Energy Clearing Price Prediction and Confidence Interval Estimation With Cascaded Neural Networks

Li Zhang, *Member, IEEE*, Peter B. Luh, *Fellow, IEEE*, and Krishnan Kasiviswanathan

***Abstract*—The energy market clearing prices (MCPs) in deregulated power markets are volatile. Good MCP prediction and its confidence interval estimation will help utilities and independent power producers submit effective bids with low risks. MCP prediction, however, is difficult since bidding strategies used by participants are complicated and various uncertainties interact in an intricate way. Furthermore, MCP predictors usually have a cascaded structure, as several key input factors need to be predicted first. Cascaded structures are widely used, however, they have not been adequately investigated. This paper analyzes the uncertainties involved in a cascaded neural-network (NN) structure for MCP prediction, and develops the prediction distribution under the Bayesian framework. A computationally efficient algorithm to evaluate the confidence intervals by using the memoryless Quasi-Newton method is also developed. Testing results on a classroom problem and on New England MCP prediction show that the method is computationally efficient and provides accurate prediction and confidence coverage. The scheme is generic, and can be applied to various networks, such as multilayer perceptrons and radial basis function networks.**

***Index Terms*—Bayesian inference, cascaded structure, confidence interval, market clearing price, neural networks, power systems, prediction, risk management.**

I. Introduction

THE deregulated power market is an auction market, and energy market clearing prices (MCPs) are volatile. High-quality MCP prediction and its confidence interval (CI) estimation can help utilities and independent power producers submit effective bids with low risks. However, good prediction and confidence interval estimation are difficult since bidding strategies used by participants are complicated, and various uncertainties interact in an intricate way.

Among the variety of prediction methods, neural networks (NNs) have been widely used [1]–[3]. They can approximate any continuous multivariate function to a desired degree of accuracy, provided that there are a sufficient number of hidden neurons [4], [5]. MCP prediction using NNs can generally be divided into three stages: training, prediction, and update. An NN is first trained by using historical data to approximate the input–output relationship, with associated measurement uncertainties in input (e.g., load) and output data (e.g., MCP). When training is finished and new input data are available, the NN predicts MCPs. In practice, certain key input factors (e.g., load) may not be available in real time and need to be predicted, with associated uncertainties in predicted values. The prediction system thus has a cascaded structure as shown in Fig. 1. Since predicted data are generally less accurate than measured data, an NN will be less accurate if predicted input data were used in training or update. This is the reason that when measured values of these factors become available at a later time, they are used together with actual MCPs in update or retraining as shown in Fig. 2. The uncertainties involved in a cascaded structure thus interact in a complicated way, affecting confidence interval estimation. Such cascaded structures are generic and widely exist in practice [1], [6].

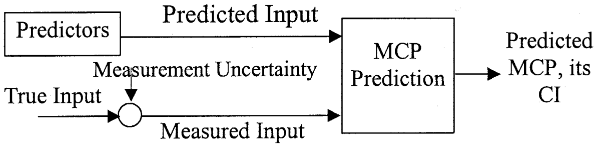

Fig. 1. Structure of cascaded MCP prediction.

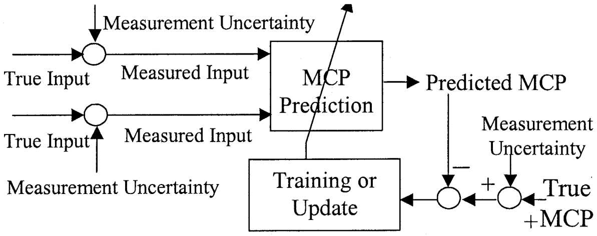

Fig. 2. Structure of MCP training and update.

Results on prediction distribution in a noncascaded structure have been reported in [13]. Measurement uncertainties[1] in input and output result in weight uncertainty through training and update. The prediction distribution is then developed using Bayesian techniques combining weight uncertainty and the measurement uncertainties in input and output. The derivation of the prediction distribution for a noncascaded structure is summarized in Section II.

In this paper, confidence interval estimation for a cascaded structure will be addressed. For a cascaded NN, predicted values are used for part of the input factors in the prediction stage while measured values are used in training and update, as shown in Figs. 1 and 2. How would prediction and measurement uncertainties affect prediction distribution for such a network? The

Manuscript received July 30, 2002. This work was supported in part by Select Energy, Inc., and by the National Science Foundation, under Grant ECS 9726577.

L. Zhang and P. B. Luh are with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269-2157 USA (e-mail: zhangL@engr.uconn.edu; Luh@engr.uconn.edu).

K. Kasiviswanathan is with Select Energy, Inc., Berlin, CT 06037 USA (e-mail: kasivk@selectenergy.com).

Digital Object Identifier 10.1109/TPWRS.2002.807062

[1] For simplicity, all data uncertainties in a noncascaded structure are called measurement uncertainties in the paper.

key is to examine the differences between a cascaded network and a noncascaded network. We discovered that under some general assumptions, a cascaded network can be regarded as a noncascaded network with an additional error term for each predicted input. The cascaded prediction distribution can thus be derived based on a noncascaded structure by using Bayesian techniques, and is approximated to be Gaussian as presented in Section III.

