the NWP wind directions. The change in speed appears speed dependent, and is due to the uniform selection of data from a fixed speed range as noted earlier. Finally, the change in speed is larger than the change in direction. Describing the noise on the winds in terms of (χ, s) components might improve the model, although this effect may be due to the shape of the manifold in $\boldsymbol{\sigma}^o$ space.

.6. Discussion

This paper presents a Bayesian framework for the calculation of the predictive distribution for the output of a regression network where it is assumed not only that there is noise on the target vector but that there is also noise on the input. It is shown that, provided the conditional distribution $p(\boldsymbol{z}|\boldsymbol{x})$ can be determined (e.g. via some off-line calibration process), then it is possible to calculate the predictive distribution $p(t^*|\boldsymbol{z}^*, D')$. Furthermore, in the limit where the noise process is additive Gaussian and small the Laplace approximation gives an additional term to the error bar. This term is proportional to the input noise variance and agrees with that predicted, for the RBF network, by Townsend et al [14] using a linear perturbative approach.

The difficulty with this approach, and ultimately its limitation, is that it is necessary to have a model of the noise process and the prior over $\boldsymbol{x}$ the noiseless input. In practice the noise model is unlikely to be Gaussian. In cases where the noise is large in magnitude and non-symmetric it is necessary to estimate the predictive distribution by sampling using Monte Carlo methods. To do this the Metropolis algorithm was used to sample the joint distribution over $\boldsymbol{w}$ and $\boldsymbol{x}^n$.

This provides the prospect of being able to train an accurate model from inaccurate data, which was illustrated in the satellite data problem. Here an accurate wind vector to satellite radar forward model was constructed using noisy wind vector data which has an uncertainty far greater than that produced within the satellite system. Allowing for the input noise during the training, therefore, offers the possibility of training an accurate model of a system using inaccurate data.

A limitation of this approach is that knowledge of the input noise process is required. In some circumstances it may not be possible to determine the exact nature of the distribution over the noise process or alternatively the process may be known but may change. Here the "identification" of the noise process may be accommodated by the re-estimation of hyper-parameters in the noise model. For the noisy sine wave example explored in this paper the input noise process has a single hyper-parameter (γ), the inverse variance. However, any real sized problem this will require a large amount of data and the practicality of estimating the variance over the input noise process needs to be proved.

The need to determine the prior $p(\boldsymbol{x})$ presents a different problem. For the simulations undertaken here the prior was assumed to be a uniform distribution over the range of the training data. However, in general a more complex model for the prior will be necessary. Although the MCMC framework will support the use of a more informative prior the Laplacian approximation approach relies on the adoption of a uniform distribution.

A further problem with this approach is that the contributions to the final posterior from the noise component and regression component of the model is ambiguous. This is easy to see from figure 4 which shows the noisy data generated from a sine wave. If the data shown in this figure is considered without knowledge of how it was generated it is hard to determine if the data originates from *target noise* applied to irregular samples of the sine wave or, at the other extreme, noise on the *inputs* with no target noise. Without an appropriate noise model and prior over the noiseless input the model becomes *non-identifiable.*

Acknowledgements

The author would like to thank the members of the NCRG at Aston University, in particular David Barber, Chris Williams, David Lowe and Chris Bishop, for their help and useful discussions. WAW would like to thank The Isaac Newton Institute for Mathematics Sciences and the EPSRC grant, GR/K 51792 for the support of this work. The results presented in this paper were generated using the Aston NetLab neural network code.

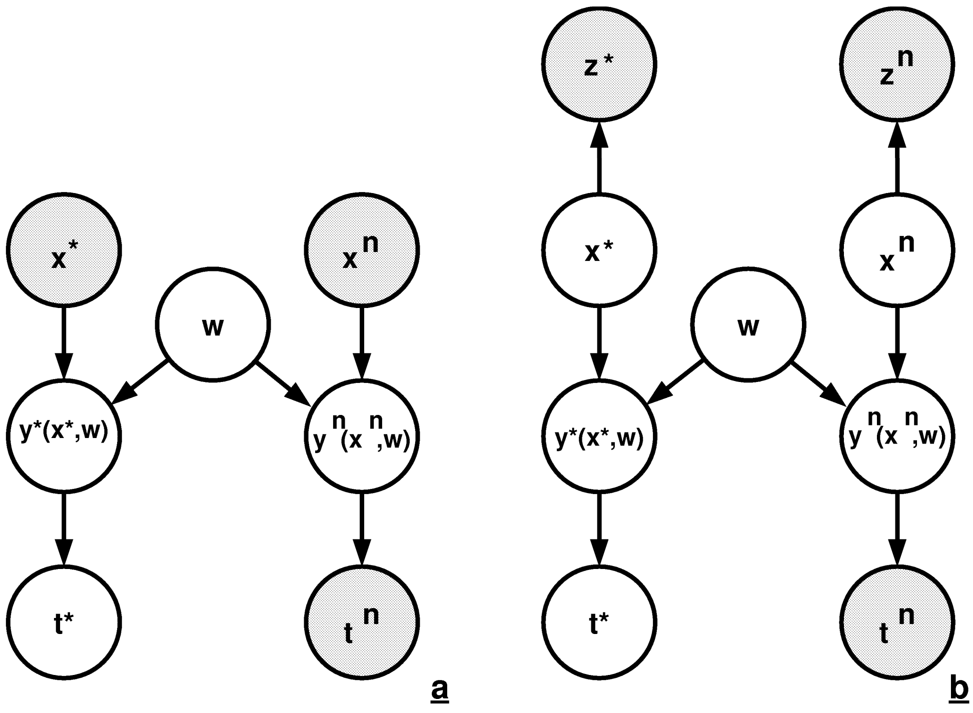

Fig. 1. Graphical representation of the regression over noisy data *without* (a) and *with* (b) the input noise model and hidden input variable. Note: t is distinguished from $y(x, w)$ to allow for the noise on the output and shaded nodes denote observed data.

