Nonlinear modeling with confidence estimation using Bayesian neural networks

A.T.C. Goh and C.G. Chua
Nanyang Technological University,
School of Civil and Environmental Engineering, University, Singapore 639798

ABSTRACT
There is a growing interest in the use of neural networks in civil engineering to model complicated nonlinearity problems. A recent enhancement to the conventional back-propagation neural network algorithm is the adoption of a Bayesian inference procedure that provides good generalization and a statistical approach to deal with data uncertainty. A review of the Bayesian approach for neural network learning is presented. One distinct advantage of this method over the conventional back-propagation method is that the algorithm is able to provide assessments of the confidence associated with the network's predictions. Two examples are presented to demonstrate the capabilities of this algorithm. A third example considers the practical application of the Bayesian neural network approach for analyzing the ultimate shear strength of deep beams.

KEY WORDS
Back-propagation neural network, Bayesian neural network, deep beams, neural network, non-linear modeling, uncertainty.

1 Introduction

Neural networks are an emerging computational tool that offers a new strategy to analyze complicated multivariate problems. Neural networks are computer algorithms based loosely on modelling the stimulus-response neuronal structure of the brain. They are typically used to learn an input-output mapping of a set of example patterns. The functional relationships between the input-output variables are "learned" without the need to specify the relationship between variables. They are particularly useful for problems in which there is a lack of a complete understanding of the relationship between the variables [1, 2, 3, 4].

To date, most of the neural network applications in civil engineering have focused on the use of the back-propagation learning algorithm [5] because of the simplicity of the methodology. As shown in Fig. 1, the architecture of a back-propagation neural network is composed of an input layer, one or more hidden layers, and an output layer. Each layer contains neurons that are fully connected with neurons in the neighboring layers by weights.

The objective of training is to modify the connection weights to reduce the errors between the actual output values and the target output values to a satisfactory level. This is carried out through the minimization (optimization) of the sum squared error function using the gradient descent approach. In mathematical terms, the algorithm essentially seeks in a step-wise method to search for the optima in a high-dimensional weight space with the objective of minimizing the sum squared error. At the end of training, the associated trained weights of the model are tested with a separate data set, to assess the generalization capability of the neural network model to give good predictions on a set of data that the network has not seen during training.

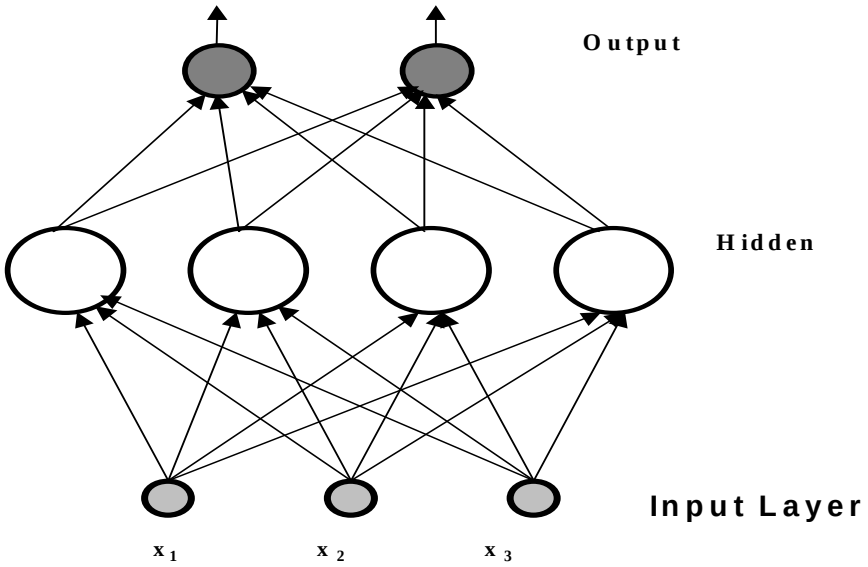

Figure 1: Neural network architecture

Generally, the neural network predictions become more accurate as the number of hidden layers and/or neurons increases. However, if too complicated an architecture is used, "overfitting" will occur. The training data will be well modeled and the sum squared error will be small. However, the network will be modeling the noise in the data as well as the trends, as illustrated for a simple example of a single input-output relationship in Fig. 2. Therefore, the network will not generalize well on the testing data set. To overcome overfitting, the technique of early stopping is commonly used. This approach involves monitoring the generalization error of the testing data set and to stop training when the minimum testing error is observed. However, some care and judgment is needed to decide when to stop, since the error surface of the weight space is in general not a smooth surface, but contains many local and global minima, and/or long flat regions preceding a steep drop-off.