With the prediction distribution approximated as Gaussian, the variance of each output can be obtained from the corresponding diagonal element of the covariance, and the confidence intervals can then be obtained by deviating a certain number of standard deviations from the prediction. The calculation of the covariance involves the inverse Hessian matrix of the cost function with respect to weights, and this computation is expensive for practical MCP prediction. An important question is how to develop a computationally efficient algorithm to evaluate MCP confidence intervals. In this paper, a fast algorithm based on the memoryless Quasi-Newton method is presented in Section IV. Numerical testing results for a classroom problem and for New England MCP prediction presented in Section V demonstrate that our method is computationally efficient, and provides accurate confidence interval coverage. Although we have much less information than ISO New England does, our prediction is comparable in quality to ISO New England's.

II. Literature Review

A. Confidence Interval Estimation Methods

NN prediction methods have been briefly described in Section I. The methods to estimate confidence intervals can roughly be classified into three categories: resampling [7], [8], perturbation model [9]–[11], [17], [18], and Bayesian inference [4], [12]–[14] to be briefly described below. Variations of these methods were compared in [15] and [17].

The resampling method derives confidence intervals by randomly selecting data points with replacement from an original data set to form multiple sample data sets. The confidence interval of the mean can then be calculated from the means of the sample data sets. This method resamples output data, and cannot effectively consider input uncertainties. The perturbation model examines the effect on output if some parameters are perturbed. It uses Taylor series expansion to relate changes in the output to perturbed parameters and obtains output covariance matrix. Confidence intervals with input and weight uncertainties by using perturbation model were presented in [11]. Confidence intervals for a nonlinear regression using NN models were presented in [17] by using the least-square linear Taylor expansion (LS LTE) approach, and a tool to detect the ill-conditioning of NNs was provided. The perturbation model methods, however, are difficult for high-dimensional NNs with multiinput multioutput because it requires complicated covariance matrices.

Bayesian learning for NNs has attracted much attention recently. Starting with a prior distribution of weights for an NN, the method develops a posterior distribution of the weights from historical data. By linearizing the NN, the prediction distribution conditioned on a new input and weights can be derived and approximated as Gaussian for a noncascaded single-output structure [13]. A vector version will be briefly presented in the next subsection. Since the Gaussian approximation was based on simplification, general numerical techniques were developed to obtain the prediction distribution by integrating over the multidimensional weight space. Among these methods, the Markov chain Monte Carlo methods [4] and [13] obtain random sampling of points in the weight space and approximate the integration using a finite sum. The name "Markov chain" was given because a new sample depends on the previous one plus a random walk. The Metropolis algorithm [16] is one of the best Markov Chain Monte Carlo methods, and if a candidate sample leads to a reduction in the value of the posterior of weights, that sample will be rejected with a certain probability. For a practical NN with a high dimensional weight space, the proportion of rejected samples is high, rendering the method impractical [13, p. 1270]. For practical applications, such as MCP prediction, efficiency of an algorithm is important even though some assumptions have to be made. Gaussian approximations will be used in this paper, similar to the linearization step taken in many practical nonlinear system applications. A fast implementation for confidence intervals will then be developed in Section IV based on the Gaussian approximation. Numerical testing results show that the Gaussian approximation is satisfactory.

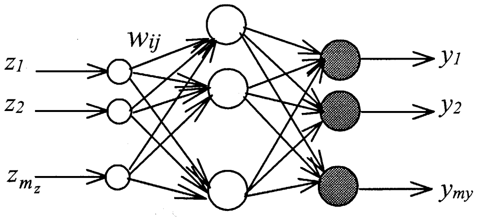

Fig. 3. Multilayer perceptron network.

B. Bayesian Prediction Distribution for a Noncascaded Structure

To summarize the results of [13], consider a multilayer perceptron (MLP) network with a single hidden layer, as shown in Fig. 3. The MLP can be trained by using a set of historical observations $D = \{Z, Y\}$, where $Z = \{z_1, \ldots, z_N\}$ is the input vector set with $m_z \times 1$ vector element z_n. The corresponding output vector set is $Y = \{y_1, \ldots, y_N\}$ with $m_y \times 1$ vector element y_n. The measurement uncertainties in output $\{\delta y_n\}$ are assumed to be independent, identically distributed (i.i.d.) zero-mean Gaussian with covariance matrix I/β. The measurement uncertainties in input $\{\delta x_n^m\}$ are also assumed to be i.i.d. zero-mean Gaussian with covariance matrix Σ_x. The Gaussian assumptions for measurement uncertainties are acceptable for most cases. A measured input z_n is then equal to the true input x_n plus the measurement uncertainty as follows:

$$z_n = x_n + \delta x_n^m. \tag{1}$$

The network is trained by using a cost function to maximize the posterior distribution of weights to be presented later. Let the input–output relationship after training be expressed as $\hat{y}(z_n, w)$, where $n \in [1, N]$ and w are the $m_w \times 1$ weight vector of the network. The prediction distribution $p(y^* | x^*, D)$ given a new true input x^* has been approximated to be Gaussian [4]. If only a measured noisy input z^* is available, the prediction distribution can be obtained by using the Bayes rule

$$p\left(y^* \left| z^*, D\right.\right) = \int p\left(y^* \left| x^*, D\right.\right) p\left(x^* \left| z^*\right.\right) dx^*. \qquad (2)$$