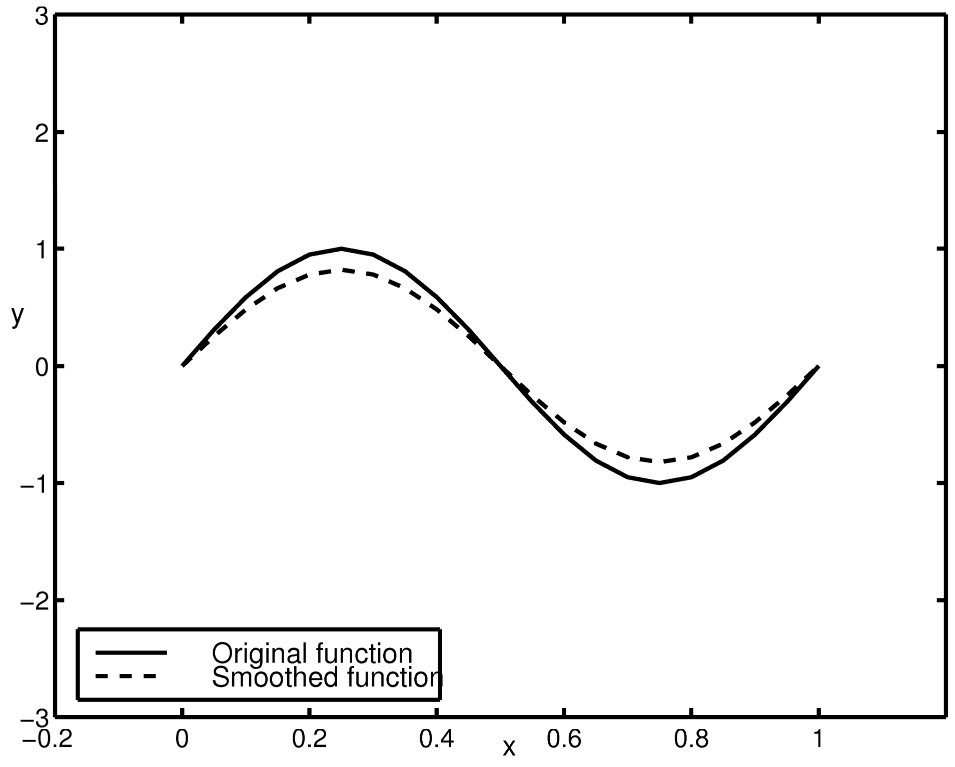

Fig. 2. Sine wave smoothed by convolving the x coordinate with the Gaussian $\mathcal{N}(0, \sigma_x^2 = 0.1)$.

