Neural Network Regression with Input Uncertainty

W. A. Wright

British Aerospace, Sowerby Research Centre
FPC 267, PO Box 5, Bristol, BS12 7QW, England

Abstract

It is generally assumed when using Bayesian inference methods for neural networks that the input data contains no noise or corruption. For real-world (errors in variable) problems this is clearly an unsafe assumption. This paper presents a Bayesian neural network framework which allows for input noise given that some model of the noise process exists. In the limit where this noise process is small and symmetric it is shown, using the Laplace approximation, that there is an additional 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 Markov Chain Monte-Carlo methods, it is demonstrated that it is possible to infer the unbiased regression over the *noiseless* input.

1 Introduction

It can generally be assumed that any data produced in the real world will have some degree of uncertainty. 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 neural networks. Such methods allow the predicted distribution of the output, given the model parameters and 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 uncertainty in the model parameters) [1], [7]. Usually no allowance is made for what is termed "errors in variables" or uncertainty in the input data. Allowing for such uncertainty is important where it is necessary to understand the underlying function of the network, such as in regression. There are other instances where allowing for input noise is desirable. Consider the situation where noise is introduced into a system by a sensor. Here the sensor noise may change and so the output of the system becomes a function of the sensor's dynamics if it is not allowed for. In other circumstances it may be necessary to

0-7803-5060-X/98/$10.00 © 1998 IEEE

fuse data from more than one sensor. Here allowing for the true uncertainty is advantageous since one sensor may have a different noise characteristic than the other.

A number of researchers have considered the errors in variable problem for neural networks. Tresp et al [6], in considering the issue of 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 [5] have produced a similar result by taking a perturbative approach. Their method also allows for the error induced in the weights. They show that for small 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 to the calculation of the predictive distribution for the network has been taken. It is shown that, given a model of the input noise process, it is possible to obtain an estimate of the posterior distribution on the output which allows for the uncertainty due to the input noise. A result which in the limit where the input noise is additive, symmetric and small, agrees with the results of Townsend and Tarassenko. Although both these approaches give a more accurate estimate of the uncertainty Tresp et al show that the final regression will be biased. However, it is shown here that by sampling using a Markov Chain Monte-Carlo (MCMC) method [4] over the noiseless input variable it is possible, given an appropriate prior over the noiseless data, to infer the unbiased regression (i.e. that given the *noiseless* input).

2 Neural Network Regression with Noisy Input

Consider the 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 target $t^n = t_1, \ldots, t_N$. Here $D = \{t^n, \boldsymbol{x}^n\}$ 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

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

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

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

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 $\boldsymbol{w}$ then, using the Laplace approximation [1], the predictive

distribution may be approximated as:

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

Here $\boldsymbol{w}_{\mathrm{MP}}$ is the most probable weight vector obtained through the expansion[1] and minimisation of

$$S(\boldsymbol{w}) = -\frac{\beta}{2}\sum_{i=1}^{N}\{y(\boldsymbol{x}_i; \boldsymbol{w}) - t_i\}^2 - \frac{\alpha}{2}||\boldsymbol{w}||^2$$

with respect to $\boldsymbol{w}$, $\beta = 1/\sigma_t^2$ and similarly α is the inverse of the varience in $\boldsymbol{w}$. It has been assumed that $p(\boldsymbol{w}) \propto \exp(\frac{-\alpha}{2}||\boldsymbol{w}||^2)$. The variance

$$\sigma_d^2 = \frac{1}{\beta} + \boldsymbol{g}^T \boldsymbol{A}^{-1} \boldsymbol{g}, \tag{3}$$

where $\boldsymbol{g} = \nabla_w y(\boldsymbol{x}^*; \boldsymbol{w})|_{\boldsymbol{w}=\boldsymbol{w}_{\mathrm{MP}}}$ and $\boldsymbol{A}$ is the Hessian $\boldsymbol{A} = \nabla_w \nabla_w S(\boldsymbol{w}_{\mathrm{MP}})$, provides an unbiased estimate of the uncertainty (or "error bar") about the predicted mean $y_{\mathrm{MP}} = E[t^*|\boldsymbol{x}^*, D]$.

