Optimized Approximation Algorithm in Neural Networks Without Overfitting

Yinyin Liu, Janusz A. Starzyk, *Senior Member, IEEE*, and Zhen Zhu, *Member, IEEE*

***Abstract*—In this paper, an optimized approximation algorithm (OAA) is proposed to address the overfitting problem in function approximation using neural networks (NNs). The optimized approximation algorithm avoids overfitting by means of a novel and effective stopping criterion based on the estimation of the signal-to-noise-ratio figure (SNRF). Using SNRF, which checks the goodness-of-fit in the approximation, overfitting can be automatically detected from the training error only without use of a separate validation set. The algorithm has been applied to problems of optimizing the number of hidden neurons in a multilayer perceptron (MLP) and optimizing the number of learning epochs in MLP's backpropagation training using both synthetic and benchmark data sets. The OAA algorithm can also be utilized in the optimization of other parameters of NNs. In addition, it can be applied to the problem of function approximation using any kind of basis functions, or to the problem of learning model selection when overfitting needs to be considered.**

***Index Terms*—Function approximation, neural network (NN) learning, overfitting.**

I. Introduction

UNKNOWN function approximation or model approximation is needed in many areas, and has been widely investigated. Without prior knowledge of the system properties, we can only obtain a limited number of observations and use a set of basis functions to fit the data to a desired level of accuracy in order to approximate the system function. The basis functions could be, for instance, a set of orthogonal functions, Walsh functions, sinusoidal waves, or sigmoid functions. Common approximation methods include least squares fit, neural networks (NNs) in the form of feedforward multilayer perceptrons (MLPs) [1], radial basis function (RBF) networks [2], etc. In NN learning, adding more hidden neurons is equivalent to adding more basis functions in function approximation. In addition to the number of hidden neurons, the training accuracy could also be affected by several other parameters, including the number of layers, the number of training samples, the length of learning period, the choice of neuron activation functions, and the training algorithm. Previous work has shown that NNs can be used as universal approximators [3]–[5]. For universal approximators, how to determine the proper parameters to use in the model without a preset target for training accuracy is one of the major challenges, which makes the design and use of NNs more of an art than a science [6].

In order to optimize the number of hidden neurons, several techniques have been developed in literature, which correlate it with the number of training samples or the input and output layer sizes [7]–[9]. Other work estimates the complexity of the desired function and relates it to the number of hidden neurons [10]. If the NN training uses backpropagation (BP) algorithm, it has been shown that increasing the number of hidden neurons and the number of weights makes it easier to find the global minimum [11], [12]. However, without examining the goodness-of-fit or considering the statistical characteristics of the training data, these approaches are less theoretically sound. Geometric interpretation given in [6] provides some insight into the problem of determining the number of neurons. It helps to find the minimum structure of MLP necessary for a satisfactory approximation of a given problem. However, such method can be only applied to problems with the input space's dimensionality up to two. Some work [13]–[16] on estimating the number of hidden neurons focused on the learning capabilities of the MLP on a training data set without considering the possibility of overfitting.

Using an excessive number of basis functions will cause overfitting, which means that the approximator overestimates the complexity of the target problem. This is usually referred to as the bias/variance dilemma [17]. A natural upper limit for the number of basis functions is the number of available training data points. The major purpose of developing function approximation is to interpolate in a meaningful way between the training samples [18], in order to generalize a model from existing training data and make predictions for novel data. Such generalization capability, usually measured by the generalization error [19], is degraded by overfitting, which leads to a significant deviation in prediction. It was addressed in [6] that finding the minimum structure of MLP in most cases results in the least cost of computation, least requirements on implementation resources, and best generalization. In this sense, determining the optimum number of neurons or finding the minimum structure to prevent overfitting are critical in function approximation.

During BP training in NNs, the weights are adjusted incrementally. Therefore, besides the network size, training accuracy also depends on the number of training epochs. Too many epochs used in BP training will lead to overtraining, which is a concept similar to overfitting.