To analyze the right-hand side of (2), it is noted that $p(x^*|z^*)$ can be obtained from (1) following the Gaussian assumption:

$$p(x^*|z^*) = \frac{1}{\left|2\pi \, \Sigma_x\right|^{1/2}} \exp\left(\frac{-1}{2}\left(x^* - z^*\right)^T \Sigma_x^{-1} \left(x^* - z^*\right)\right). \qquad (3)$$

In addition, assume that the uncertainty in the measured input is small, and $\hat{y}(x, w)$ in the $p(y^*|x^*, D)$ can be linearized around z. The prediction distribution (2) can then be obtained as follows by using the linearized model, the Bayes rule, and (3):

$$\begin{aligned} p\left(y^*|z^*, D\right) &= \frac{1}{Z_x}\int \exp\left(\frac{-1}{2}\left\{\left(y^* - \hat{y}\left(z^*, w\right)\right)^T B\left(y^* - \hat{y}\left(z^*, w\right)\right)\right\}\right) \\ &\quad \cdot \exp\left(\frac{-1}{2}\sum_{n=1}^{N}\left\{\left(y^n - \hat{y}(z_n, w)\right)^T B\left(y^n - \hat{y}(z_n, w)\right)\right\}\right. \\ &\quad \left. - \frac{\alpha}{2}\sum_{i=1}^{m_w} w_i^2\right) dw \end{aligned} \qquad (4)$$

where

$$B^{-1} = I/\beta + h\,\Sigma_x\,h^T.$$

In the above equation, Z_x is a normalizing constant; α is a parameter controlling the prior distribution of weights; and h is the gradient of $\hat{y}(x, w)$ with respect to the measured input (i.e., a Jacobian matrix with dimension $m_y \times m_z$)

$$h \equiv \nabla_x \, \hat{y}(x, w_{MP})\big|_{x=z}. \qquad (5)$$

To train the network, it is observed that the first exponential term in (4) is contributed by the new input z^*, whereas the second exponential term is from historical data D, and can be shown to be the posterior distribution of the weights $p(w|D)$. The optimized weight vector w_{MP} is obtained by maximizing the posterior distribution $p(w|D)$, or equivalently minimizing the cost function below in the NN training stage [4]

$$\begin{aligned} S(w) = \tfrac{1}{2}\sum_{n=1}^{N}&\left\{\left(y^n - \hat{y}(z_n, w)\right)^T B\left(y^n - \hat{y}(z_n, w)\right)\right\} \\ &+ \frac{\alpha}{2}\sum_{i=1}^{m_w} w_i^2. \end{aligned} \qquad (6)$$

The cost function $S(w)$ is the usual sum of squares with an additional "weight-decay regularization term" $(\alpha/2)\sum_{i=1}^{m_w} w_i^2$, which is from the Gaussian prior assumption for the weights, and could reduce the sensitivity of model prediction with respect to input uncertainty [4]. With noisy input and the use of weight-decay regularization term, the estimator is biased and the accuracy of the confidence intervals is affected [13], [17]. However, the effect of weight decay becomes minor when the number of data sets is large, as the predictor becomes asymptotically unbiased [17], [18].

With E_D defined as

$$E_D \equiv \tfrac{1}{2}\sum_{n=1}^{N}\left\{\left(y^n - \hat{y}(z_n, w)\right)^T B\left(y^n - \hat{y}(z_n, w)\right)\right\}$$

the $m_w \times m_w$ Hessian matrix of $S(w)$ with respect to the weights can be expressed as

$$A \equiv \nabla^2 S\big|_{w=w_{MP}} = \nabla^2 E_D^{MP} + \alpha I.$$

To further derive the prediction distribution, $S(w)$ is approximated by its second-order Taylor series expansion. Noting that the first-order term is zero at the optimized weight w_{MP}, $S(w)$ can be approximated as follows:

$$S(w) \cong S(w_{MP}) + \tfrac{1}{2}(w - w_{MP})^T A (w - w_{MP}). \qquad (7)$$

Substituting (7) into (4) leads to

$$\begin{aligned} p\left(y^*|z^*, D\right) &= \frac{1}{Z_x}\int \exp\left(-\frac{1}{2}\left\{\left(y^* - \hat{y}\left(z^*, w\right)\right)^T B\left(y^* - \hat{y}\left(z^*, w\right)\right)\right\}\right) \\ &\quad \cdot \exp\left(-\tfrac{1}{2}\left(w - w_{MP}\right)^T A\left(w - w_{MP}\right)\right) dw. \end{aligned} \qquad (8)$$

To evaluate the integration over w in (8), $\hat{y}(z^*, w)$ can be linearized around the optimized weight w_{MP} (as opposed to around input z as before), since the change in weights per iteration is small. The prediction distribution can then be expressed as

$$\begin{aligned} p\left(y|z^*, D\right) = \frac{1}{\left|2\pi\hat{\Sigma}_y\right|^{1/2}} \exp&\left(\frac{-1}{2}\left(y - \hat{y}\left(z^*, w_{MP}\right)\right)^T\right. \\ &\left. \cdot \hat{\Sigma}_y^{-1}\left(y - \hat{y}\left(z^*, w_{MP}\right)\right)\right) \end{aligned} \qquad (9)$$

which is Gaussian with the following covariance matrix $\hat{\Sigma}_y$:

$$\hat{\Sigma}_y = \frac{1}{\beta}I + gA^{-1}g^T + h\Sigma_x h^T \qquad (10)$$

where g is the partial derivative of $\hat{y}(z^*, w)$ with respect to weights (i.e., a Jacobian matrix with dimension $m_y \times m_w$)

$$g \equiv \nabla_w \hat{y}(z, w)\big|_{w=w_{MP}}.$$

There are three terms in the prediction covariance (10). The first term is from output measurement uncertainty, the second term is contributed by the weight uncertainty, and the third term is propagated from the input uncertainty. These terms are additive in view of the linearization of the network.