Unfortunately for many real problems this estimate of uncertainty is incomplete, as it is likely that the input will also be uncertain. To allow for this it is necessary to look at how it would enter the inferencing system. Assume that the random vector $\boldsymbol{x}$, the *true* input, is hidden and so cannot be observed but samples from another random vector $\boldsymbol{z}$, the *noisy* input, can where, $\boldsymbol{z} = f_x(\boldsymbol{x}, \boldsymbol{\gamma})$. Here $\boldsymbol{\gamma}$ is a random noise vector, independent of $\boldsymbol{x}$, with distribution $p_\gamma(\boldsymbol{\gamma})$. In general, since only $\boldsymbol{z}$ can be observed then only data $D' = \{t^n, \boldsymbol{z}^n\}$ exists, where $t^n = t_1, \ldots, t_N$ and $\boldsymbol{z}^n = \boldsymbol{z}_1, \ldots, \boldsymbol{z}_N$. However, if there is some model of the input noise process then the inference system can be represented as two separate components.

- A generative component from which the noisy inputs are produced given unknown noiseless inputs. This could be obtained off line through some calibration process.
- The regression model which takes the noiseless inputs and generates an appropriate output.

Given these two components the predictive distribution $p(t^*|\boldsymbol{z}^*, D')$ for the noisy input regression can be expressed in terms of the marginal distribution

$$p(t^*|\boldsymbol{z}^*, D') = \int p(t^*|\boldsymbol{x}^*, D')\; p(\boldsymbol{x}^*|\boldsymbol{z}^*)\; d\boldsymbol{x}^*, \tag{4}$$

where now $\boldsymbol{x}^*$ is the new (noiseless or latent) input. After some manipulation expanding the integral over the latent data points $\boldsymbol{x}^n$ and exploiting the

[1] A quadratic regularisation function for the form of the prior in the weights has been used here.

independence of $\boldsymbol{w}$ on $\boldsymbol{x}^*$ and t^n on $\boldsymbol{z}^n$ given $\boldsymbol{x}^n$ then:

$$\begin{aligned} p(t^*|\boldsymbol{z}^*, D') &= \frac{1}{Z_{tz}} \int p(t^*|\boldsymbol{w}, \boldsymbol{x}^*) p(\boldsymbol{z}^*|\boldsymbol{x}^*) p(\boldsymbol{x}^*) \\ &\quad p(t^n|\boldsymbol{x}^n, \boldsymbol{z}^n) p(\boldsymbol{z}^n|\boldsymbol{x}^n) p(\boldsymbol{x}^n) p(\boldsymbol{w})\ d\boldsymbol{w}\ d\boldsymbol{x}^n d\boldsymbol{x}^*. \end{aligned} \tag{5}$$

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 $\boldsymbol{x}^n$. Considering the different components of equation 5 is possible to see how they contribute to the posterior and so to the uncertainty of the expectation $E[t^*|\boldsymbol{z}^*, D']$. The integral over $\boldsymbol{w}$, in the usual way, may be interpreted as providing contributions to the posterior which allows 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 the contribution to the predictive distribution that allows for the uncertainty in the new input.

2.1 Laplace approximation of the predictive distribution

Generally the integral in equation 5 can only be effectively estimated by using MCMC methods. However, it is instructive to evaluate analytically the error bars using the Laplace approximation. To do this it is necessary to make certain assumptions to make the calculation more tractable. Consider the integrals over $\boldsymbol{x}^n$ and $\boldsymbol{x}^*$ in equation 5. These are of the form

$$I_x = \int p(t|\boldsymbol{w}, \boldsymbol{x}) p(\boldsymbol{z}|\boldsymbol{x}) p(\boldsymbol{x})\ d\boldsymbol{x}. \tag{6}$$

If $p(\boldsymbol{x})$ is assumed[2] to vary much more slowly than the other distributions and the noise process is additive Gaussian $\mathcal{N}(0, \sigma_x^2)$ then the integral simplifies to

$$I_x = \int \exp\left(-\frac{\beta}{2}\{t - y(\boldsymbol{x}; \boldsymbol{w})\}^2 - \frac{1}{2\sigma_x^2}\{\boldsymbol{x} - \boldsymbol{z}\}^T\{\boldsymbol{x} - \boldsymbol{z}\}\right) d\boldsymbol{x}. \tag{7}$$

