An Evaluation of Confidence Bound Estimation Methods for Neural Networks

Luren Yang[1], Tom Kavli[1], Mats Carlin[1], Sigmund Clausen[1], and Paul F. M. de Groot[2]
[1]SINTEF Electronics and Cybernetics
P. O. Box 124 Blindern, N-0316 Oslo, Norway
Phone: +47 22 06 73 00, Fax: +47 22 06 73 50
email: Luren.Yang@sintef.no
[2]de Groot - Bril Earth Sciences B.V.
Boulevard 1945 nr. 24, 7511 AE Enschede, The Netherlands
Phone: +31 53 4315155, Fax: +31 53 4315104
email: paul@dgb.nl

ABSTRACT: When artifical neural networks (ANN) are used in the prediction problems, it is usually desirable that some form of confidence bound is placed on the predicted value. Methods to estimate the confidence bound are available. However, these methods are valid under certain assumptions, which are rarely satisfied in practice. The behavior of the estimated confidence bound are not well understood when the assumptions are violated. We have designed some test functions to examine the behavior, and suggest how the estimated confidence bound can be corrected. The suggested method is used in the prediction of rock porosity values from seismic data for oil reservoir characterisation.

KEYWORDS: artificial neural network, prediction, confidence bound, evaluation, oil reservoir characterisation

INTRODUCTION

Artificial neural networks (ANN) are a class of non-linear models that have been successfully applied in many areas such as prediction, pattern recognition, classification and process control (Hertz *et. al.* 1991, Haykin 1994). The ANN models are inspired by brain architecture and are capable of learning from data. They are commonly used in problems where the underlying physical models are unknown. Our application is to predict rock porosity values from seismic data for oil reservoir characterisation. Compared to the output of other non-linear models, the output of a neural network is more dependent on the initialisation and the learning process. To measure the reliability of the neural network output is therefore very interesting. It is usually desirable that some form of confidence bound is placed on the output of a neural network. For our porosity prediction application to be successful in the oil industry, we need to estimate the confidence bounds. The problem of confidence bound estimation has been studied recently for prediction problems. A few methods are now available and some theories have been developed (Leonard *et. al.* 1992, Chryssolouris *et. al.* 1996, Shao *et. al.* 1997, Hwang and Ding 1997, de Veaux *et. al.* 1998).

The existing confidence bound estimation methods are asymptotically valid when the number of training points goes to infinite (Hwang and Ding 1997). It is assumed that the model errors are independent and normally distributed with zero means, there is no observation error (Chryssolouris *et. al.* 1996), and the neural network is trained to convergence (de Veaux *et. al.* 1998). In reality, these assumptions are rarely satisfied. The behaviours of the estimated confidence bounds are not well understood when these assumptions are violated. Further more, the existing confidence bound estimation methods have not been evaluated based on comparison. It is therefore difficult to choose the right method for a particular ANN application.

We designed some test functions to evaluate the performance of the confidence bound estimation methods, and to reveal the behaviours of the estimated confidence bounds in various situations. In addition to some simple, one and two-dimensional test functions, we designed a 10-dimensional test function, which represents a demanding, but still realistic prediction case. In this 10-dimensional function, some variables are correlated, and some are irrelevant to the output. The coverage of the confidence interval was used as a quantitative measure of the size of the confidence interval. Here, the coverage is the percent of targets that falls within the confidence interval. Also, we examined whether the estimated confidence bound reflected the distribution of the training

data. It is assumed that the confidence interval should be large in the area where the training data are less dense or the training data are extrapolated. Two types of neural networks were tested. They were networks with sigmoid activation functions and networks with radial basis activation functions.

The experimental results showed that the estimated confidence intervals are not always correct. With a large number of training points, however, the confidence interval will normally reflect the distribution of the training data. The size of the estimated confidence intervals depends on various conditions such as the level of observation noise and the training process, and sometimes needs to be corrected. We used porosity prediction case to show how such correction can be done.