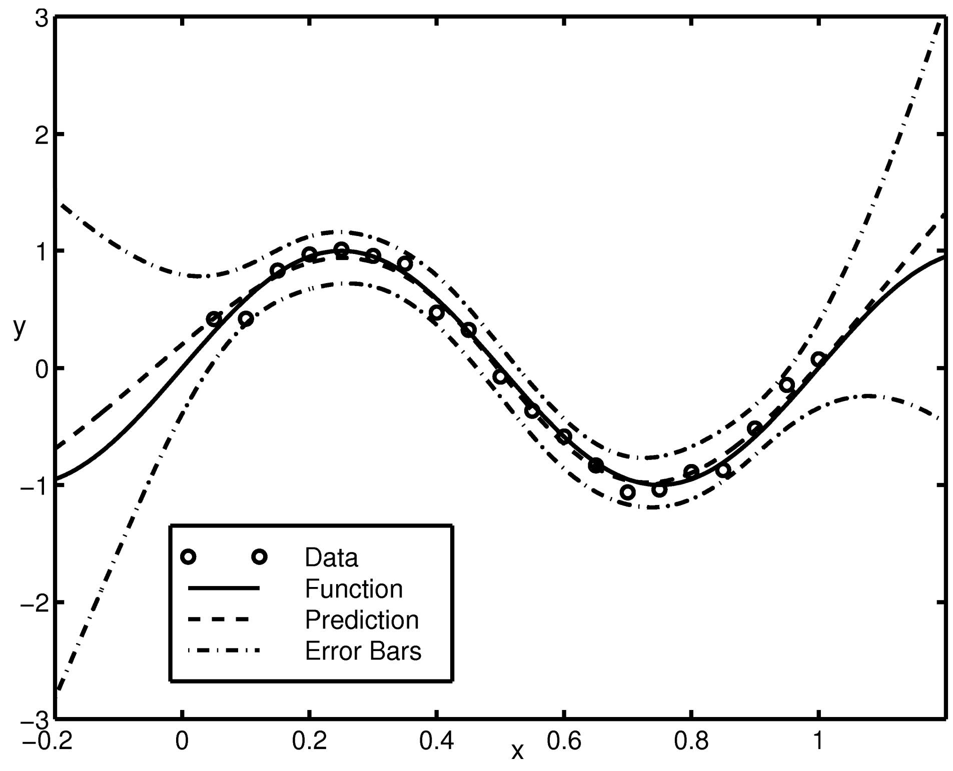

Fig. 3. Regression where no noise has been added to the input. Here the $\circ$ are the data points generated by adding noise ($\mathcal{N}(0, \sigma_t^2 = 0.1)$) to the y coordinate of 20 points sampled evenly between $x = 0 - 1$.

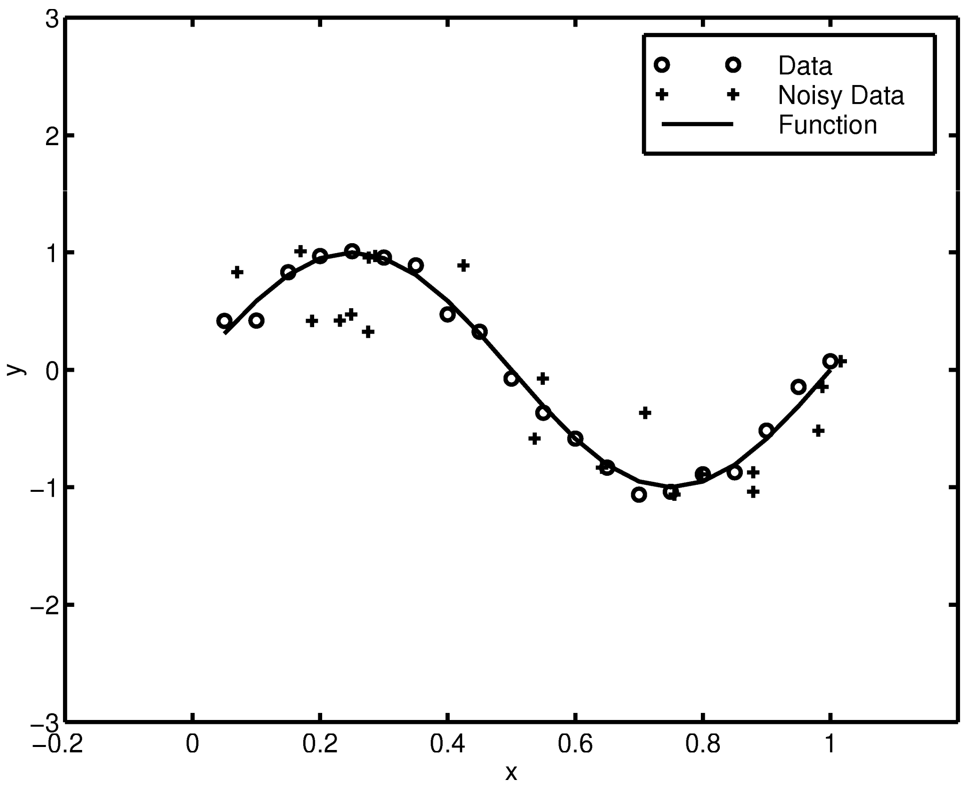

Fig. 4. Figure showing the data points generated from the sine wave. Here $\circ$ are the data points generated by adding noise ($\mathcal{N}(0, \sigma_t^2 = 0.1)$) to the y coordinate of 20 points sampled evenly for $x = [0, 1]$ and the $+$ are the same points but with noise ($\mathcal{N}(0, \sigma_x^2 = 0.1)$) also added to the x coordinate.