III. Bayesian Inference for Cascaded Prediction

With the noncascaded prediction distribution expressed in (9), we shall next develop the relationship between a cascaded structure and a noncascaded one. The cascaded prediction distribution is then derived and the covariance matrix computed.

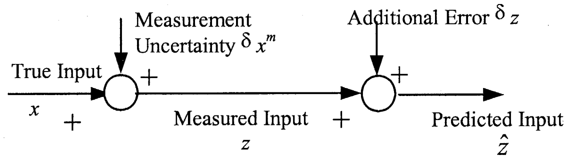

Fig. 4. Predicted input expressed as a measured input plus an additional error term for a cascaded structure.

A. Relationship Between Cascaded and Noncascaded Structures

To develop the relationship between the two prediction structures, the key is to examine the difference in their inputs. In a noncascaded structure, measured input is used in prediction as expressed in (1). In a cascaded structure, predicted values are used in prediction. Assuming that the prediction uncertainty δx^e is i.i.d. zero mean Gaussian and independent of the measurement uncertainty δx^m for the same input factor, the predicted input $\hat{z}$ is given by

$$\hat{z} = x + \delta\, x^e. \tag{11}$$

With (1) and (11), the difference between the predicted input and the measured input δx^e given by

$$\delta z \equiv \hat{z} - z = \delta x^e - \delta x^m$$

is also i.i.d. zero mean Gaussian. Let its covariance matrix be denoted as P. The predicted input can then be expressed as

$$\hat{z} = x + \delta\, x^e = x + \delta x^m + \delta z = z + \delta z. \tag{12}$$

A cascaded structure can thus be regarded as a noncascaded structure with the predicted input expressed as the measured input plus an additional error term as shown in Fig. 4. The cascaded prediction distribution will next be derived from a noncascaded prediction distribution by using Bayesian techniques.

B. Bayesian Prediction for a Cascaded Structure

For a new predicted input $\hat{z}^*$ in a cascaded prediction, the prediction distribution of the output can be expressed by using the Bayes rule as follows:

$$p(y|\hat{z}^*, D) = \int p(y|z^*, D) p(z^*|\hat{z}^*)\, dz^* \tag{13}$$

where $p(y|z^*, D)$ is the noncascaded prediction distribution in (9). To analyze the right-hand side of (13), it is noted that $p(z^*|\hat{z}^*)$ can be obtained from (12) following the Gaussian assumption:

$$p(z^*|\hat{z}^*) = |2\pi P|^{-1/2} \exp\left(\frac{-1}{2} (z^* - \hat{z}^*)^T P^{-1} (z^* - \hat{z}^*) \right). \tag{14}$$

In addition, assume that the uncertainty in the predicted input is small, $\hat{y}(z^*, w_{MP})$ in $p(y|z^*, D)$ can therefore be linearized by using the Taylor series expansion around $\hat{z}^*$, that is

$$\hat{y}(z^*, w_{MP}) \cong \hat{y}(\hat{z}^*, w_{MP}) + h(z^* - \hat{z}^*) \tag{15}$$

where

$$h \equiv \nabla_{z^*} \hat{y}(z^*, w_{MP})\big|_{z^* = \hat{z}^*}. \tag{16}$$

The prediction distribution (13) can then be obtained by using the linearized model, the Bayes rule, and (14) as follows:

$$\begin{aligned} &p(y|\hat{z}^*, D)\\ &\quad= \exp\left(\frac{-1}{2} \{y - \hat{y}(\hat{z}^*, w_{MP})\}^T \hat{\Sigma}_y^{-1} \{y - \hat{y}(\hat{z}^*, w_{MP})\} \right)\\ &\qquad \cdot \int \exp\Big(-\tfrac{1}{2}\, \delta z^T \left\{ h^T \hat{\Sigma}_y^{-1} h + P^{-1} \right\} \delta z\\ &\qquad\qquad - \{y - \hat{y}(\hat{z}^*, w_{MP})\}^T \hat{\Sigma}_y^{-1} h \delta z \Big)\ dz^*. \end{aligned} \tag{17}$$