CONFIDENCE BOUND ESTIMATION

It is commonly assumed that the ANN satisfies the nonlinear regression model

$$y = f(\mathbf{x}; \theta^*) + \varepsilon \tag{1}$$

where $\mathbf{x}$ is the inputs, y is one of the outputs, θ^* represents the true values of the set of parameters, and ε is the error associated with the function f in modeling the system. Let $\hat{\theta}$ be the least squares estimate of θ^* obtained by minimizing the error function

$$S(\theta) = \sum_{i=1}^{n} \left(y_i - f(\mathbf{x}_i; \theta)\right)^2 \tag{2}$$

for a training set $(y_i, \mathbf{x}_i)$, $i = 1, \cdots, n$. The predicted output of the ANN, for the input $\mathbf{x}_0$, is

$$\widehat{y_0} = f(\mathbf{x}_0; \hat{\theta}) \tag{3}$$

Assume that ε is independently and normally distributed with zero means. The $100(1-\alpha)$ percent confidence interval for the predicted value $\widehat{y_0}$ is $\widehat{y_0} \pm c$, where c is (Chryssolouris *et. al.* 1996)

$$c = t_{n-p}^{\alpha/2} s \left(1 + \mathbf{f}_0^T \left(\mathbf{F}^T \mathbf{F}\right)^{-1} \mathbf{f}_0\right)^{1/2} \tag{4}$$

Here $t_{n-p}^{\alpha/2}$ is the inverse of the Student t cumulative distribution function with $n-p$ degrees of freedom, evaluated at $\alpha/2$, p is the number of parameters (dimension of θ), and $s^2 = S(\hat{\theta})/(n-p)$. The vector $\mathbf{f}_0$ is given by

$$\mathbf{f}_0 = \left[\begin{array}{cccc} \frac{\partial f(\mathbf{x}_0;\theta^*)}{\partial \theta_1^*} & \frac{\partial f(\mathbf{x}_0;\theta^*)}{\partial \theta_2^*} & \cdots & \frac{\partial f(\mathbf{x}_0;\theta^*)}{\partial \theta_p^*} \end{array} \right]^T \tag{5}$$

$\mathbf{F}$ is the Jacobian matrix given by

$$\mathbf{F} = \left[\begin{array}{cccc} \frac{\partial f(\mathbf{x}_1;\hat{\theta})}{\partial \widehat{\theta_1}} & \frac{\partial f(\mathbf{x}_1;\hat{\theta})}{\partial \widehat{\theta_2}} & \cdots & \frac{\partial f(\mathbf{x}_1;\hat{\theta})}{\partial \widehat{\theta_p}} \\ \frac{\partial f(\mathbf{x}_2;\hat{\theta})}{\partial \widehat{\theta_1}} & \frac{\partial f(\mathbf{x}_2;\hat{\theta})}{\partial \widehat{\theta_2}} & \cdots & \frac{\partial f(\mathbf{x}_2;\hat{\theta})}{\partial \widehat{\theta_p}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f(\mathbf{x}_n;\hat{\theta})}{\partial \widehat{\theta_1}} & \frac{\partial f(\mathbf{x}_n;\hat{\theta})}{\partial \widehat{\theta_2}} & \cdots & \frac{\partial f(\mathbf{x}_n;\hat{\theta})}{\partial \widehat{\theta_p}} \end{array} \right] \tag{6}$$

Replacing θ^* in Eq. (5) by $\hat{\theta}$, we can estimate the confidence interval straightforwardly. Chryssolouris *et. al.* (1996) used this method to compute the confidence bounds for a neural network model of a manufacturing system. Hwang and Ding (1997) showed that this straightforward estimator is asymptotically valid, i.e.

$$P(y_0 \in \widehat{y_0} \pm c) \to 1 - \alpha \tag{7}$$

when number of training data points n goes to infinity. They also proposed that the size of the estimated confidence interval could be used to choose the number of units in the network.

Shao *et. al.* (1997) incorporated the influence of the distribution of the training data in the estimation of the confidence intervals. They used a wavelet-based method to estimate the density of the training data, and modified the confidence interval as

$$c_s = \frac{2c}{1 + \rho/\rho_{\max}} \tag{8}$$

where ρ is the density estimate and $\rho_{\max}$ is the maximum value of ρ.