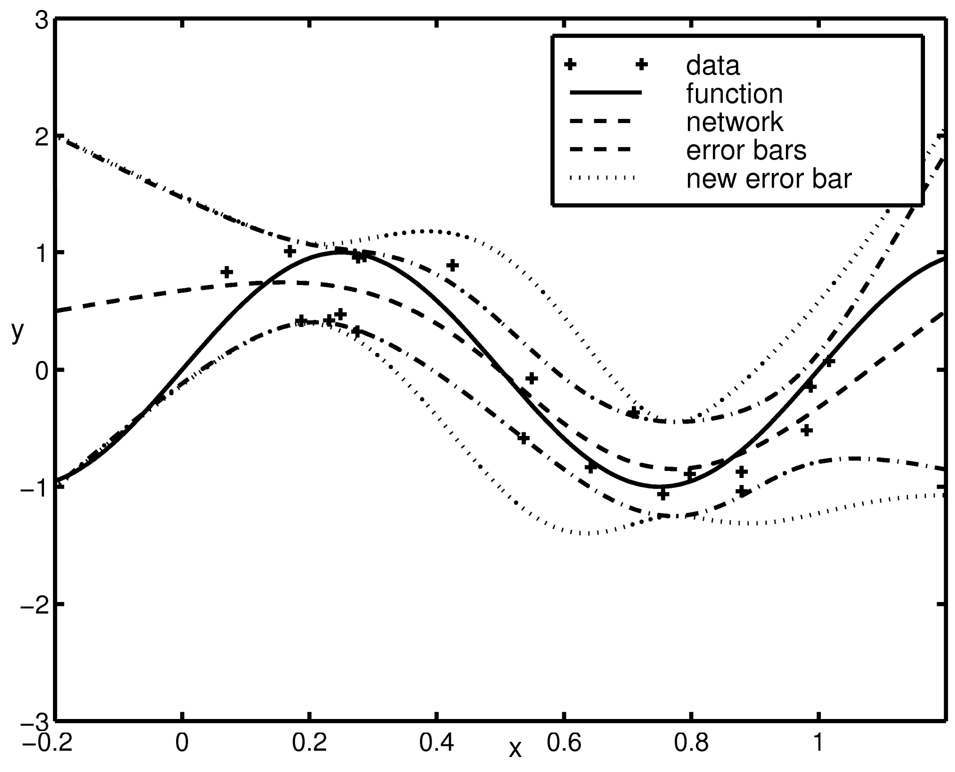

Fig. 5. Figure showing the error bars determined by using a Laplace approximation. Note that the error bars differ when the input noise *is* (the dotted line) and *is not* (the dash-dotted line) allowed for but the regression function remains the same.

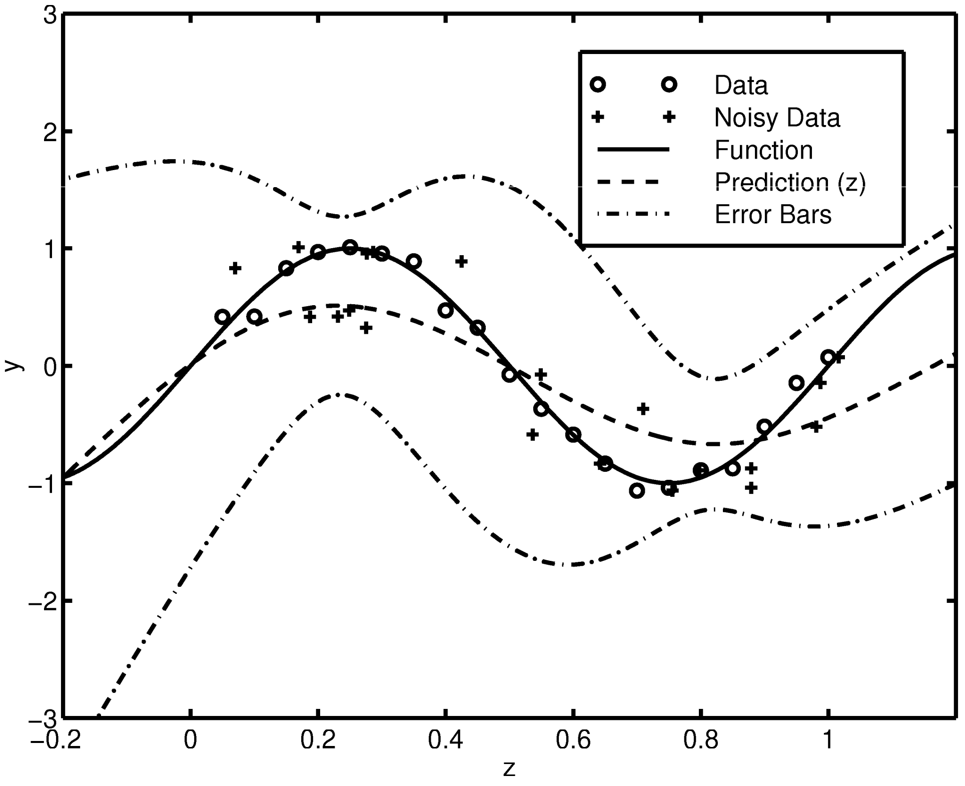

Fig. 6. Figure showing the regression over the noisy data where the input noise has been allowed for. Here $\circ$ are the data points generated by adding noise ($\mathcal{N}(0, \sigma_t^2 = 0.1)$) to the y coordinate of 20 points sampled evenly for $x = [0, 1]$ and the $+$ are the same points but with noise ($\mathcal{N}(0, \sigma_x^2 = 0.1)$) also added to the x coordinate.