The integral in (17) can be calculated by using the following Gaussian integral property [4, App. B]:

$$\begin{aligned} &\int \exp\left(-\tfrac{1}{2}\, w^T A w + h^T w \right)\, dw\\ &\qquad = (2\pi)^{m_w/2} |A|^{-(1/2)} \exp\left(\tfrac{1}{2}\, h^T A^{-1} h \right) \end{aligned} \tag{18}$$

where w is a m_w-dimensional vector. Applying (18) to (17) leads to

$$\begin{aligned} &p(y|\hat{z}^*, D)\\ &\quad = (2\pi)^{m_w/2} \left| h^T \hat{\Sigma}_y^{-1} h + P^{-1} \right|^{-(1/2)}\\ &\qquad \cdot \exp\Bigg(-\tfrac{1}{2} \{y - y(\hat{z}^*, w_{MP})\}^T\\ &\qquad\qquad \cdot \left[\hat{\Sigma}_y^{-1} - \hat{\Sigma}_y^{-1} h \left(h^T \hat{\Sigma}_y^{-1} h + P^{-1} \right)^{-1} h^T \hat{\Sigma}_y^{-1^T} \right]\\ &\qquad\qquad \cdot \{y - y(\hat{z}^*, w_{MP})\} \Bigg). \end{aligned} \tag{19}$$

It is noted that (19) is close to Gaussian when the terms can be organized. With the following obtained from (10) and matrix operation:

$$\begin{aligned} \Sigma_y &\equiv \left(\hat{\Sigma}_y^{-1} - \hat{\Sigma}_y^{-1} h \left(h^T \hat{\Sigma}_y^{-1} h + P^{-1} \right)^{-1} h^T \left(\hat{\Sigma}_y^{-1} \right)^T \right)^{-1}\\ &= \left[h \left(I + P h^T \hat{\Sigma}_y^{-1} h \right) h^T \right] \left(\hat{\Sigma}_y^{-1} h h^T \right)^{-1}\\ &= \frac{1}{\beta} I + g A^{-1} g^T + h \left(\Sigma_x + P \right) h^T \end{aligned} \tag{20}$$

the cascaded prediction distribution can then be expressed as

$$\begin{aligned} p(y|\hat{z}^*, D) = \frac{1}{|2\pi \Sigma_y|^{1/2}} &\exp\left(\frac{-1}{2} \{y - \hat{y}(\hat{z}^*, w_{MP})\}^T \right.\\ &\left. \cdot\, \Sigma_y^{-1} \{y - \hat{y}(\hat{z}^*, w_{MP})\} \right) \end{aligned} \tag{21}$$

which is Gaussian with the covariance matrix Σ_y.

As shown in (20), the covariance matrix of the cascaded prediction distribution is similar to the covariance of a noncascaded structure in (10), with one extra term hPh^T. This is intuitively plausible since the predicted input has been decomposed into the measured input related to Σ_z plus an independent Gaussian noise related to P in (12). The contribution of output measurement noise I/β remains the same, and so does the contribution

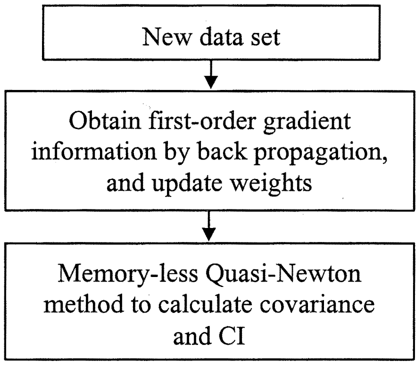
New data set
Obtain first-order gradient information by back propagation, and update weights
Memory-less Quasi-Newton method to calculate covariance and CI

Fig. 5. Flowchart of the CI algorithm.

of weight uncertainty $gA^{-1}g^T$ since predicted input is not used to update the weights. The covariance matrix in (20) is the interaction of the four uncertainties and the structure of the network, and can be obtained for a differentiable NN. Our method is therefore generic in terms of model independence.

IV. Computationally Efficient Algorithm

A utility company needs to consider predicted prices and other market information to prepare bids within a short time window before bid submission deadline. The allocated time for price prediction is limited (e.g., 10 min). It is therefore necessary to implement the method in an efficient way for it to be used on a daily basis. Instead of using the time-consuming Markov chain Monte Carlo algorithms to evaluate an integration such as (4) over a high dimensional weight space, the prediction distribution and covariance in (20) and (21) will be based on the Gaussian approximation in conjunction with the memoryless Quasi-Newton method to be explained next.

Quasi-Newton methods approximate the Newton direction to solve an optimization, while avoiding second derivative calculations associated with the Newton's method [19]–[21]. The methods approximate inverse Hessian of the cost function (6), A^{-1}, they could thus be used to help calculate the second term of the prediction covariance (20). Memoryless Quasi-Newton method [19, p. 158] further reduces computational requirements by avoiding the calculation of inverse Hessian matrix $A^{-1}$and requiring no matrix manipulation. With this method, the second term in (20) can be iteratively approximated as

$$A^{-1^{(k+1)}} g^{(k+1)^T} = g^{(k+1)^T} + \frac{p^{(k)^T} g^{(k+1)^T}}{p^{(k)^T} q^{(k)}} p^{(k)} - \frac{q^{(k)^T} g^{(k+1)^T}}{q^{(k)^T} q^{(k)}} q^{(k)} \quad (22)$$

where k is the iteration number and

$$p^{(k)} \equiv w^{(k+1)} - w^{(k)}, \qquad q^{(k)} \equiv e^{(k+1)} - e^{(k)}$$

and

$$e^{(k)} \equiv \& \frac{\partial S\left(w^{(k)}\right)}{\partial w}.$$

The gradient information can be easily obtained by using back-propagation. The flowchart of the method is shown in Fig. 5.