A lot of effort has been put into studying the overfitting problem in NNs. Some studies show that generalization performance of NN can be improved by introducing additive noise

Manuscript received October 6, 2006; revised April 11, 2007 and August 23, 2007; accepted October 12, 2007.

The authors are with the School of Electrical Engineering and Computer Science, Ohio University, Athens, OH 45701 USA (e-mail: yliu@bobcat.ent.ohiou.edu; starzyk@bobcat.ent.ohiou.edu; zhuz@ohiou.edu).

Digital Object Identifier 10.1109/TNN.2007.915114

to the training samples [18]–[20]. In [18], noise is added to the available training set to generate an unlimited source of training samples. This is interpreted as a kernel estimate of the probability density that describes the training vector distribution. It helps to enhance the generalization performance, speed up the BP algorithm, and reduce the possibility of local minima entrapment [20]. These methods provide a useful tool to expand data sets. However, they only demonstrate improvement on an existing network with preset network parameters. Optimization of the network architecture (for example, the number of neurons) has not been addressed. The design of NNs to avoid overfitting remains an open problem.

To find the optimal network structure with an optimal size of the hidden layer or optimal value of a certain network parameter, constructive/destructive algorithms were adopted to incrementally increase or decrease the parameter to be optimized [21]–[24]. During the constructive/destructive process, cross validation is commonly used to check the network quality [25] and the design parameter is chosen using early stopping [26]–[28]. In these approaches, the available data are divided usually into two independent sets: a training set and a validation or testing set. Only the training set participates in the NN learning, and the validation set is used to compute validation error, which approximates the generalization error. The performance of a function approximation during training and validation is measured, respectively, by training error $\varepsilon_{\text{train}}$ and validation error $\varepsilon_{\text{valid}}$ presented, for instance, in the form of mean squared error (MSE). Once the validation performance stops improving as the target parameter continues to increase, it is possible that the training has begun to fit the noise in the training data, and overfitting occurs. Therefore, the stopping criterion is set so that, when $\varepsilon_{\text{valid}}$ starts to increase, or equivalently, when $\varepsilon_{\text{train}}$ and $\varepsilon_{\text{valid}}$ start to diverge, it is assumed that the optimal value of the target parameter has been reached.

Singular value decomposition (SVD) approach was also used to quantify the significance of increasing the number of neurons in the hidden layer in the constructive/destructive process [29]. The number of neurons is considered sufficient when each additional neuron contributes effect that is lower than an arbitrary threshold. There are several other model selection criteria, such as Akaike's information criterion (AIC) [30] and the minimum description length (MDL) [31], as a function of the model complexity, the training performance, and the number of training samples. Some work applied such information criteria in the problem of finding optimal NN structures [32], [33]. AIC was introduced in order to maximize the mean log-likelihood of a model while avoiding unnecessary complexity. A penalty term was applied to make model with excessive number of independent parameters less desirable. The algorithm using AIC as stopping criterion will choose the model with the minimum AIC. The bias/variance decomposition [13] is a method used to decompose the bias and variance terms from MSE and to measure the sensitivity of a learning model to the training data. Fitting into the available data will reduce the bias while overfitting may induce large variance. In practice, the bias and variance components for a certain learning model are estimated statistically over several training sets samples from the same function. Among several model choices, the one with least bias and variance is chosen as the optimum. Overall, cross validation and early stopping are still the common techniques used in finding optimal network structure up to date.