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

$$I_x = \frac{1}{Z_z} \exp\left(-\frac{\beta'}{2}\{t - y(\boldsymbol{z}; \boldsymbol{w})\}^2\right), \tag{8}$$

where Z_z is a normalising constant, $\frac{1}{\beta'} = \frac{1}{\beta} + \sigma_x^2\, \boldsymbol{h}^T\, \boldsymbol{h}$ and $\boldsymbol{h} = \nabla_x y(\boldsymbol{x}; \boldsymbol{w})|_{\boldsymbol{x}=\boldsymbol{z}}$.

[2]For the demonstration used in this paper this is a reasonable assumption since the prior ($p(x)$) is uniform over the training data. However, in general this this may not be the case.

Undertaking the integral over $\boldsymbol{x}^*$ and $\boldsymbol{x}^n$ in equation 5, again using a quadratic regularisation term for the prior over the weights and the Laplace approximation, the predictive distribution may be approximated by the Gaussian:

$$p(t^*|\boldsymbol{z}^*, D') = \frac{1}{(2\pi\sigma_d^2)^{1/2}} \exp\left(-\frac{\{t^* - y(\boldsymbol{z}^*; \boldsymbol{w}_{\mathrm{MP}})\}^2}{2\sigma_d^2}\right). \tag{9}$$

However, now

$$\sigma_d^2 = \frac{1}{\beta} + \sigma_x^2 \, \boldsymbol{h}^T \, \boldsymbol{h} + \boldsymbol{g}^T \boldsymbol{A}^{-1} \boldsymbol{g}. \tag{10}$$

Compared to the previous error bar (equation 3) 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 a *small* level of *Gaussian* 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 [5] who using a variational approach looked at the effect of perturbing the input to the network.

As an illustration of the effect of this additional term the regression over 20 data points generated from a sine wave ($y = \sin(2\pi x)$, $x = [0, 1]$) with Gaussian additive noise on both the targets $\mathcal{N}(0, \sigma_t^2)$ ($\sigma_t = 0.1$) and the inputs $\mathcal{N}(0, \sigma_x^2)$ ($\sigma_x = 0.1$) was considered. Here the evidence approach [2] was used to approximate the mean and variance of the output of an MLP with five hidden units and a linear output activation unit. From figure 1 it can be seen that the regression allowing for the input 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.

2.2 Monte-Carlo Simulation

For a non-symmetric input noise process it is necessary to perform the integrations in equation 5 using a MCMC approximation. Using Bayes' rule equation 5 can be rewritten in terms of the joint distribution of $\boldsymbol{w}$ and $\boldsymbol{x}^n$ given D':

$$p(t^*|\boldsymbol{z}^*, D') = \int p(t^*|\boldsymbol{w}, \boldsymbol{x}^*) p(\boldsymbol{x}^*|\boldsymbol{z}^*) p(\boldsymbol{x}^n, \boldsymbol{w}|D') \, d\boldsymbol{x}^* \, d\boldsymbol{w} \, d\boldsymbol{x}^n. \tag{11}$$

Here the integral over the joint variables $\boldsymbol{w}$ and $\boldsymbol{x}^n$ can be approximated using a Metropolis method [3]. This leaves the line integral over $\boldsymbol{x}^*$ which can be undertaken separately numerically.

To demonstrate the MCMC approach the regression over the noisy sine wave problem with a five hidden unit MLP with linear activation output units was again considered. 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

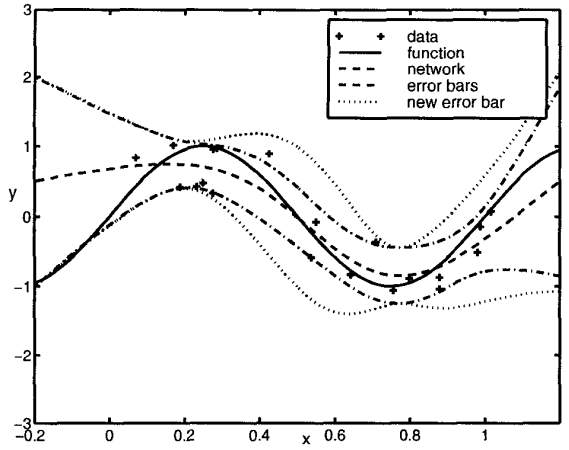

Figure 1: Figure showing the error bars determined by using a Laplace approximation. Note that the error bars differ when the input noise is allowed for but the regression function remains the same.