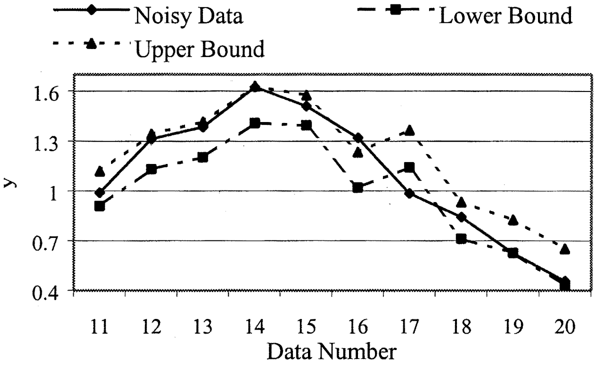

Fig. 6. CI with a cascaded structure.

V. Numerical Results

The NN prediction and confidence interval estimation method has been implemented in C++ on a Pentium III 500-MHz personal computer (PC). Two examples are presented below. The first classroom-type problem shows that the cascaded structure provides accurate confidence intervals. The second New England MCP prediction problem shows that our method provides accurate confidence intervals, and the prediction is comparable to ISO New England's prediction.

Example 1: A three-layer MLP network with neuron numbers of 1–5–1 was used to approximate the following nonlinear function:

$$y = \frac{2}{1 + \exp\left(\left(\frac{1}{1+e^{(-0.1-0.5x)}}\right) + \left(\frac{1}{1+e^{(0.5+0.4x)}}\right) - 1\right)} + 0.5\sin(0.5x)$$

which was composed of sigmoidal activation functions plus a sinusoidal term. To simulate the cascaded prediction structure, there were measurement uncertainties in training and predicted uncertainties in prediction for input; and measurement uncertainties for output. Specifically, 20 noisy data sets $\{z, \tilde{y}\}$ were randomly generated for training with $z = x + 0.1\varepsilon_1$, $x = -11 + i$, $\tilde{y} = y + 0.05\varepsilon_2$, where $i \in [1, 2, \ldots, 20]$, and $\varepsilon_1, \varepsilon_2 \in N(0, 1)$. In prediction, another 20 random data sets $\{\hat{z}, \tilde{y}\}$ were generated with $\hat{z} = x + 0.6\varepsilon_1$, where the input standard deviation was changed to 0.6.

Two cases were tested in this example. In the first case, confidence intervals derived from (20) were used, and input in prediction was known to have a standard deviation 0.6. One-sigma confidence intervals covered 70% with six points outside intervals among the 20 points, and this coverage was close to 68% Gaussian coverage. The noisy data, confidence interval lower bound, and upper bound for data points from 11 to 20 are shown in Fig. 6, where four points (points 14, 16, 17, and 19) are outside the confidence intervals. It can be seen that the cascaded structure provides accurate confidence intervals. In the second case, the noncascaded confidence intervals from (10) were used, with the assumption that the inputs for training and prediction had a same standard deviation 0.1. The one-sigma confidence intervals only covered 40% in this case. The prediction results for data points from 11 to 20 are shown in Fig. 7, where seven

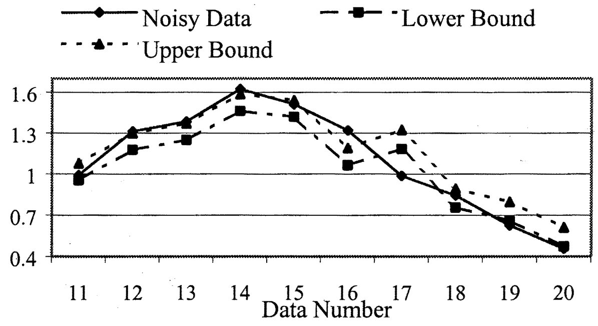

Fig. 7. CI with a noncascaded structure.

TABLE I
ONPEAK MCPS PREDICTION AND CI ESTIMATION

Date	July	Aug.	Sep.	Oct.	Nov	Overall
MAE ($)	2.81	7.01	4.68	4.14	4.05	4.54
MAPE (%)	6.27	13.4	9.35	7.28	7.7	8.80
MAPE of ISO-NE (%)	10.85	9.68	12.95	7.05	8.2	9.73
# of Days outside CI	7	17	8	11	8	51
Coverage (%)	77	42	73	64	73	66.6

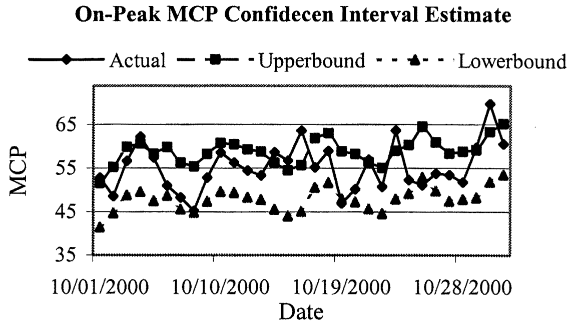

Fig. 8. On-peak MCP confidence intervals for October 2000.