De Veaux *et. al.* (1998) showed that the above method to compute the confidence interval works well when the number of the training data is large. However, when the training data set is small and the network is trained

to convergence, the $\mathbf{F}^T\mathbf{F}$ matrix can be nearly singular. In this case, the estimated confidence intervals are unreliable. Stopping the training prior to convergence, to avoid overfitting, reduces the effective number of parameters and can lead to confidence intervals that are too wide. The authors used the weight decay method (Haykin 1994) to prevent overfitting. That is to minimize the error function

$$S(\theta) + a\sum_{i=1}^{p}\theta_i^2 \tag{9}$$

instead of $S(\theta)$, in the training ($a > 0$). The confidence interval for networks trained by weight decay is (de Veaux *et. al.* 1998)

$$c = t_{n-p}^{\alpha/2} s\left(1 + \mathbf{f}_0^T\left(\mathbf{F}^T\mathbf{F} + a\mathbf{I}\right)^{-1}\mathbf{F}^T\mathbf{F}\left(\mathbf{F}^T\mathbf{F} + a\mathbf{I}\right)^{-1}\mathbf{f}_0\right)^{1/2} \tag{10}$$

For radial basis function neural networks, Leonard *et. al.* (1992) used a system called validity index network (VI-net) to compute the confidence intervals. The system is called a network because the computation of the confidence intervals and some other reliability measures are implemented as extra neural network units associated to the original neural network. The confidence intervals given by the VI-net, for input $\mathbf{x}_0$, is

$$c = \frac{\sum_{j=1}^{m} v_j(\mathbf{x}_0)c_j}{\sum_{j=1}^{m} v_j(\mathbf{x}_0)} \tag{11}$$

Here we assume that the network has only one hidden layer with m hidden units, and $v_j(\mathbf{x}_0)$ is the output of the j'th hidden unit. The value of c_j associated to the j'th hidden unit can be predetermined as

$$c_j = t_{n_j-1}^{\alpha/2} S_j\left(1 + \frac{1}{n_j}\right)^{1/2} \tag{12}$$

n_j is given by

$$n_j = \sum_{i=1}^{n} v_j(\mathbf{x}_i) \tag{13}$$

where $v_j(\mathbf{x}_i)$ is the output of the j'th hidden unit for training data $\mathbf{x}_i$. (Assuming that the training set has n elements.) S_j is given by

$$S_j^2 = \frac{\sum_{i=1}^{n} v_j(\mathbf{x}_i)\left(y_i - f(\mathbf{x}_i;\hat{\theta})\right)^2}{n_j - 1} \tag{14}$$

Here $\left(y_i - f(\mathbf{x}_i;\hat{\theta})\right)^2$ gives the difference between the desired output (target) y_i and the neural network output $f(\mathbf{x}_i;\hat{\theta})$ for training data $\mathbf{x}_i$.

In addition to the confidence interval estimation, the VI-net also computes the density of the training data, and an indicator showing whether the network is extrapolating the training data at a given data point. The density at data point $\mathbf{x}_0$ is

$$\rho(\mathbf{x}_0) = \frac{\sum_{j=1}^{m} v_j(\mathbf{x}_0)\rho_j}{\sum_{j=1}^{m} v_j(\mathbf{x}_0)} \tag{15}$$

where ρ_j can be predetermined according to the training data

$$\rho_j = \frac{\sum_{i=1}^{n} v_j(\mathbf{x}_i)}{n\left(\pi^{1/2}\sigma\right)^N} \tag{16}$$

in which N is the number of dimensions of the input data.

The extrapolation indicator is the maximum activation of the hidden units

$$\text{max-act} = \max_j\{v_j(\mathbf{x}_0)\} \tag{17}$$

If the test point $\mathbf{x}_0$ moves away from the training data, the value of the maximum activation will decrease. A small value of max-act can thus indicate extrapolation. We note that max-act and the density ρ are two related quantities. A small value of max-act may occur for interpolation as well. However, there are some differences between these two quantities (Leonard *et. al.* 1992).

POROSITY ESTIMATION

Seismic methods, which measure and interpret the response of earth subsurface due to a generated source wavefield, have played an important role in oil exploration. Using modern acquisition technology, it is possible to record a large amount of seismic reflection data, covering 2-D profiles or 3-D volumes. These data are then interpreted to obtain information of subsurface structures. In addition to the structural information, it is desirable to characterize the oil reservoir quantitatively. One of the quantitative measures is the rock porosity.

In porosity estimation, a moving window is applied to the seismic trace. Features, also called seismic attributes (Justice *et. al.* 1985), are computed from the data within a window, and then used as input to a neural network for the estimation of the porosity value. The neural network is trained by using the measurements obtained in some actual wells, or data from some simulated wells (de Groot *et. al.* 1996). In this application, it is important to know the reliability of the estimator, which can be assessed by the confidence bounds.