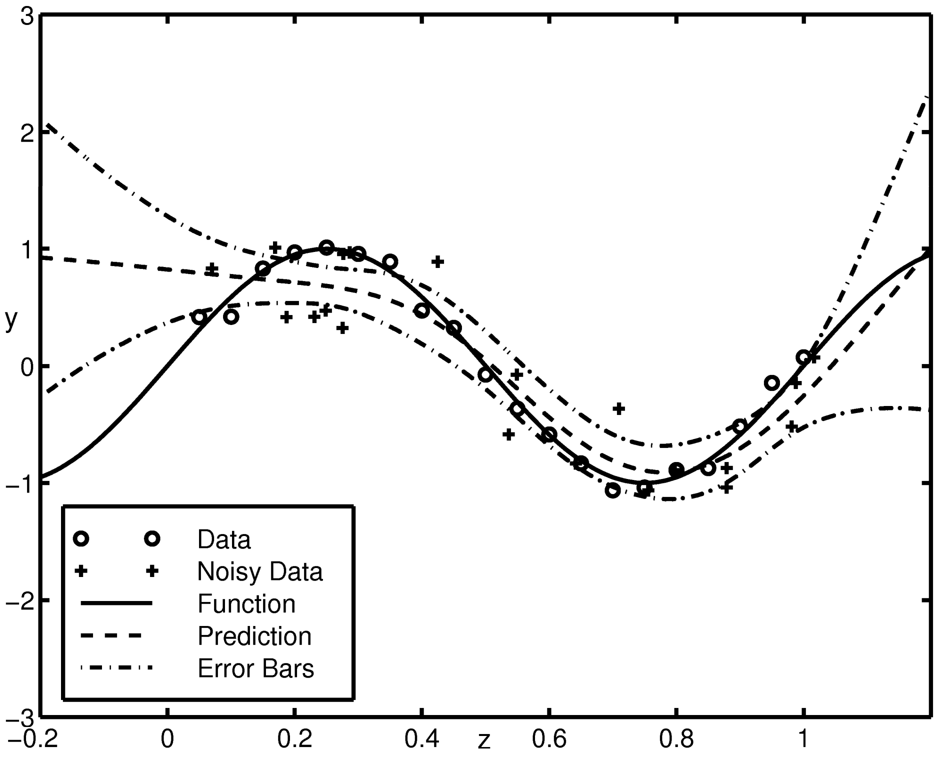

Fig. 7. Figure showing the regression over the noisy input data where the noise has *not* been allowed for. Here $\circ$ are the data points generated by adding noise ($\mathcal{N}(0, \sigma_t^2 = 0.1)$) to the y coordinate of 20 points sampled evenly for $x = [0, 1]$ and the $+$ are the same points but with noise ($\mathcal{N}(0, \sigma_x^2 = 0.1)$) also added to the x coordinate.

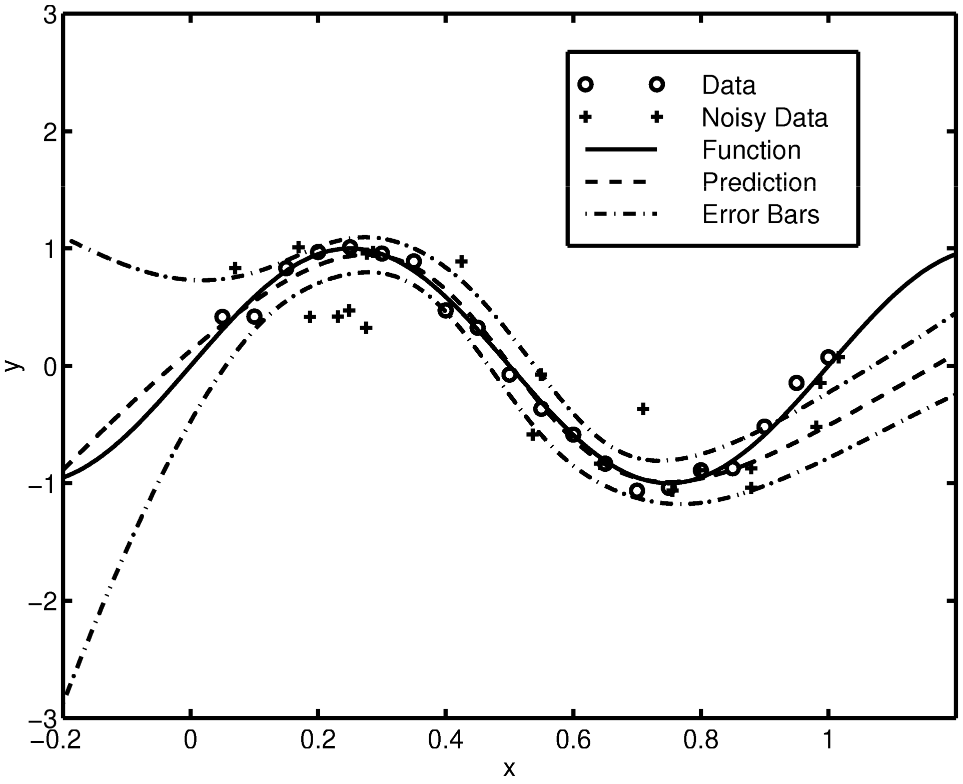

Fig. 8. Figure showing the reconstructed regression over the noiseless input but where the network was trained using noisy input data. As before the ∘ are the data points generated by adding noise ($\mathcal{N}(0, \sigma_t^2 = 0.1)$) to the y coordinate of 20 points sampled evenly for $x = [0, 1]$ and the $+$ are the same points but with noise ($\mathcal{N}(0, \sigma_x^2 = 0.1)$) also added to the x coordinate.