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 from the posterior of the hyper-parameter using a gamma distribution for their prior as described by Neal [4]. 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 achieved a rejection rate of approximately 50% in all the results presented.

It can be seen from figures 2 and 3 that allowing for the input noise has a marked effect on the prediction of the regression function and the estimate of the error bars. The error bars in figure 2 grossly underestimate the uncertainty, whereas figure 3 produces a similar result to that in figure 1. In both cases poor estimates of the true regression are obtained. This is expected since it can be shown that 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.

It is possible to reconstruct the regression over the *true noiseless* input. Taking the right hand-side of equation 11 and considering the joint integral over $\boldsymbol{w}$ and $\boldsymbol{x}^n$ gives the relationship,

$$p(t^*|\boldsymbol{x}^*, D') = \int p(t^*|\boldsymbol{w}, \boldsymbol{x}^*) p(\boldsymbol{x}^n, \boldsymbol{w}|D')\, d\boldsymbol{w}\, d\boldsymbol{x}^n. \tag{12}$$

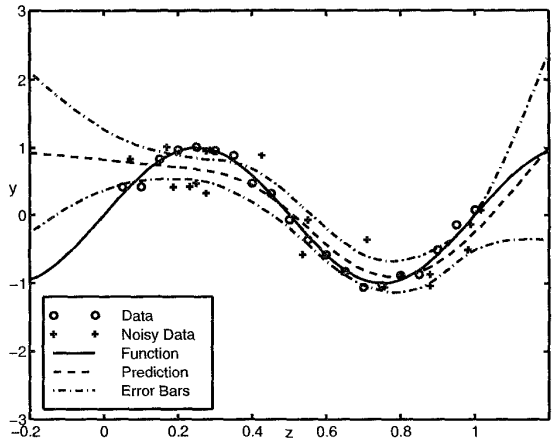

Figure 2: Figure showing the regression over noisy input data where the noise has not been allowed for. Here $\sigma_x = 0.1$.

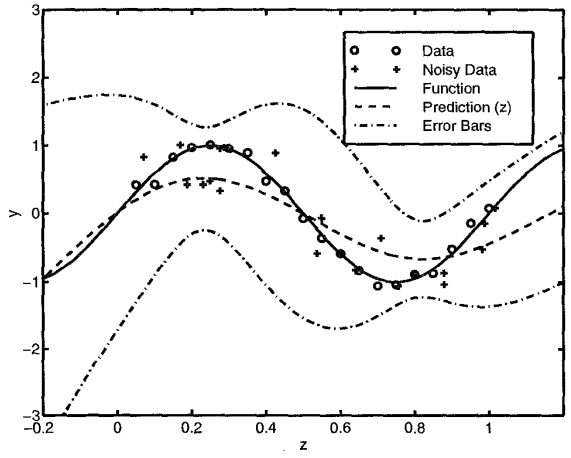

Figure 3: Figure showing the regression over the noisy data where the input noise has been allowed for. Here $\sigma_x = 0.1$.

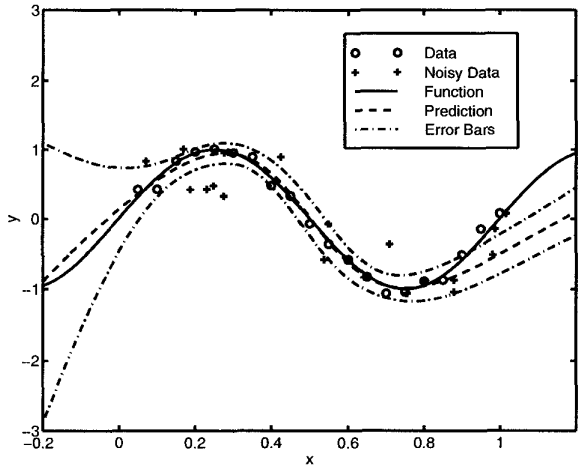

Figure 4: Figure showing the reconstructed regression over the (true) noiseless input for where the training data had input noise $\sigma_x = 0.1$.