TEST FUNCTIONS

Some simple test functions, such as the sine function, were used to evaluate the confidence bound estimation methods. In addition to these simple test functions, we have designed a 10-dimensional test function. Seven of the ten input variables, $x_1, \cdots, x_7$, are

$$\begin{aligned} x_1 &= -t_1 \\ x_2 &= t_1^2 \\ x_3 &= 1/(t_1+2) \\ x_4 &= t_1 + t_2 \\ x_5 &= -t_2 \\ x_6 &= t_2^2 \\ x_7 &= 1/(t_2+2) \end{aligned}$$

where t_1 and t_2 are random variables. Their distributions will have effect on the test result. The other three input variables, x_8, x_9 and x_{10}, are normally distributed random variables with zero means. The output is $y = \sin(\pi t_1)$. In this test function, $x_1, \cdots, x_4$ and $x_4, \cdots, x_7$ are two correlated input groups, and $x_5, \cdots, x_{10}$ are irrelevant to the output. In all these test functions, we can add an observation noise to the input and the output of the training data, and to the input of the test data. The observation noise is normally distributed with a zero mean and a standard deviation up to 20% of the standard deviation of the data.

EXPERIMENTS

We first examined the shape of the estimated confidence intervals, especially how the estimated confidence interval behaved when the density of the training data varied and when the test data extrapolated the training data. We also examined the size of the estimated confidence interval. This was done by computing the coverage of the confidence interval and examining whether the obtained coverage was, in average, close to the desired coverage, and how large the variance of the coverage was.

We first used a sigmoid neural network to model a sine function $y = \sin x$. The network had one hidden layer of three nodes. The 30 training data points were unevenly distributed between $-\pi$ and π. The 200 test data points were evenly distributed between $-\pi$ and $\pi + \pi/2$. When $x > \pi$, the neural network extrapolated the training data. The confidence bounds were computed by Eq. (4) when a squared error function was used, and computed by Eq. (10) when a weigth decay error function was used. Typical results are shown in Figure 1. When the standard algorithm given by Eq. (4) was used, the size of the estimated confidence interval could clearly reflect the density of the training data. The confidence interval was large when the density of the training data was low, and was extremely large in the extrapolation area. When x was increased, the size of the confidence interval converged to a large value. In the case shown in Figure 1 (left), the size converged to about 600. When the weight decay algorithm was used, the size of the confidence interval could still reflect the density of the training data. Compared to the results given by the standard algorithm, the weight decay algorithm gave more smooth confidence bounds. We also used a radial basis neural network with three hidden nodes to simulate the sine function. The performance of the standard confidence bound algorithm, used together with a radial basis neural network, was very similar to that used with a sigmoid neural network (Figure 2 left). However, the confidence interval given by the VI-net less reflected the distribution of the training data (Figure 2 right).

Using 50 trials, we computed the average and the standard deviation of the coverage, to see whether the size of the confidence interval is correct. For the sine test function simulated by the sigmoid neural network, the results are shown in Figure 3 and Figure 4. We can see that a reduction of the size of the training set would increase the average coverage (Figure 3). When there was no observation error (Figure 3 left), the average