Nevertheless, in cross validation and early stopping, the use of the stopping criterion based on $\varepsilon_{\text{valid}}$ is not straightforward and requires definite answers to several issues. For example, users have to find out the distribution of data so that training and validation sets can be properly divided and to assure that each of them have good coverage of the input space. In addition, as demonstrated in [11], the validation data have to be representative enough due to its size and data distribution, so that $\varepsilon_{\text{valid}}$ can provide an unbiased estimate of the actual network performance and the real generalization error ε_{gen}. As validation data are statistically sampled, $\varepsilon_{\text{valid}}$ has only a statistical chance to correlate with the generalization error, thus it is not a reliable measure. $\varepsilon_{\text{valid}}$, as a function of target parameter, may have many local minima during the training process. It is not definite which one indicates the occurrence of overfitting [27], [28] and it is even more difficult to find out how likely overfitting actually happened. Therefore, during the constructive/destructive process, users have to go through the process of adjusting the target parameter and observing the variation of $\varepsilon_{\text{valid}}$ to vaguely determine a good place to stop, which is a somewhat empirical and a not well-quantified process. Three classes of better defined stopping criteria based on the concept of early stopping were proposed in [27], from which users can choose based on different concerns on efficiency, effectiveness, or robustness. The first class of stopping criteria (GL) proposes to stop training as soon as the generalization loss, measured by the increase of $\varepsilon_{\text{valid}}$, exceeds a certain threshold. The second class (PQ) evaluates the quotient of generalization loss and training progress so that even if generalization error increases, the rapid decrease of training error will suggest continuation of the process. The third class (UP) suggested stopping the process when the generalization error kept increasing in several successive steps. It helped the users to choose stopping criterion in a systematic and automatic way to avoid the *ad hoc* process. However, as long as cross validation is used, the methods require omission of the validation set in the training stage, which is a significant waste of the precious data available for training in some real-life cases, e.g., plant data set [34].

In general, overfitting occurs when excessive number of neurons is used in the network. In these cases, although the $\varepsilon_{\text{valid}}$ may not be severely degraded, the network does overestimate the complexity of the problem and it cost more resources to train and implement. The case of severe overfitting that goes undetected using the validation set can be easily illustrated with an example of a synthetic data set obtained from a noisy sine wave signal approximated by polynomial functions. Fig. 1(a) shows training and validation data sets. Fig. 1(b) shows the values of training and validation errors as a function of the orders of approximating polynomials. Also shown in Fig. 1(b) is (usually unknown) generalization error, which measures the deviation of the approximating result from the original sine wave. As illustrated in Fig. 1(b), the validation error did not increase significantly enough to indicate severe overfitting that occurs when the order of approximating polynomial was higher than 18.

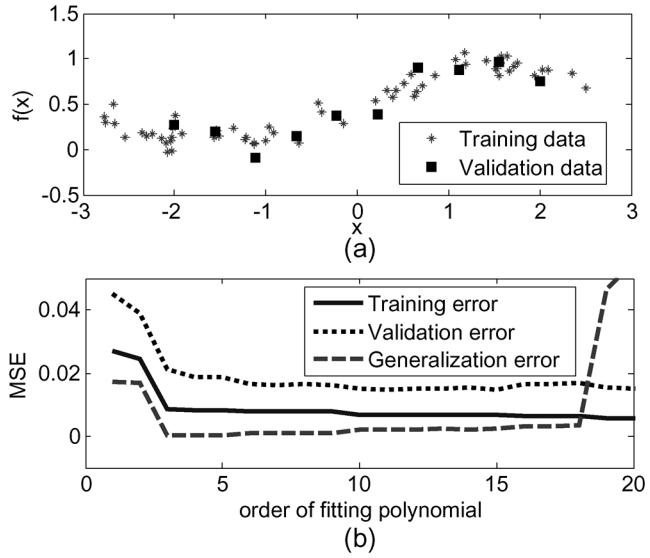

Fig. 1. (a) Training and validation set. (b) Variation of errors in function approximations.

Thus, it is desirable to have a measure that can quantify underfitting or overfitting of a network on a given learning problem. An algorithm based on such measure should be able to recognize the occurrence of overfitting by examining the training error without using a validation set and show where the process can be safely stopped so that the optimal structure of the MLP for a given problem is found. In this paper, a signal-to-noise-ratio figure (SNRF) is defined to measure the goodness-of-fit using the training error. Based on the SNRF measurement, an optimized approximation algorithm (OAA) is proposed to avoid overfitting in function approximation and NN design applications.