2 Bayesian neural network approach

To overcome the limitations of the conventional back-propagation neural network, Mackay [6] and Neal [7] proposed the use of Bayesian inference to analyze the neural network data. In conventional back-propagation neural networks the weights are assigned deterministic values. In the Bayesian framework, the weights of the neural network are considered random variables and are characterized by a joint probability distribution representing the degree of belief in the different values of the weight vector. The objective in training is to maximize the posterior distribution over the weights $\boldsymbol{w}$, to obtain the most probable parameters values $\boldsymbol{w}_{\mathrm{MP}}$ in the network. The posterior distribution is then used to evaluate the predictions of the trained network for new values of the input variables.

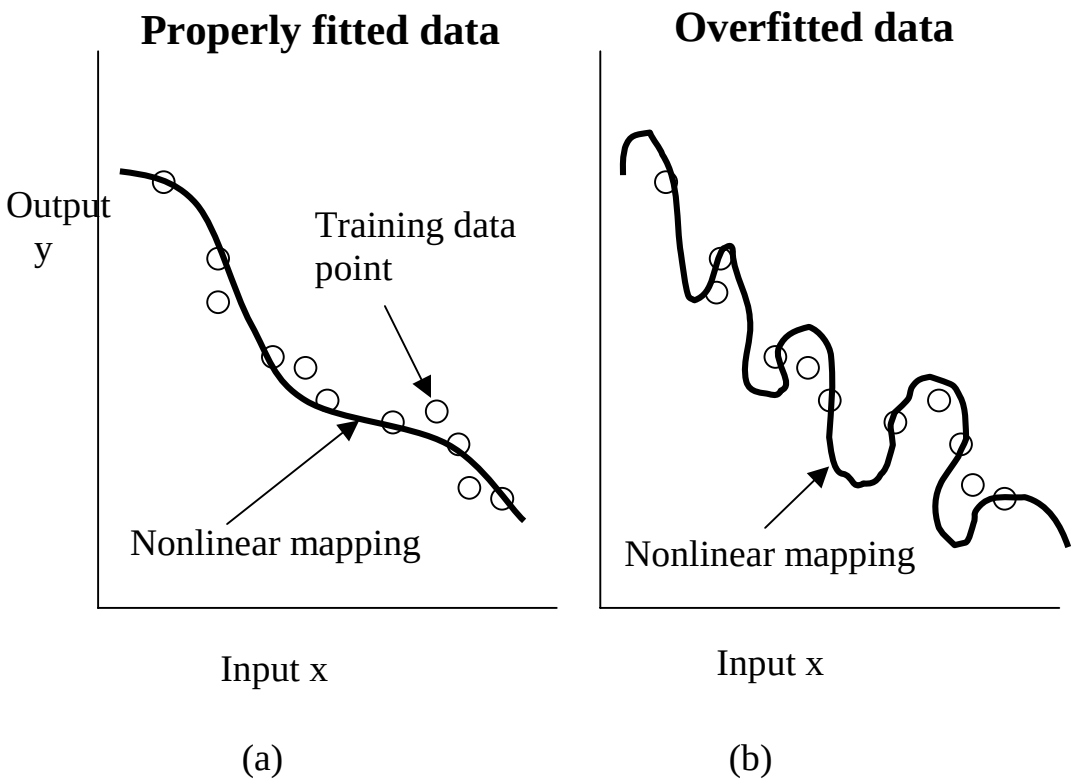

Figure 2: A simple illustration of a well-fit model and an overfitted model.

The Bayesian approach involves the optimization of the objective function $S(\boldsymbol{w})$ that comprises the conventional sum squared error function E_D as well as an additional weight error term E_W which is the sum of the square of all the weights. $S(\boldsymbol{w})$ is defined as

$$S(\boldsymbol{w}) = \theta E_D + \xi E_W \tag{1}$$

in which θ and ξ are termed regularization parameters or hyperparameters. The motivation for including E_W is to penalize the more complex weight functions in favor of simpler functions. When the weights are kept small, the neural network response will be smooth. This decreases the tendency of the neural network to fit the noise in the training data. The hyperparameter ξ controls the weight distribution of the network model and hence its nonlinear mapping ability. Noise present in the data is expressed as θ.