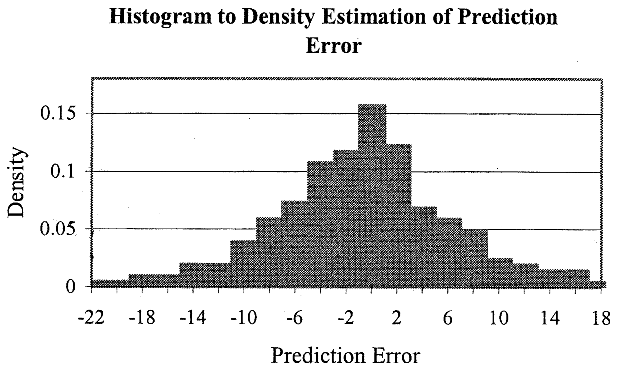

Fig. 9. Histogram-to-density estimation of on-peak MCP prediction error.

points (points 12, 13, 14, 16, 17, 19, 20) are outside the confidence intervals. This case shows that the noncascaded confidence interval scheme should not be used in the cascaded prediction structure and cascaded prediction structure provides accurate confidence interval for a classroom problem.

Example 2: In this example, a cascaded MLP NN model was used to predict day-head onpeak New England MCPs (average MCPs from 7 AM to 11 PM) and its confidence intervals. A three-layer MLP was constructed for MCP prediction.

The New England energy market follows a single-settlement system. After receiving generator offers, ISO runs a unit commitment program and selects generators based on merit order. Prices are not financially binding until after the fact, also generators and load sell and buy at real-time price. The real-time market activities can affect MCPs, thus, the day-ahead MCP prediction is difficult under this market structure.

ISO New England's predicted load and actual load, predicted MCP, and actual MCP, etc., were collected from May 1, 1999 to the end of November 2000 from ISO New England's website (www.iso-ne.com). More than 50 input factors were used for MCP prediction, including predicted load, historical load, historical MCPs, projected surplus (total available capacity minus required capacity), etc. The NN thus had a total number of weights over 500 and was trained from May 1, 1999 to June 30, 2000, then predicted from July 1, 2000 to November 30, 2000.

The onpeak MCP prediction results are summarized in Table I. The mean absolute percentage error (MAPE) of the prediction is 8.8%. One-sigma confidence intervals have a coverage of 66.6%, which is close to 68% of Gaussian coverage. The MAPE for ISO New England's prediction for the same period was 9.73%. While having less information than what ISO New England had, our NN prediction is comparable in quality to ISO New England's prediction. To elaborate the prediction results, the confidence intervals for October 2000 with upper bounds, lower bounds, and actual MCPs are shown in Fig. 8. There are 11 days outside the confidence intervals with 64% coverage.

The confidence intervals for cascaded prediction are overall satisfactory and the Gaussian assumption of the prediction distribution is overall acceptable, as shown in Fig. 9 the histogram to the density estimation of onpeak MCP prediction error. However, Table I also shows that the confidence intervals are not always consistent with the 68% Gaussian coverage for each month, and the coverage in August is only 42%. The inconsistency might be caused by abnormal behaviors in the markets or by the inaccuracy of Gaussian approximation [18].

Since MLPs are sensitive to their configuration, the structure of the MLP needs to be fine-tuned based on applications. The number of output is decided by the problem. For this example, since onpeak MCP is to be predicted, the MLP has one output. The input factors are decided by their relevance to the output, including load, historical MCP, etc., with a total number of 56 factors. The number of hidden neurons is a parameter adjusted to yield the best prediction result, which is eight. In this example, prediction results are acceptable for a network with six to ten hidden neurons with a slight increase in prediction error. It is important to note that by using the memoryless Quasi-Newton method, the computational requirements to obtain confidence intervals are not significant. It took 0.08 s to provide the five months' MCP prediction only, while taking 0.1 s to provide MCP prediction and confidence interval estimation. Another note is that this paper is for MCP prediction and confidence interval estimation to be used for bidding preparation, but does not directly address bidding preparation. ISO New England's MCP prediction cannot be used by market participants to prepare their bids, since it is published after market participants submit bids.

VI. Conclusions

Cascaded prediction structures are widely used, however, they have not been adequately analyzed. This paper presents an NN prediction and confidence interval estimation method for a cascaded structure and develops a fast implementation algorithm. The method obtains predictions and confidence intervals accurately and efficiently for a classroom problem and for New England ISO's MCP prediction. While MCP prediction by ISO New England is decided by solving a unit commitment and economic dispatch problem based on the bids submitted, our NN prediction is comparable in quality to ISO's prediction. As the New England energy market will be changed to a multisettlement system with separated day-ahead and real-time markets by the end of 2002, our method is currently being extended to predict both markets' MCPs. The method is also generic in the sense that if the gradient information is available, confidence intervals can be derived. Our method can therefore be extended to a broad class of differentiable NNs such as multilayer perceptrons and radial basis function networks.

Acknowledgment

The authors would like to thank S. Rourke of Select Energy and D. Gionet of ISO New England. They would also like to thank the anonymous reviewers whose constructive comments contributed to improve the quality of this paper.

References

[1] A. Khotanzad, R. Afkhami-Rohani, and D. Maratukulam, "ANNSTLF—Artificial Neural Network Short-Term Load Forecaster—Generation three," *IEEE Trans. Power Syst.*, vol. 13, pp. 1413–1422, Nov. 1998.

[2] D. W. Bunn, "Forecasting loads and prices in competitive power markets," *Proc. IEEE*, vol. 88, pp. 163–169, Feb. 2000.