The organization of this paper is as follows. In Section II, the definition and estimation method of SNRF is introduced. The OAA procedure utilizing SNRF-based stopping criterion is demonstrated in Section III. The OAA is validated using both simulated and benchmark data, as shown in Section IV. Finally, features of the OAA are discussed in Section V.

II. Estimation of Signal-to-Noise-Ratio Figure

A. SNRF of the Error Signal

In order to have a clear indication of overfitting, we need to examine the difference between the approximated function and the training data. This difference, which is defined as the error signal in this work, comes from two possible sources: the approximation error due to the limited training accuracy in approximation with the given set of basis functions and an unknown level of noise in the training data. The noise can be the result of multiple causes, such as input noise, output noise, or system disturbance which will all be treated as the output noise. In function approximation, without any knowledge of the noise sources and based on the central limit theorem, we can assume the noise as white Gaussian noise (WGN) without losing generality. A critical question is whether there is still useful signal information left to be learned in the error signal. If there is, based on the assumption that the target function we try to approximate is continuous and that the noise is WGN, we can estimate the level of signal and noise in the error signal. The ratio of the estimated signal level to the noise level in the error signal is defined as SNRF, and it is used to measure the amount of information left unlearned in the error signal. The SNRF can be precalculated for a signal that contains solely WGN. The comparison of SNRF of the error signal with that of WGN determines whether WGN dominates in the error signal. If the noise dominates, there is little useful information left in the error signal, and there is no point to reduce it anymore as this will lead to overfitting. The estimation of SNRF will be first illustrated using a 1-D function approximation problem, followed by the discussion for multidimensional problems.

B. SNRF Estimation for 1-D Function Approximation

Assume that in a 1-D function approximation problem, training data are uniformly sampled from the input space $X \in \Re^1$ with additive noise at an unknown level. An approximation is obtained using a certain set of basis functions. The error signal e contains a noise component denoted by n, and an approximation error signal component, which is the useful signal left unlearned, and therefore, denoted by s

$$e_i = s_i + n_i = s_i + \beta m_i, \qquad (i = 1, 2, \ldots N) \tag{1}$$

where N represents the number of samples. Without losing generality, n can be modeled as a WGN process with standard deviation β, and m stands for a WGN process with unit standard deviation. The energy of the error signal e is also composed of signal and noise components

$$E_{s+n} = E_s + E_n. \tag{2}$$

The energy of e can be calculated using the autocorrelation function

$$E_{s+n} = C(e_i, e_i) = \sum_{i=1}^{N} e_i^2 \tag{3}$$

where C represents the correlation calculation. Notice that a presumption is made that the target function needs to be continuous, and the approximation $\hat{F}$ is usually a continuous function. Practically, the useful signal left unlearned s is also a continuous function. We could further assume that, if treated as time signals, the target function and $\hat{F}$ both have relatively small bandwidth compared to the sampling rate or to the noise bandwidth. As a result, there is a high level of correlation between two neighboring samples of s. Consequently

$$C(s_i, s_{i-1}) \approx C(s_i, s_i) \tag{4}$$

where s_{i-1} represents the (circular) shifted version of the s. Due to the nature of WGN, noise of a sample is independent of noise of neighboring samples

$$C(n_i, n_{i-1}) = C(\beta m_i, \beta m_{i-1}) = 0 \tag{5}$$

where n_{i-1} represents a replica of n_i shifted by one sample. Because the noise component is independent of the signal com-