Consider the data set D with N training patterns of inputs $\boldsymbol{x}$ and corresponding targets t, and W number of connection weights. For a neural network model with weights $\boldsymbol{w}$, applying Bayes' rule, the posterior distribution of the weights given the observed data $p(\boldsymbol{w}|D)$ can be written as

$$p(\boldsymbol{w} \mid D) = \frac{1}{p(D)} p(D \mid \boldsymbol{w}) p(\boldsymbol{w}) \tag{2}$$

$p(D|\boldsymbol{w})$ is termed the "likelihood" function, $p(\boldsymbol{w})$ is the "prior", and $p(D)$ is a normalizing factor called the "evidence".

Mackay [6] showed that $p(\boldsymbol{w}|D)$ can be expressed as

$$p(\boldsymbol{w} \mid D) = \frac{1}{Z_S} exp\left(-\theta E_D - \xi E_W\right) = \frac{1}{Z_S} exp\left(-S(\boldsymbol{w})\right) \tag{3}$$

in which Z_S is a normalizing constant given by

$$Z_S = \int exp(-S(\boldsymbol{w}))d\boldsymbol{w} \tag{4}$$

The weight vector $\boldsymbol{w}_{MP}$ corresponding to the maximum posterior distribution $p(\boldsymbol{w}|D)$ is found by minimizing the negative logarithm of Eq. (3) with respect to the weights. Since Z_S in Eq. (4) is independent of the weights, this is equivalent to minimizing $S(\boldsymbol{w})$ given in Eq. (1).

Because the normalizing factor Z_S in Eq. (4) cannot be evaluated analytically, Mackay [6] used a Gaussian approximation for the posterior distribution by considering the Taylor expansion of $S(\boldsymbol{w})$ around its minimum value and retaining terms up to second order so that

$$S(\boldsymbol{w}) \approx S(\boldsymbol{w}_{MP}) + \frac{1}{2}\Delta\boldsymbol{w}^T A \Delta\boldsymbol{w} \tag{5}$$

in which $\boldsymbol{A}$ is the Hessian (second partial derivative) matrix of the total regularized error function given in Eq. (1)

$$\boldsymbol{A} = \nabla_w \nabla_w S(\boldsymbol{w}_{MP}) \tag{6}$$

and

$$\Delta\boldsymbol{w} = \boldsymbol{w} - \boldsymbol{w}_{\mathrm{MP}}. \tag{7}$$

The expansion of Eq. (5) leads to a posterior distribution that is now a Gaussian function of the weights, given by

$$p(\boldsymbol{w} \mid D) = \frac{1}{Z_S^*} exp\left(-S(\boldsymbol{w}_{MP}) - \frac{1}{2}\Delta\boldsymbol{w}^T \boldsymbol{A} \Delta\boldsymbol{w}\right) \tag{8}$$

and the normalization term becomes

$$Z_S^* = e^{-S(\boldsymbol{w}_{MP})} (2\pi)^{W/2} |\boldsymbol{A}|^{-1/2} \tag{9}$$

in which W is the total number of connection weights. For a new input $\boldsymbol{x}^*$ and target t^*, the predicted output is $y(\boldsymbol{x}^*; \boldsymbol{w})$. The conditional probability density of t^* can be written as

$$p(t^* \mid \boldsymbol{x}^*, D) = \int p(t^* \mid \boldsymbol{x}^*, \boldsymbol{w}) p(\boldsymbol{w} \mid D) d\boldsymbol{w} \tag{10}$$

in which the integral is over the whole $\boldsymbol{w}$-space. In the Bayesian approach, the noise (i.e., the error between the target and predicted value) is assumed to be Gaussian with zero mean and variance $1/\theta$, which results in

$$p(t^* \mid \mathbf{x}^*, \mathbf{w}) \propto \exp\left(-\frac{\theta}{2}\{y(\mathbf{x}^*; \mathbf{w}) - t^*\}^2\right) \tag{11}$$

Substituting Eq. (8) and Eq. (11) into Eq. (10) leads to the relationship

$$p(t^* \mid \boldsymbol{x}^*, D) \propto \int exp\left(-\frac{\theta}{2}\{y(\boldsymbol{x}^*;\boldsymbol{w}) - t^*\}^2 - \frac{1}{2}\Delta\boldsymbol{w}^T \boldsymbol{A}\Delta\boldsymbol{w} \right) d\boldsymbol{w} \tag{12}$$