It follows that by sampling over $\boldsymbol{w}$ and $\boldsymbol{x}^n$ using the Metropolis method it is possible to calculate the expectation of the noiseless regression given the input corrupted data D'. Figure 4 shows the reconstructed regression over the noiseless hidden input calculated using the Metropolis method where the input noise has been allowed for. Here the error bars reflect only the uncertainty in the target noise and that induced in the weights. This can be seen by comparing these results with figure 5 which shows the regression for the same data but where there is no input noise present.

3 Discussion

This paper presents a Bayesian framework for the calculation of the predictive distribution for the output of the regression system where it is assumed not only that there is noise on the target vector but that there is also noise on the input. Here 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 both the MLP and RBF networks, by Townsend et al [5] using a linear perturbative approach.

The difficulty with this approach, and ultimately its limitation, is that

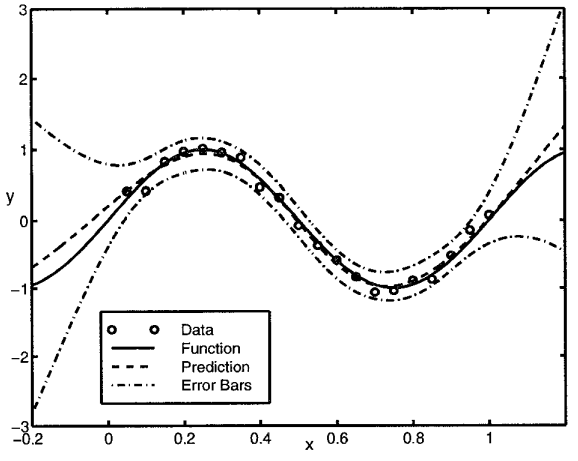

Figure 5: Regression where no noise has been added to the input.

it is necessary to have a model of the noise process and the prior over $\boldsymbol{x}$. 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 MCMC methods. For the simulations undertaken here the Metropolis algorithm was used to sample the joint distribution over $\boldsymbol{w}$ and $\boldsymbol{x}^n$. Importantly by adopting this approach it is shown that it is also possible to infer the unbiased regression, that given for the *noiseless* input $\boldsymbol{x}^*$.

In some circumstances it may not be possible to determine the exact nature of the distribution over the noise process. Here the "identification" of noise process may be accommodated by the re-estimation of hyper-parameters in the noise model. The determination of $p(\boldsymbol{x})$ presents a different problem. Here it was assumed that the prior is uniform over the unit interval from which the training data was selected. This approximation breaks down at the edge of the sampled data, as can be seen in figure 4. Generally a more complex model for the prior will be necessary.

4 Acknowledgements

The author would like to thank the members of the NCRG at Aston University, in particular David Barber and Chris Williams, for both their help and useful discussions of this work. The author would also like to thank The Isaac Newton Institute for Mathematics Sciences and the EPSRC grant, GR/K 51792 for the support of this work.

References

[1] D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

[2] D. J. C. MacKay. A practical Bayesian framework for back-propagation networks. *Neural Computation*, 4(3):448–472, 1992.

[3] 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.

[4] R. M. Neal. *Bayesian Learning for Neural Networks.* "Lecture Notes in Statistics 118". Springer, 1996.

[5] N.W. Townsend and L. Tarassenko. Estimations of error bounds for RBF networks. In *IEE Artificial Neural Networks*, pages 227 – 232, 1997.

[6] V. Tresp, S. Ahamad, and R. Neuneier. Training neural networks with deficient data. In J. D. Cowan, G. T. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 128 – 135. Morgan Kaufmann, 1994.

[7] C. K. I. Williams, C. Qazaz, C. M. Bishop, and H. Zhu. On the relationship between Bayesian error bars and the input data density. In *Proceedings Fourth IEE International Conference on Artificial Neural Networks*, pages 160–165, Cambridge, UK, 1995. IEE.