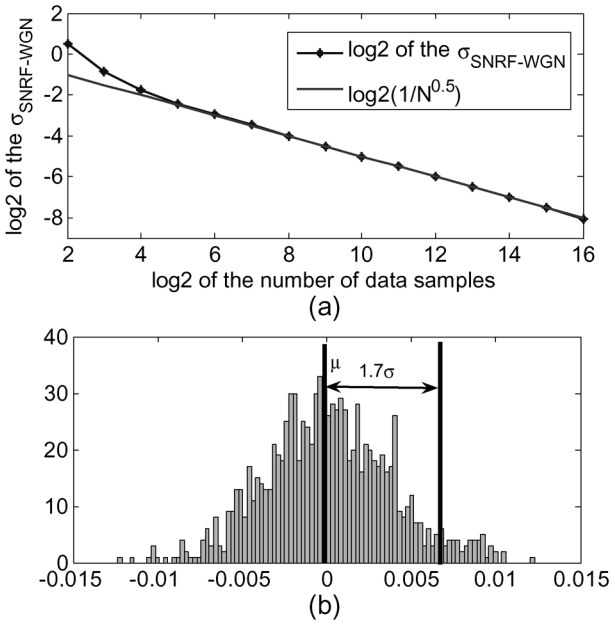

Fig. 2. (a) Standard deviation of SNRF_{WGN} in a 1-D case. (b) Histogram of SNRF_{WGN} for 2^{16} samples in a 1-D case.

ponent, the correlation of e_i with its shifted copy e_{i-1} approximates the signal energy, as shown in

$$C(e_i, e_{i-1}) = C(s_i, s_{i-1}) \approx E_s. \tag{6}$$

The difference between the autocorrelation with no time shift defined in (3) and $C(e_i, e_{i-1})$ gives the noise energy in the error signal

$$E_n = E_{s+n} - E_s = C(e_i, e_i) - C(e_i, e_{i-1}). \tag{7}$$

The ratio of the signal level to the noise level, defined as the SNRF of the error signal, is obtained as

$$\text{SNRF}_e = \frac{E_s}{E_n} = \frac{C(e_i, e_{i-1})}{C(e_i, e_i) - C(e_i, e_{i-1})}. \tag{8}$$

Notice that in SNRF, the signal component and noise component are decomposed by using the correlation between neighboring samples. In the bias/variance decomposition, similar estimations of the signal or noise level are obtained from bias and variance components, which are calculated statistically in common practice.

When learning of the target function improves, it is expected that the useful signal left unlearned in the error signal is reduced, while the noise component does not change so that SNRF_e will decrease. In order to detect the existence of useful signal in e, the SNRF_e has to be compared with the SNRF estimated for WGN using the same number of samples. When there is no signal in e, we have

$$e = s + n = n = \beta m. \tag{9}$$

The SNRF for WGN is calculated as

$$\begin{aligned} \text{SNRF}_{\text{WGN}} &= \frac{C(n_i, n_{i-1})}{C(n_i, n_i) - C(n_i, n_{i-1})} \\ &= \frac{C(m_i, m_{i-1})}{C(m_i, m_i) - C(m_i, m_{i-1})}. \end{aligned} \tag{10}$$

It is observed that the SNRF_{WGN} is independent of the noise level β, which means that SNRF_{WGN} only needs to be estimated with unit standard deviation in order to obtain the general characterization for any level of WGN. The expected value of the correlation $C(m_i, m_{i-1})$ is zero, which would intuitively indicate a zero SNRF_{WGN}. However, SNRF_{WGN} is estimated using a limited number of samples, thus it is a random value related to the number of samples N. Average value and standard deviation of SNRF_{WGN} can be derived for a given N

$$\mu_{\text{SNRF\_WGN}}(N) = \mu\left[\frac{C(m_i, m_{i-1})}{C(m_i, m_i) - C(m_i, m_{i-1})}\right]. \tag{11}$$

Because $C(m_i, m_i) \gg C(m_i, m_{i-1})$, we have

$$\begin{aligned} \mu_{\text{SNRF\_WGN}}(N) &\approx \mu\left[\frac{C(m_i, m_{i-1})}{C(m_i, m_i)}\right] \approx \frac{\mu[C(m_i, m_{i-1})]}{N} \\ &= 0. \end{aligned} \tag{12}$$

$$\begin{aligned} \sigma_{\text{SNRF\_WGN}}(N) &= \sigma\left[\frac{C(m_i, m_{i-1})}{C(m_i, m_i) - C(m_i, m_{i-1})}\right] \\ &\approx \sigma\left[\frac{C(m_i, m_{i-1})}{C(m_i, m_i)}\right] \\ &\approx \frac{\sigma[C(m_i, m_{i-1})]}{N} \\ &= \sqrt{\sum_{i=1}^{N} (\sigma(m_i m_{i-1}))^2} \Big/ N \\ &= \frac{\sqrt{N}}{N} = \frac{1}{\sqrt{N}}. \end{aligned} \tag{13}$$