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

$$y(\boldsymbol{x}^*;\boldsymbol{w}) \approx y(\boldsymbol{x}^*;\boldsymbol{w}_{MP}) + \boldsymbol{g}^T\Delta\boldsymbol{w} \tag{13}$$

in which the partial derivative $\boldsymbol{g}$ is defined as

$$\boldsymbol{g} = \nabla_{\boldsymbol{w}} y(\boldsymbol{x}^*;\boldsymbol{w})|_{\boldsymbol{w}=\boldsymbol{w}_{MP}} \tag{14}$$

Substituting into Eq. (12) and evaluating the integral over $\boldsymbol{w}$ leads to the expression

$$p(t^* \mid \boldsymbol{x}^*, D) = \frac{1}{(2\pi\sigma_t^2)^{1/2}} exp\left(-\frac{\{y(\boldsymbol{x}^*;\boldsymbol{w}_{MP}) - t^*\}^2}{2\sigma_t^2} \right) \tag{15}$$

in which the standard deviation is given by

$$\sigma_t = \sqrt{\frac{1}{\theta} + \boldsymbol{g}^T \boldsymbol{A}^{-1}\boldsymbol{g}} \tag{16}$$

in which $\boldsymbol{A}$ is the data Hessian and $\boldsymbol{g}$ is the partial derivative of the output with respect to the weights.

The adjustment of the hyperparameters and the weight vector to their near optimal values is carried out iteratively during training, and therefore is less computationally intensive than the heuristic search procedures required in conventional back-propagation to find the optimal network for generalization.

Another limitation of the conventional back-propagation neural network algorithm is the lack of a method for analyzing the confidence intervals of the predictions. The Bayesian neural network approach yields the posterior distribution of the prediction and allows the calculation of the standard deviation on the network output, instead of just providing a single output. The standard deviation can be interpreted as an error bar on the mean value of the prediction. The standard deviation is computed from Eq. (16).

Some enhancements have been carried out to the original Bayesian neural network algorithm developed by Mackay [6]. The main features of this hybrid model [8, 9], called the evolutionary Baysesian back-propagation (EBBP) algorithm, are the use of the genetic algorithms (GA) search technique and a higher order search algorithm, the Levenberg-Marquardt algorithm.

3 Analyses using the hybrid Bayesian neural network

Three examples are presented that demonstrate the convergence and generalization capabilities of the hybrid Bayesian neural network approach.

3.1 Robot arm problem

A complicated validation problem commonly used to evaluate the non-linear mapping capabilities of neural network and other computer algorithms, the “robot arm” with two input and two output variables was considered. The task in the robot arm problem is to learn the

mapping from joint angles to position for an imaginary "robot arm". The actual relationship between inputs and outputs is as follows:

$$y_1 = 2.0 \cos(x_1) + 1.3 \cos(x_1 + x_2) + \text{noise} \quad (17)$$

$$y_2 = 2.0 \sin(x_1) + 1.3 \sin(x_1 + x_2) + \text{noise} \quad (18)$$

in which the noise is independent Gaussian noise with standard deviation of 0.05.

Comparisons were carried out with the test error reported by Mackay [6] and Neal [7]. The data sets of training and testing were obtained from Mackay's website (http:// wol.ra.phy.cam.ac.uk/mackay/Bayes_FAQ.htm.) Both these data sets contain 200 input-target pairs, which were randomly generated by picking x_1 uniformly from the ranges [1.932, -0.453] and [+0.453, +1.932], and x_2 uniformly from the range [0.534, 3.142]. The distribution for the two targets in y_1 and y_2 space is shown in Fig. 3. The sum squared errors of the testing patterns from the three different network models are summarized in Table 1. Overall, the hybrid Bayesian neural network (EBBP) gives comparable results to the other models.

Table 1: Average squared test errors for robot arm problem.

Neural network model	**Average squared test error**
Mackay [6]	0.00557
Neal [7]	0.00554
EBBP [8, 9]	0.00542