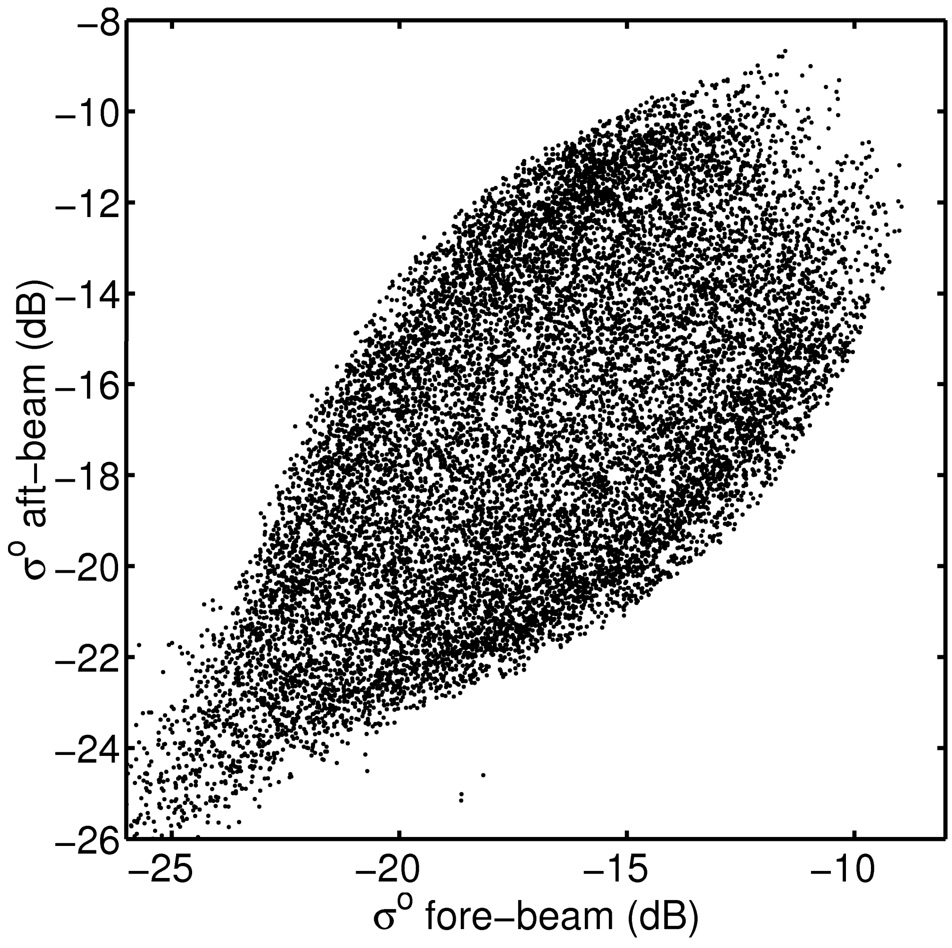

Fig. 9. The manifold in σ^o space at an incidence angle of 34.9°.

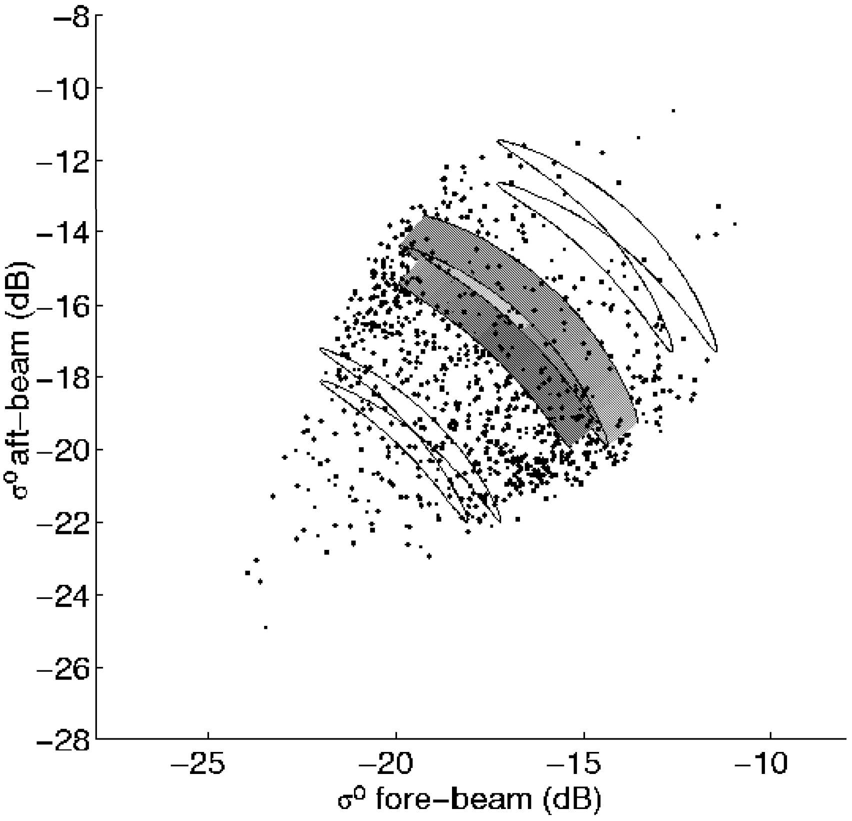

Fig. 10. Points sub-sampled from the manifold in σ^o space for $s = 0 - 9\ ms^{-1}$. The surface drawn if for the same range of speeds. The other lines represent wind speeds of 6 (bottom left) and 13 ms^{-1} (top right).