[3] F. Gao, X. Guan, X. Cao, and A. Papalexopoulos, "Forecasting power market clearing price and quantity using a neural network method," in *2000 Power Eng. Soc. Summer Meeting*, vol. 4, Seattle, WA, pp. 2183–2188.

[4] C. M. Bishop, *Neural Networks for Pattern Recognition*. London, U.K.: Oxford Univ. Press, 1995.

[5] S. Haykin, *Neural Networks—A Comprehensive Foundation*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[6] A. S. AlFuhaid, M. A. El-Sayed, and M. S. Mahmoud, "Cascaded artificial neural networks for short-term load forecasting," *IEEE Trans. Power Syst.*, vol. 12, pp. 1524–1529, Nov. 1997.

[7] A. P. Alves da Silva and L. S. Moulin, "Confidence intervals for neural network based short-term load forecasting," *IEEE Trans. Power Syst.*, vol. 15, pp. 1191–1196, Nov. 2000.

[8] C.-Y. Tsai and C.-N. Lu, "Bootstrap application in ATC estimation," *IEEE Power Eng. Rev.*, pp. 40–42, Feb. 2001.

[9] G. Chryssolouris, M. Lee, and A. Ramsey, "Confidence interval prediction for neural network models," *IEEE Trans. Neural Networks*, vol. 7, pp. 229–232, Jan. 1996.

[10] D. K. Ranaweera, G. G. Karady, and R. G. Farmer, "Effect of probabilistic inputs on neural network-based electric load forecasting," *IEEE Trans. Neural Networks*, vol. 7, pp. 1528–1532, Nov. 1996.

[11] N. W. Townsend and L. Tarassenko, "Estimations of error bounds for neural-network function approximators," *IEEE Trans. Neural Networks*, vol. 10, pp. 217–230, Mar. 1999.

[12] D. J. C. MacKay, "Bayesian interpolation," *Neural Comput.*, vol. 4, no. 3, pp. 415–447, 1992.

[13] W. A. Wright, "Bayesian approach to neural-network model with input uncertainty," *IEEE Trans. Neural Networks*, vol. 10, pp. 1261–1270, Nov. 1999.

[14] L. Zhang and P. B. Luh, "Confidence regions for cascaded neural network prediction in power markets," in *IEEE Power Eng. Soc. Winter Meeting*, Columbus, OH, Jan. 2001, pp. 533–538.

[15] G. Papadopoulos, P. J. Edwards, and A. F. Murray, "Confidence estimation methods for neural networks: A practical comparison," *IEEE Trans. Neural Networks*, vol. 12, pp. 1278–1287, Nov. 2001.

[16] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087–1092, 1953.

[17] I. Rivals and L. Personnaz, "Construction of confidence intervals for neural networks based on least squares estimation," *Neural Networks*, vol. 13, pp. 463–484, 2000.

[18] J. R. Donaldson and R. B. Schnabel, "Computational experience with confidence regions and confidence intervals for nonlinear least squares," *Technometrics*, vol. 29, pp. 67–82, Feb. 1987.

[19] D. P. Bertsekas, *Nonlinear Programming*, Second ed: Athena Scientific, 1999.

[20] D. F. Shanno, "Conjugate gradient methods with inexact searches," *Math. Oper. Res.*, vol. 3, no. 3, pp. 244–256, 1978.

[21] E. Barnard, "Optimization for training neural nets," *IEEE Trans. Neural Networks*, vol. 3, pp. 232–240, Mar. 1992.

Li Zhang (S'99–M'02) received the B.S. degree in information and control engineering from Xi'an Jiaotong University, China, in 1992, the M.S. degree in automatic control from Shanghai Jiaotong University, in 1995, and the M.Eng. degree in electrical engineering from National University of Singapore in 1997, respectively. He is currently pursuing the Ph.D. degree at the University of Connecticut, Storrs.

He worked as an Engineer in Singapore for half a year in 1997. He joined the University of Connecticut as a Research Assistant in 1997. From 1999 to 2000, he was a System Administrator for the computer network in the Electrical and Computer Engineering Department, University of Connecticut. His research interests include intelligent systems, signal processing, control systems and optimization, and power systems.

Peter B. Luh (M'80–SM'91–F'95) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C, in 1973, the M.S. degree in aeronautics and astronautics engineering from Massachusetts Institute of Technology, Cambridge, MA, in 1977, and the Ph.D. degree in applied mathematics from Harvard University, Cambridge, MA, in 1980.

Currently, he is the Southern New England Telephone Professor of Communications and Information Technologies with the Department of Electrical and Computer Engineering at the University of Connecticut, Storrs, and is the Director of the Taylor L. Booth Center at the University of Connecticut for Computer Research and Applications. Since 1980, he has been with the University of Connecticut. His major research interests include schedule generation and reconfiguration for manufacturing and power systems.

Dr. Luh is the Editor-in-Chief of the IEEE Transactions on Robotics and Automation, and was an Associate Editor of the IEEE Transactions on Automatic Control.

Krishnan Kasiviswanathan received the M.S. degree in electrical engineering from the University of Connecticut, Storrs, in 1997.

Currently, he is a Senior Financial Engineer at Select Energy, Inc., Berlin, CT, which is a subsidiary of the Northeast Utilities system. His research interests include price and load forecasting and financial modeling in energy markets.