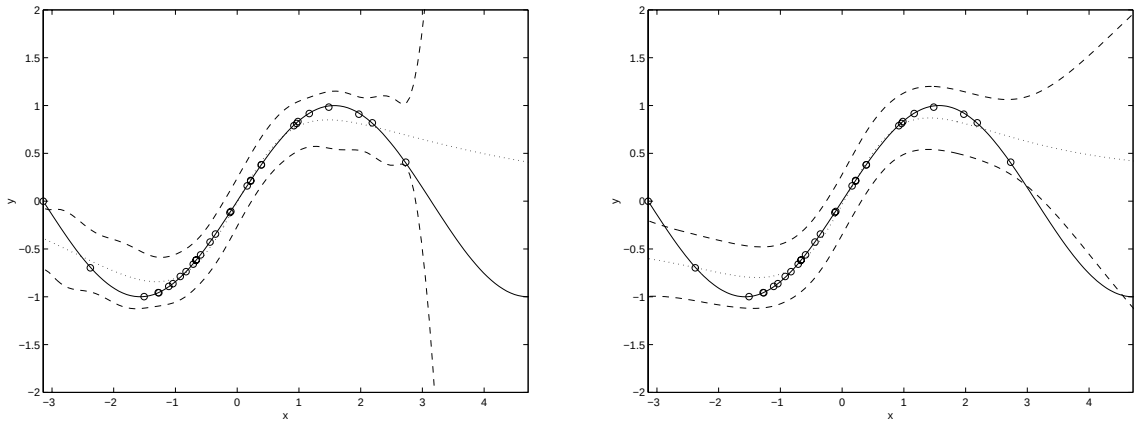

Figure 1: *A sine function simulated by a sigmoid neural network with one hidden layer of three nodes, trained by minimizing a squared error function (left) and by minimizing a weigth decay error function (right). No observation noise was added. Circles: training data points. Solid line: target function. Dotted line: neural network output. Dashed lines: 90% confidence intervals.*

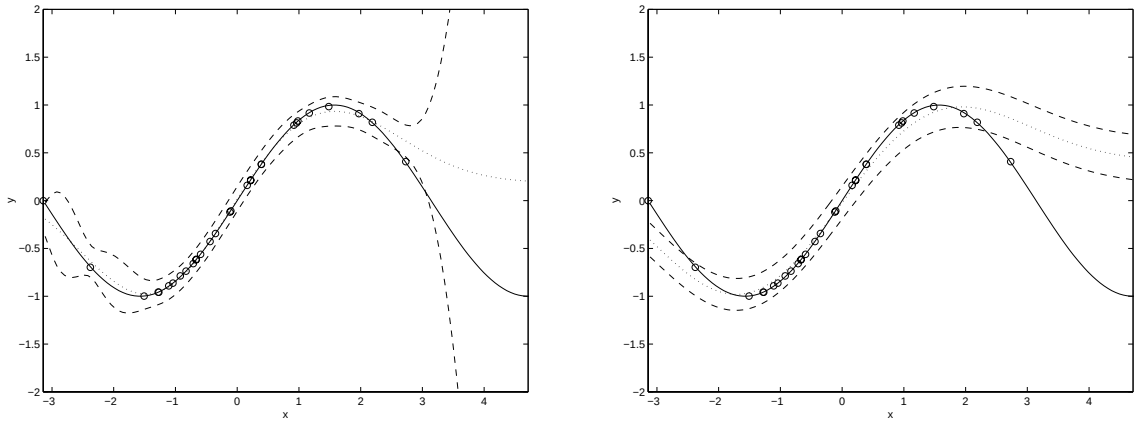

Figure 2: *A sine function simulated by a radial basis neural network with three hidden nodes. Confidence bounds esitmated by using Eq. (4) (left) and by using a VI-net (right). Circles: training data points. Solid line: target function. Dotted line: neural network output. Dashed lines: 90% confidence intervals.*

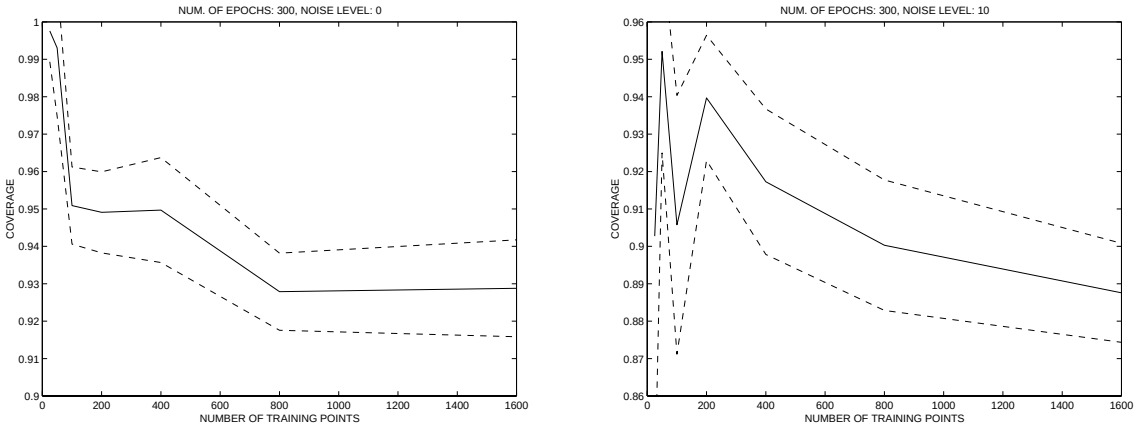

Figure 3: *Average coverage as a function of number of training points. The sine test function was used. Left: no observation noise. Right: 10% observation noise. Solid lines: average coverage computed from 50 trials. Dashed lines: average coverage plus/minus its standard deviation.*