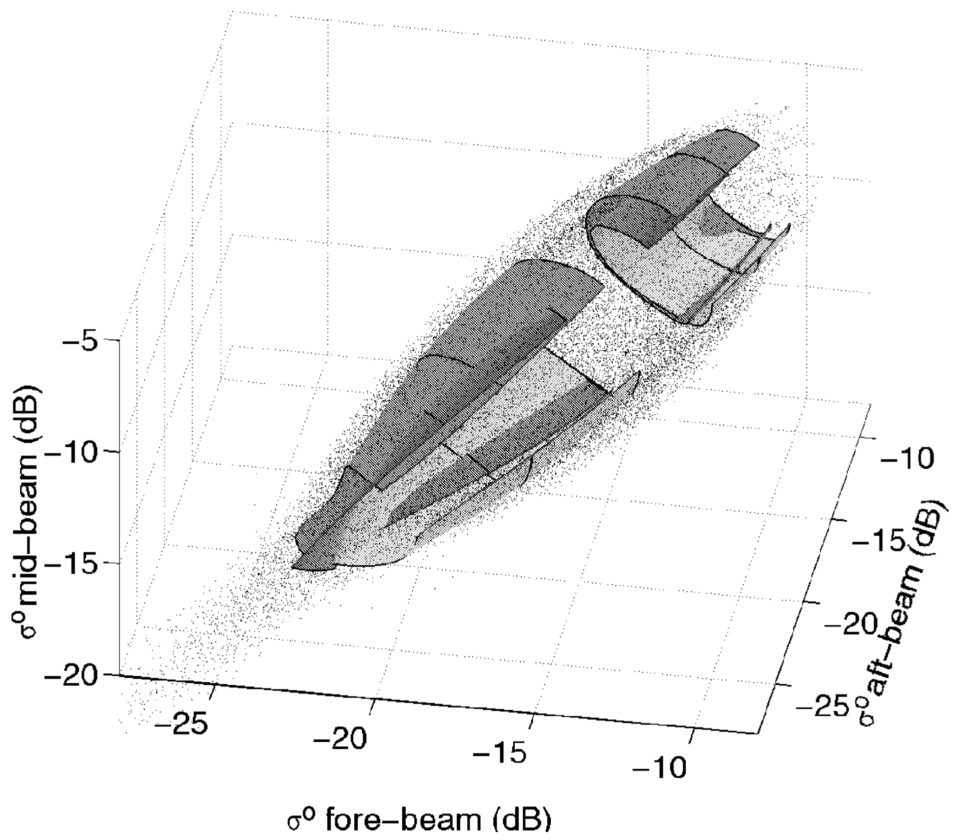

Fig. 11. In target space, the surface defined by the model when trained ignoring the effect of input noise. Note that the surface lies lie inside the target points. Some parts of the manifold are removed to allow visualisation. Shading represents wind direction, lines on the surfaces represent constant wind speeds of 4, 8, 12, 16, 20 and 24 ms^{-1}. The surface is not drawn for $12 - 16$ ms^{-1}.

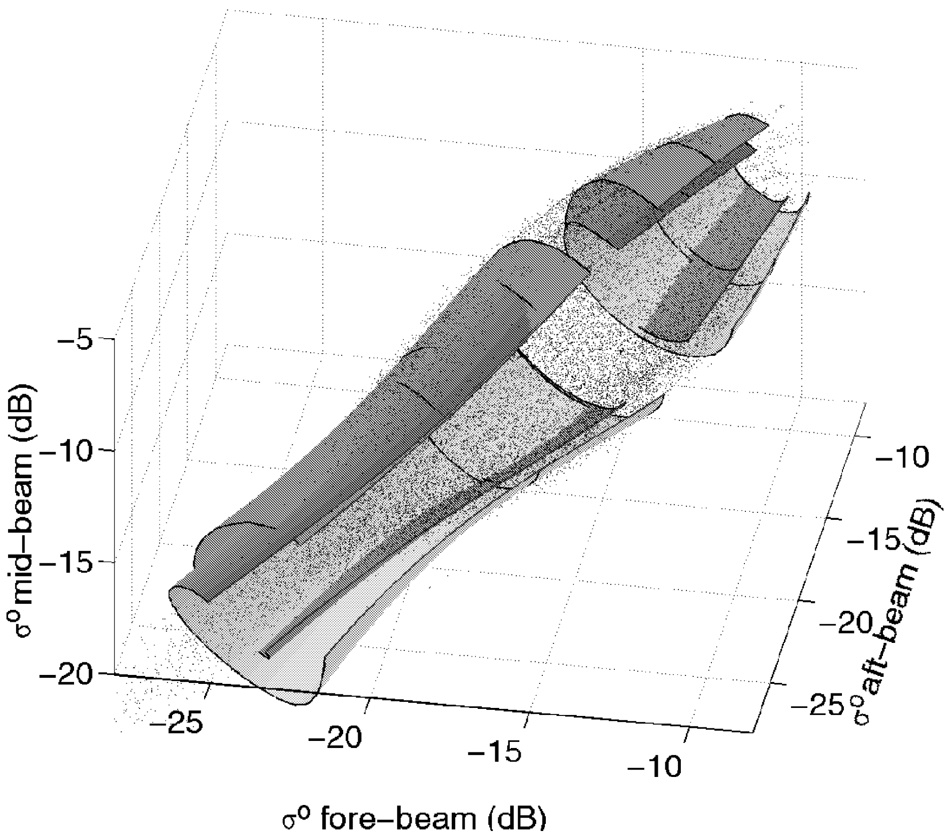

Fig. 12. In target space, the surface defined by the model when trained allowing for input noise. Note that the surface is extended and lies in the target points. Some parts of the manifold are removed to allow visualisation. Shading represents wind direction, lines on the surfaces represent constant wind speeds of 4, 8, 12, 16, 20 and 24 ms^{-1}. The surface is not drawn for $12 - 16$ ms^{-1}.