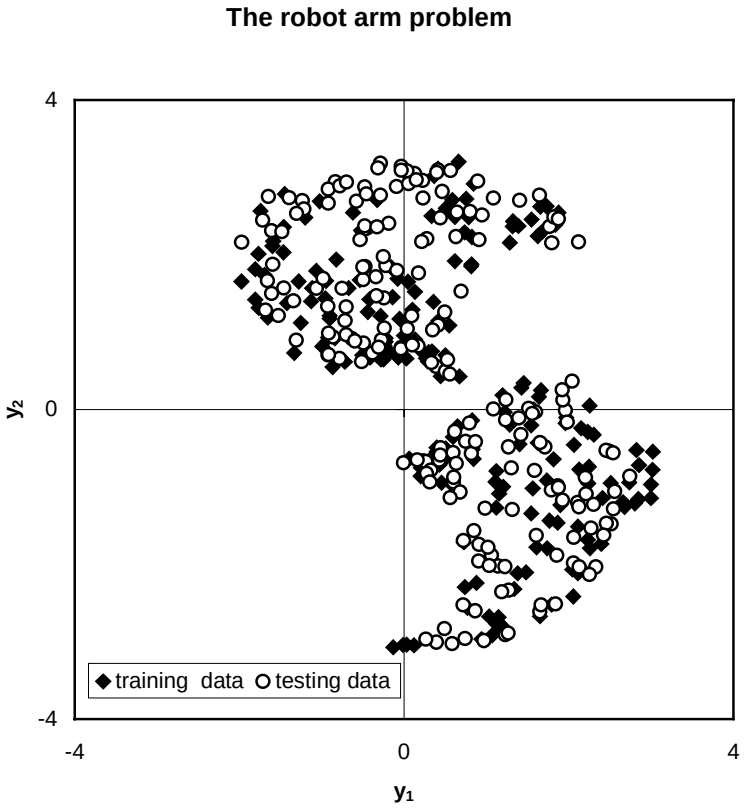

Figure 3: arget values in y_1 and y_2 space for robot arm problem.

3.2 Sine function with additive noise

A total of 21 data points were randomly generated from a simple sine function with additive Gaussian noise as shown in Fig. 4. This example serves to demonstrate the capabilities of the hybrid Bayesian neural network model (EBBP) to deal with noise and an uneven data density. The solid curve in Fig. 5 shows the function learned by the neural network and the dashed curves represent the associated error bars on the predictions. As expected, the error bars are higher for regions where the data density is low and the error bars are small where the data density is high.

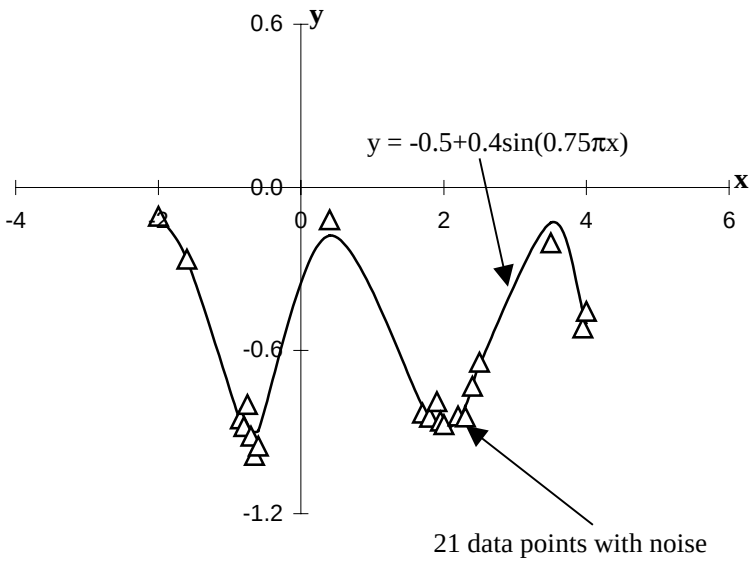

Figure 4: Plot of training data for the sine function.

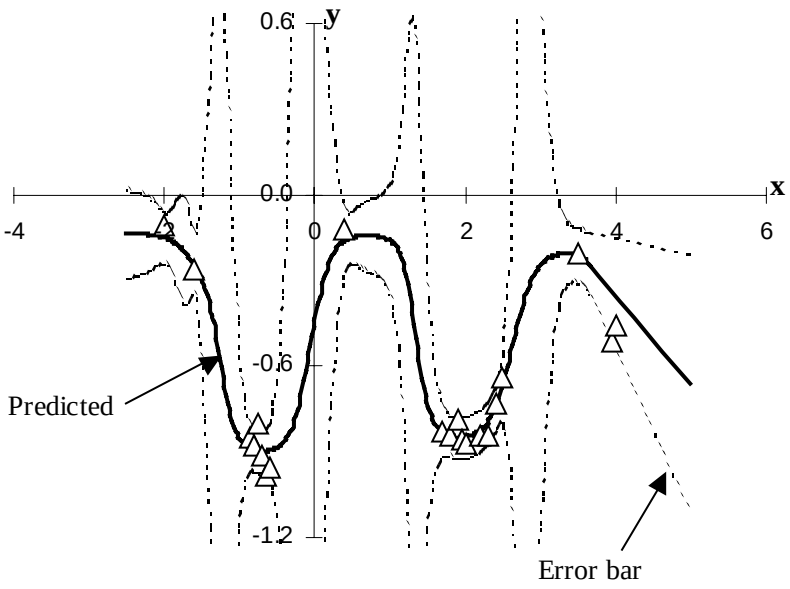

Figure 5: Predicted results (solid curve) and associated error bars (dashed curves).