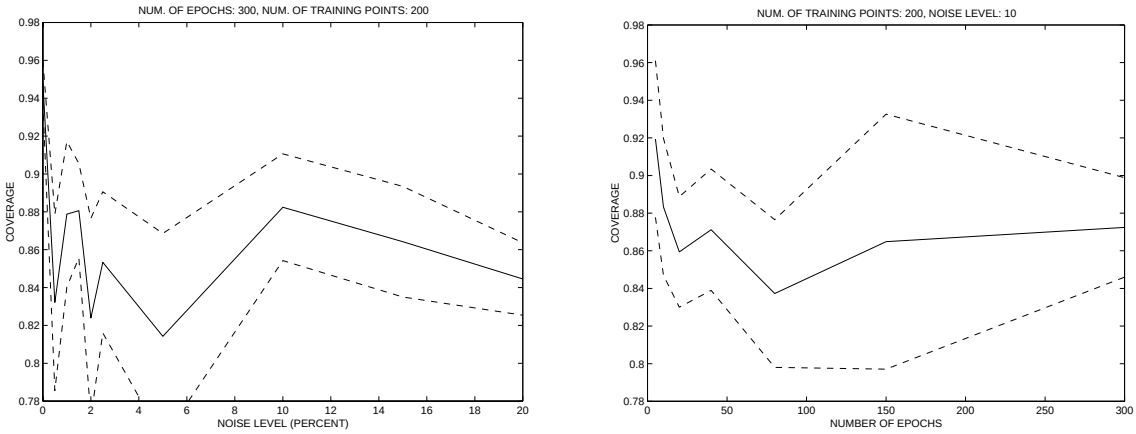

Figure 4: *Average coverage as a function of noise level (left) and as a function of number of training epochs (right). The sine test function was used. Solid lines: average coverage computed from 50 trials. Dashed lines: average coverage plus/minus its standard deviation.*

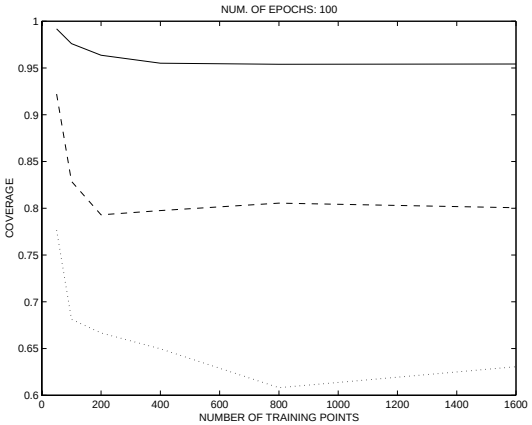

Figure 5: *Average coverage of the confidence interval, computed from 50 trials on a 10-dimensional test function. Desired coverage was 90%. Solid line:* $x_1, \cdots, x_3$ *were used as inputs. No observation noise. Dashed line:* $x_1, \cdots, x_{10}$ *were used as inputs. No observation noise. Dotted line:* $x_1, \cdots, x_{10}$ *were used as inputs. 10% observation noise.*

coverage was smaller than the desired coverage (90% in this case), and went towards the desired coverage as the number of training points increased. We added a normal distributed observation noise with zero mean and a standard deviation which was of 10% of the standard deviation of the data without noise (Figure 3 right). When an observation noise was added, the average coverage was significantly reduced and the variance of the coverage was increased. However, the level of the observation noise did not have a clear effect on the average coverage (Figure 4 left). The number of training epochs had neither a clear effect on the average coverage, except when there was an observation noise and when the number of epochs was very small. In this special case, the average coverage was increased (Figure 4 right). We have also observed that a reduction of training epochs would increase the variance of the coverage.