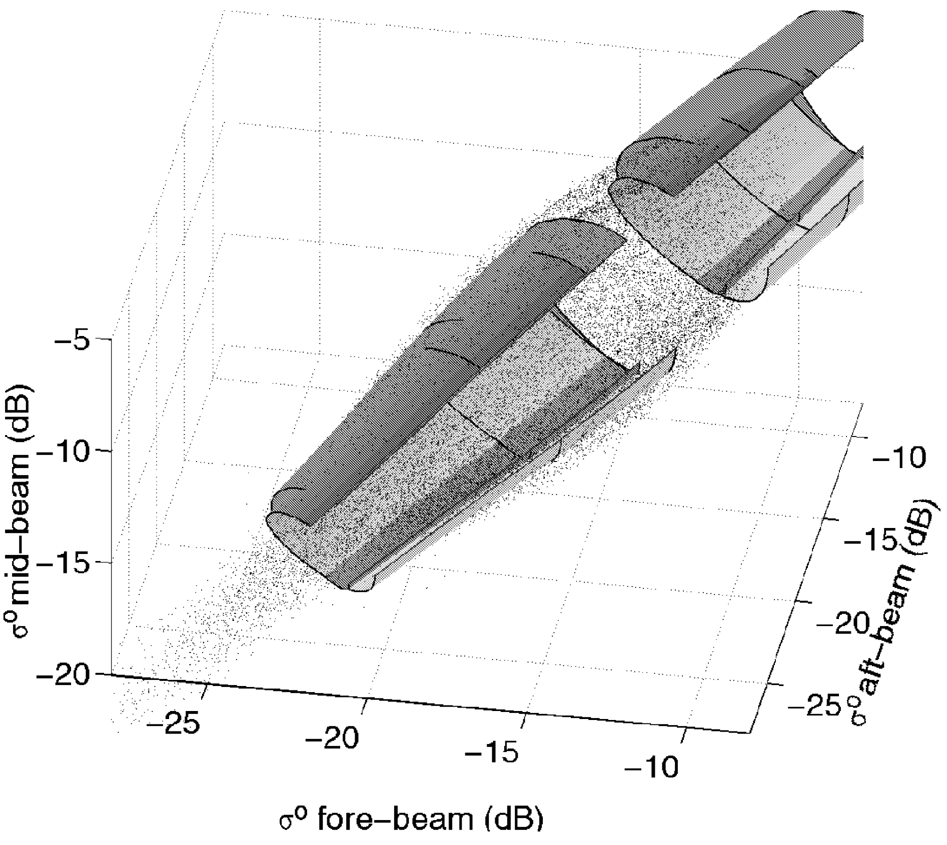

Fig. 13. Current operational model, CMOD4, plotted as in figures 11 and 12

References

1. C. Bishop. *Neural networks for pattern recognition.* Oxford University, Oxford, UK, 1995.
2. C.M. Bishop and C.S. Qazaz. Regression with input-dependent noise: A bayesian treatment. In M.C. MOzer, M.J. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 347–353. 1997.
3. S.A. Dellaporta, P. Stephens. Bayesian analysis of error in variable regression models. *Biometrics*, 51:1085–1095, 1995.
4. D. J. C. MacKay. *Bayesian Methods for Adaptive Models.* PhD thesis, California Institute of Technology, 1991.
5. D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.
6. D. J. C. MacKay. A practical Bayesian framework for back-propagation networks. *Neural Computation*, 4(3):448–472, 1992.
7. C. Mejia, S. Thiria, M. Crepon, and F. Badran. Determination of the geophisical model function of the ERS-1 scatterometer by the use of neural networks. *Journal of geophysical research*, 103:12853–12866, 1998.
8. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.
9. I.T. Nabney, D. Cornford, and C.K.I Williams. Bayesian inference for wind field retrieval. *Neurocomputing Letters*, 1998. submitted.
10. R. M. Neal. *Bayesian Learning for Neural Networks.* Springer, 1996. Lecture Notes in Statistics 118.
11. D. Offiler. The calibration of ERS-1 satellite scatterometer winds. *Journal of atmospheric and oceanic technology*, 11:1002–1017, 1994.
12. G. Ramage. Neural networks for modelling wind vectors. Master's thesis, Aston University, 1998.
13. A. Stoffelen and D. Anderson. Scatterometer data interpretation: Estimation and validation of the transfer function CMOD4. *Journal of geophisical research*, 102:5767–5780, 1997.
14. N.W. Townsend and L. Tarassenko. Estimations of error bounds for RBF networks. In *IEE Artificial Neural Networks*, pages 227–232. 1997.
15. V. Tresp, S. Ahamad, and R. Neuneier. Training neural networks with deficient data. In *Neural Information Processing Systems*, volume 6. 1994.