3.3 Deep beam analysis

The design of deep beams is of considerable relevance in structural engineering and is commonly used in tall buildings and offshore structures. Deep beams have depths that are comparable to their span lengths. The behavior of deep reinforced concrete beams has been the subject of numerous experimental and analytical studies. Because of the significant number of factors (parameters) that affect the behavior of deep beams and the complexity of behavior of these beams when subjected to shear failure, to date, the understanding of deep beam behavior is still limited. Several design methods have been proposed, each based on differing assumptions and concepts. It is beyond the scope of this paper to discuss these conventional design methods. A study by Goh [10] demonstrated the feasibility of using the conventional back-propagation neural network to evaluate the ultimate shear strength of reinforced concrete deep beams. Recently, Sanad and Saka [11] carried out a similar study using an expanded database. Their study indicated that the predictions using the back-propagation neural network model were more accurate that those determined from conventional methods [12, 13, 14].

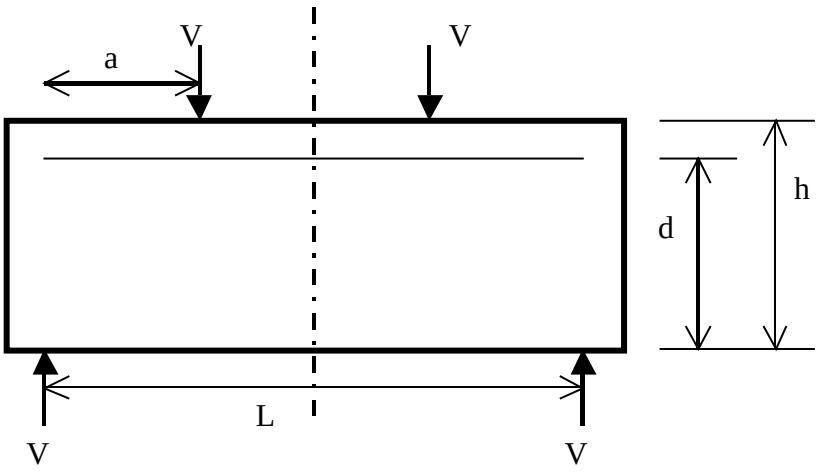

Fig. 6. Deep beam configuration (with reinforcement omitted)

In this paper, the hybrid Bayesian neural network was used to analyze this expanded database [11]. Training was carried out using 101 patterns and another ten patterns were used in the testing phase. For brevity, the data have been omitted in this paper. The basic parameters of the deep beam are shown in Fig. 6.

Following the work of Sanad and Saka [11], the nine input parameters considered in the neural network analysis are: the effective span/effective depth ratio (L/d), the effective depth/breadth ratio (d/b_w), the shear span/effective depth ratio (a/d), the cylinder compressive strength of concrete (f'_c), the yield strength of the longitudinal steel (f_{yh}), the yield strength of the transverse steel (f_{yv}), the reinforcement ratio of the horizontal tensile steel (ρ_h), the reinforcement ratio of the total horizontal steel (ρ_{ht}), and the reinforcement ratio of the transverse steel (ρ_v). The range of the input parameters is summarized in Table 2. The architecture of the neural network used in this problem is 9 input neurons, 7 hidden neurons and 1 output neuron representing the ultimate shear strength ($V/b_w d$).

Table 2. Range of data for deep beam analysis.

Parameter	Range of values
L/d	0.95-5.4
d/b_w	2.83-47
a/d	0.23-2.16
f'_c (MPa)	12.5-76
f_{yh} (MPa)	250-600
f_{yv}(MPa)	0-460
ρ_h (%)	0.05-1.94
ρ_{ht} (%)	0.14-2.95
ρ_v (%)	0-2.45

A plot of the neural network predicted versus the actual measured values for the training data patterns is shown in Fig. 7. Most of training data fall within the $\pm$ 10% error line. As shown in the plot of the testing data in Fig. 8, another advantage with the Bayesian inference is that every prediction of the test set data is associated with an error bar. These error bars are the standard deviations for the predictions based on the data distribution and inherent noise. The ratios of the predicted strength to the actual strength of the 10 testing patterns for the EBBP model are shown in Table 3 together with the neural network predictions by [11]. Also shown in Table 3 are the results based on the various conventional methods as reported by [11]. The specimen numbers are the reference numbers for the testing data patterns as reported in [11]. The results clearly demonstrate the accuracy of the neural network approaches over the conventional methods. Overall, the average ratio of the predicted strength to the actual strength for the EBBP model is slightly better than the neural network model used by [11].