In addition to the sine function, we have used some other simple test functions and the 10-dimensional function to evaluate confidence bound estimation algorithms. In the most cases, the behavior of the confidence intervals was very similar to that we observed on the sine function. The 10-dimensional function was simulated by a sigmoid neural network with one hidden layer of three nodes. In the training data, t_1 was unevenly distributed between -1 and 1, and in the test data, t_1 was evenly distributed between -1 and 2. Thus, the neural network extrapolated the training data when $t_1 > 1$. The confidence bounds were estimated by the standard algorithm. The irrelevant inputs in the 10-dimensional function reduced the average coverage in a way similar to the observation noise, and the additional observation noise further reduced the coverage (Figure 5). In the presence of both irrelevant inputs and observation noise, the average coverage was only about 60%, which was much lower than the desired 90% coverage. This means that the average size of the confidence interval was just about 1/3 of the correct size, under the assumption that the error of the neural network output was normally distributed. The irrelevant inputs also increased the variance of the coverage. When the training data set was small, the irrelevant inputs often caused spikes on the confidence bounds (Figure 6 left). It seems that the irrelevant inputs were over-fitted by the training process and they were extrapolated at the spike places. Increasing the size of

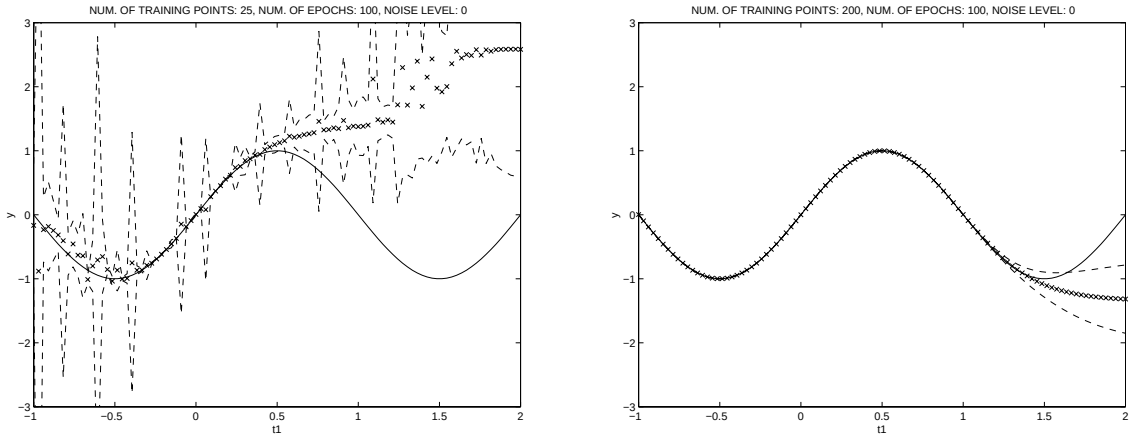

Figure 6: *Neural network simulation of a 10-dimensional function, using a training set of 25 points (left) and 200 points (right). Solid line: target function. Dashed lines: 90% confidence intervals. Crosses: neural network outputs.*

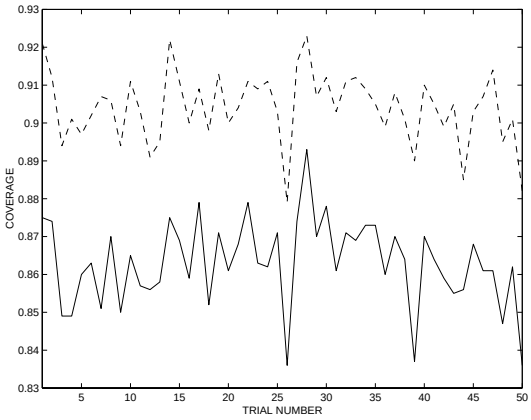

Figure 7: *Coverage of the confidence interval of the porosity estimates, obtained in 50 trials. Solid line: before correction. Dashed line: after correction.*

the training set would solve this problem (Figure 6 right).

We estimated the porosity values from synthetic seismic data by using a sigmoid neural network with one hidden layer of five nodes, and computed the confidence bounds by the standard algorithm. With carefully selected feature set, the neural network gave accurate porosity estimates and the coverage of the confidence interval was very close to the desired coverage (90%). When we added some irrelevant features to the feature set, the coverage was reduced to about 86%. We used a method to estimate the bias of the coverage and to correct the confidence bounds. We divided the data into three sets, each with 1000 data points. One of the data set was used as a training set, the others were a test set and an application set. The neural network was trained by using the training set and then applied to the test set. The average coverage $(1 - \alpha_1)$ was computed from 20 trials. When applying the neural network to the application set, the confidence bounds were corrected by

$$c_{\text{corrected}} = \frac{p(\alpha_0/2)}{p(\alpha_1/2)} c \tag{18}$$

where c is the size of confidence interval computed by the standard algorithm, function $p(\cdot)$ is the inverse of the normal cumulative distribution function, and $(1 - \alpha_0)$ is the desired coverage. After the correction, the obtained coverage was closer to the desired coverage, as shown in Figure 7.