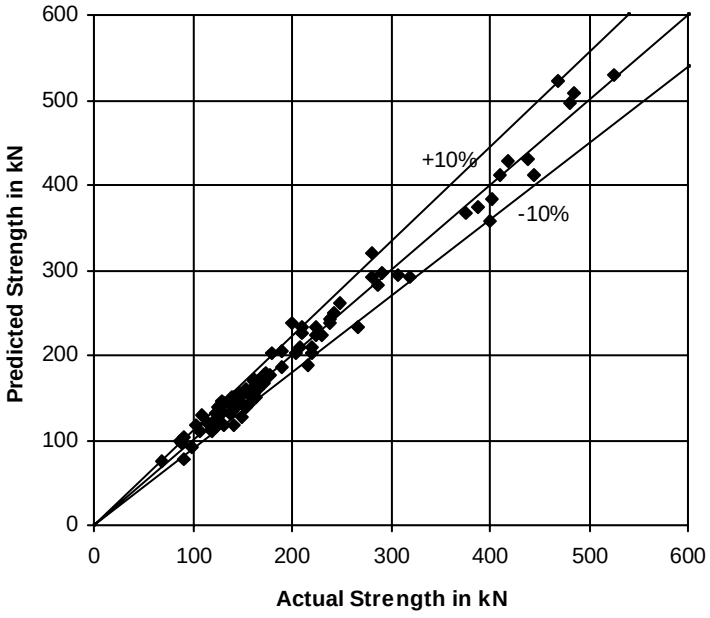

Fig. 7. Predicted versus measured values for training data.

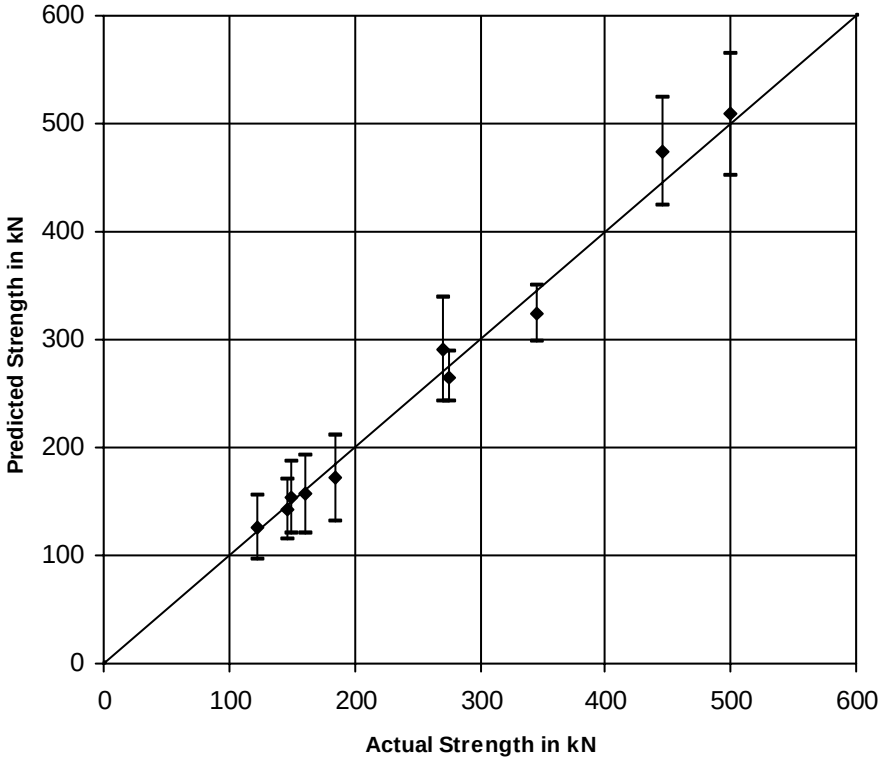

Fig. 8. Predicted versus measured values for testing data.

Table 3: Predicted shear strength results for deep beam testing data.

	Predicted strength/Actual strength				
Specimen number	ACI method [12]	Truss method [14]	Mau-Hsu method [13]	Neural network [11]	EBBP
3	0.33	0.89	0.97	0.91	0.97
7	0.52	0.84	0.94	0.98	0.93
23	0.47	1.04	1.10	1.00	1.02
36	0.49	1.26	0.95	1.11	1.03
45	0.32	1.03	0.97	1.02	0.98
80	0.44	0.76	1.67	1.01	0.94
86	0.67	2.14	2.04	1.21	0.97
93	0.36	0.86	1.48	1.01	1.08
95	0.23	1.62	1.30	1.00	1.06
99	0.31	1.91	1.45	1.07	1.02
Average	0.41	1.24	1.29	1.03	1.00

4 Summary

This paper demonstrates the robustness of the Bayesian neural network approach to model complicated nonlinear relationships. One distinct advantage of the Bayesian neural network approach is that the uncertainty of data can be indicated as an error bar based on the data distribution and intrinsic noise. These error bars will aid in giving confidence to the predicted values and the interpretation of the results. As demonstrated in the second example, in regions of low data density, the error bars of the predictions are higher than in regions of high data

density. The two other examples demonstrate the accuracy of the algorithm to model complicated multivariate relationships. As pointed out by [15], it is crucial to remember that while neural networks provide a powerful and efficient tool to model non-linear problems, neural networks do no exempt engineers from intimate and detailed knowledge of the data and problem domain.

5 References

1. Goh, A.T.C. "Seismic liquefaction potential assessed by neural networks", Journal of Geotechnical Engineering, ASCE, Vol. 120, No.9, 1994, pp.1467-1480.
2. Waszczyszyn, Z. "Some new results in application of back-propagation neural networks in structural and civil engineering", Advances in Computational Structures Technology.,Topping BHV (ed), Civil-Comp Press, Edinburgh, U.K., 1998, pp.173-187.
3. Jenkins, W.M. "Structural re-analysis by neural network", Advances in Engineering Computational Technology, Topping BHV (ed), Civil-Comp Press, Edinburgh, U.K., 1998, pp.229-237.
4. Ghaboussi, J., Garrett Jnr., J.H. and Wu, X. "Knowledge-based modeling of material behavior with neural networks", Journal of Engineering Mechanics, ASCE, Vol.117, No.1, 1991, pp.132-153.
5. Rumelhart, D.E., Hinton, G.E. and Williams RJ, "Learning internal representation by error propagation", In Parallel Distributed Processing, D.E. Rumelhart and J.L.McClelland (ed.). MIT Press: Cambridge, 1986; Vol.1, pp.318-362.
6. Mackay, D.J.C., "Bayesian methods for adaptive models", PhD Thesis, California Institute of Technology, 1991.
7. Neal, R.M., "Bayesian training of back-propagation networks by the hybrid Monte Carlo method", Technical Report CRG-TG-92-1, Department of Computer Science, University of Toronto, 1992.
8. Chua, C.G., and Goh, A.T.C. "Evolutionary Bayesian Back-propagation – An artificial neural network program", Geotechnical Research Report No. NTU/GT/2001-1, Nanyang Technological University: Singapore, 2001.
9. Chua, C.G. and Goh, A.T.C., "A hybrid Bayesian Back-Propagation Neural Network Approach to Multivariate Modeling", International Journal of Numerical and Analytical Methods in Geomechnics, 2003; Vol. 27, pp.651-667.
10. Goh, A.T.C., "Neural networks to predict shear strength of deep beams", ACI Structural Journal, Vol.92, No.1, 1995, pp.28-32.
11. Sanad, A. and Saka, M.P., "Prediction of ultimate shear strength of reinforced concrete deep beams using neural networks", Journal of Structural Engineering, ASCE, Vol.127, No.7, 2001, pp.818-828.
12. American Concrete Institute (ACI). "Building code requirements for reinforced concrete", ACI 318-95, 1995; Detriot.
13. Mau, S.T., and Hsu, C.T., "Formula for shear strength of deep beams", ACI Structural Journal, 1989; Vol.86, No.8, 1989, pp.516-523.
14. Siao, W.B., "Strut-and-tie model for shear behavior in deep beams and pile caps failing in diagonal splitting", ACI Structural Journal, Vol.90, No.4, 1993, pp.356-363.
15. Duda, R.O., Hart, P.E. and Stark, D.G., "Pattern classification", 2nd edition, John Wiley and Sons, New York, 2001.