DISCUSSION AND CONCLUSION

The existing confidence bound estimation methods are valid under certain assumptions, which are rarely satisfied in practice. In this work, we evaluated the confidence bound estimation methods in various situations, by changing the level of observation noise, the size of training set, the number of training epochs, and by adding

irrelevant inputs. The average coverage was used as a quantitative measure of the size of the confidence interval. This measure, however, only describes one aspect of the estimated confidence interval. In fact, the shape of the confidence interval is at least as important as the size of it. We do not have a quantitative measure to evaluate the shape of the interval. Instead, we inspect whether the estimated confidence interval reflects the density of the training data. We assume that the size of the confidence interval should be large in the area where the training data are less dense, or are extrapolated.

The VI-net did not perform well on our test cases, as the size of the estimated confidence interval less reflected the density of the training data. The training data density estimator and the extrapolation flag proposed by Leonard *et. al.* (1992) are not dependent on the VI-net, and can be used together with the standard confidence bound estimation algorithm. Compared to the standard algorithm, the weight decay algorithm given by de Veaux *et. al.* (1998) reduced the size of the confidence interval when the number of training epochs was small. In other cases, its performance was similar to the standard algorithm.

The standard confidence bound estimation algorithms normally gave satisfactory results. However, the size of the estimated confidence interval could be biased due to various reasons, and the shape of the confidence interval did not always reflect the density of the training data. It was observed that the estimated confidence intervals were normally larger than the desired coverage when there was no observation error. Increasing the number of training points would reduce the size of the estimated confidence intervals. The existence of irrelevant inputs and observation error would reduce the coverage of the confidence intervals. In such cases, the variance of the estimated confidence intervals could be very large when the number of training points was small. In the presence of observation error, early stop in training would increase the coverage of the confidence intervals. Irrelevant inputs would reduce the size of confidence interval in extrapolation areas, especially when the number of the training points was small. Without irrelevant inputs, the size of the confidence interval would normally approach an extremely large value when the training data were extrapolated. However, this value, depended on the neural network parameters obtained by the training process, was impossible to predict, and could be too small in certain cases.

In a practical prediction problem, it is desirable to compute the average coverage by using a test set. We have proposed a method to estimate the bias of the coverage, and applied the estimated bias to correct the confidence bounds. It is also desirable to have a large training set. Increasing the size of the training set will reduce the variance of the confidence bounds. There exist some data density measures and extrapolation indicators. These measures and indicators can be used as additional reliability assessment.

REFERENCES

Hertz, J., Krogh, A., Palmer, R. G., 1991, "Introduction to the Theory of Neural Computation", Addison-Wesley, Redwood City/CA, USA.

Haykin, S., 1994, "Neural Networks, a Comprehensive Foundation", Macmillan, New York/NY, USA.

Leonard, J. A., Kramer, M. A., Ungar, L. H., 1992, "A neural network architecture that computes its own reliability", Computers Chem. Engng. 16(9): 819–835.

Chryssolouris, G., Lee, M., Ramsey, A., 1996, "Confidence interval prediction for neural network models", IEEE Trans. Neural Networks 7(1): 229–232.

Shao, R., Martin, E. B., Zhang, J., Morris, A. J., 1997, "Confidence bounds for neural network representations", Computers Chem. Engng. 21(suppl.): S1173–S1178.

Hwang, J. T. G., Ding, A. A., 1997, "Prediction intervals for artificial neural networks", J. American Statistical Association 92(438): 748–757.

De Veaux, R. D., Schumi, J., Schweinsberg, J., Ungar, L. H., 1998, "Prediction intervals for neural networks via nonlinear regression", Technometrics 40(4): 273–282.

Justice, J. H., Hawkins, D. J., Wong, G., 1985, "Multidimensional attribute analysis and pattern recognition for seismic interpretation", Pattern Recognition 18(6): 391–407.

De Groot, P. F. M., Bril, A. H., Floris, F. J. T., Campbell, A. E., 1996, "Monte Carlo simulation of wells", Geophysics 61(3): 631–638.