Note that the samples of SNRF_{WGN} are statistically independent. According to the central limit theorem, if N is large enough, the samples of SNRF_{WGN} tend to follow Gaussian distribution with mean $\mu_{\text{SNRF\_WGN}}$ and standard deviation $\sigma_{\text{SNRF}_{\text{W}}\text{GN}}$. In Fig. 2(a), $\sigma_{\text{SNRF\_WGN}}(N)$ from a 10 000-run Monte Carlo simulation is shown in the logarithmic scale as a function of the number of samples. The estimated $\sigma_{\text{SNRF\_WGN}}(N)$ in (13) agrees with the simulation results, especially for the N values larger than 64. Such estimation is expected to work well for the sample numbers available in real-world training data sets.

C. The 1-D SNRF-Based Stopping Criterion

The stopping criterion in OAA can now be determined by testing the hypothesis that SNRF_e and SNRF_{WGN} are from the same population. The value of SNRF_e at which the hypothesis is rejected constitutes a threshold below which training OAA is stopped. Fig. 2(b) illustrates the histogram of SNRF_{WGN} with 2^{16} samples, as an example. It is observed that the $p = 5\%$ significance level [35] can be approximated by the average value plus 1.7 times standard deviations for an arbitrary N. As shown in Fig. 2(b), the threshold can be calculated using $\mu + 1.7\sigma = 0 + 1.7 \times 0.0039 = 0.006$ for 2^{16} samples. Notice it agrees with the threshold of 5% significance level calculated using Gaussian distribution with mean $\mu_{\text{SNRF\_WGN}}$ and standard deviation $\sigma_{\text{SNRF\_WGN}}$.

SNRF-based stopping criterion in OAA can be defined as an SNRF_e smaller than the threshold determined by (14), in which

case, there is at least 95% probability that error signal represents a WGN and learning must stop to avoid overfitting

$$th_{\text{SNRF\_WGN}}(N) = \mu_{\text{SNRF\_WGN}}(N) + 1.7\sigma_{\text{SNRF\_WGN}}(N). \tag{14}$$

The threshold can be recalculated for different significance levels if needed, also based on the mean $\mu_{\text{SNRF\_WGN}}$ and standard deviation $\sigma_{\text{SNRF\_WGN}}$ derived in (12) and (13).

In the previous discussion, (6) and (7) have been developed based on the assumption that e could be treated as a signal with evenly spaced samples. In a general 1-D function approximation problem, the input samples may be unevenly spaced. Yet, E_{s+n}, E_s, and E_n can still be approximated using (3), (6), and (7), respectively. In addition, in the cases when only sparse data samples are available, the data set can be expanded using the approaches in [18]–[20]. Thus, the SNRF$_e$ can be estimated using (8) and the overfitting is determined by comparison of SNRF$_e$ with the threshold in (14).

D. SNRF Estimation for Multidimensional Function Approximation

In a general multidimensional function approximation problem, the training data are usually randomly sampled from the input space $X \in \Re^D$. The method used to estimate SNRF in the 1-D case cannot be directly applied to such multidimensional problem. However, we could still assume that variation of s along each of the dimensions is slow compared to the average sampling distance. Thus, the same principle of signal and noise level estimation using correlation may be utilized. Because s changes slowly in all directions, the continuous function can be locally approximated around e_p using weighted average of a set of $M+1$ points, which includes e_p and its M neighbors with the shortest distances. These points are expected to have correlated values, whereas the noise on these points is assumed to be WGN and has independent samples. As a result, the signal and noise levels at each sample $e_p (p = 1, 2, \ldots N)$ can be estimated through the correlation with its M nearest neighbors and computed using a weighted combination of the products of e_p values with each of its neighbors $e_{pi} (i = 1, 2, \ldots M)$. Because the samples of e_i are assumed to be spatially correlated, the distances between samples can be used to calculate the weight values. In a D-dimensional space, the weights are obtained based on the scaled distance d_{pi} between e_p and e_{pi} to the power of D, and normalized, as given in the following, where $d_{pi} = \|e_p - e_{pi}\|$:

$$w_{pi} = \left[\frac{1}{d_{pi}^D}\right] \Bigg/ \left[\sum_{i=1}^{M} \frac{1}{d_{pi}^D}\right] \begin{pmatrix} p = 1, 2, \ldots N \\ i = 1, 2, \ldots N \end{pmatrix}. \tag{15}$$

Thus, the overall signal level of e can be calculated as

$$E_s = \sum_{p=1}^{N} E_{sp} = \sum_{p=1}^{N} \sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi}. \tag{16}$$

As in (3), the autocorrelation of e_i estimates signal plus noise level

$$E_{s+n} = \sum_{i=1}^{N} e_i^2. \tag{17}$$

Finally, the SNRF$_e$ for M neighbors approach in a multidimensional input space is computed as

$$\text{SNRF}_e = \frac{E_s}{E_n} = \frac{\sum_{p=1}^{N} \sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi}}{\sum_{i=1}^{N} e_i^2 - \sum_{p=1}^{N} \sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi}}. \tag{18}$$

Notice that when applied to 1-D cases with $M = 1$, (18) is identical to (8).

The same calculation is done for WGN with unit standard deviation to characterize the SNRF$_{\text{WGN}}$ in multidimensional space. When there is no signal, SNRF$_{\text{WGN}}$ is estimated using (18) with $e = n$. In the calculation of E_{sp} of WGN, $e_p \cdot e_{pi}$ is an independent random process with respect to p or i. Because

$$\sum_{i=1}^{M} w_{pi} = 1 \tag{19}$$

we can have

$$\sigma[e_p \cdot e_{pi}] \geq \sigma\left[\sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi}\right] \geq \sigma\left[\sum_{i=1}^{M} \frac{1}{M} \cdot e_p \cdot e_{pi}\right] \tag{20}$$

where $\sigma\left[\sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi}\right] = \sigma_{p,\text{WGN}}$ is the standard deviation of the perceived signal energy at sample e_p in WGN. It has the minimum value when the w_{pi} has equal values (i.e., with uniform sampling distance), which sets the lower bound. Notice that the upper and lower bounds in (20) are equal for $M = 1$, independently of the input space dimensionality. For $M > 1$, the standard deviation gets closer to the upper bound in problems with large dimensionality D, because the closest neighbor dominates the weight calculation.

In the estimation of $\sigma\left[\sum_{p=1}^{N} \sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi}\right]$, it has to be considered that not all the $e_p \cdot e_{pi}$ items are independent of each other with respect to p and i. For instance, when points p and p_1 are the closest neighbors to each other, $w_{p1} e_p e_{p1}$ is calculated twice in $\sum_{p=1}^{N} \sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi}$. In the worst case, all the terms may appear twice, therefore, we have

$$\begin{aligned} \sigma\left[\sum_{p=1}^{N} \sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi}\right] &\leq \sqrt{2}\sqrt{N} \cdot \sigma\left[\sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi}\right] \\ &\leq \sqrt{2}\sqrt{N} \cdot \sigma[e_p \cdot e_{pi}] \\ &= \sqrt{2N}. \end{aligned} \tag{21}$$

Then, we have the estimate for the standard deviation of SNRF$_{\text{WGN}}$ as follows:

$$\sigma_{\text{SNRF\_WGN}}(N) \approx \frac{\sigma\left[\sum_{p=1}^{N} \sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi}\right]}{N} \leq \frac{\sqrt{2}}{\sqrt{N}}. \tag{22}$$

Also, the average of SNRF$_{\text{WGN}}$ is estimated as

$$\mu_{\text{SNRF\_WGN}}(N) \approx 0. \tag{23}$$

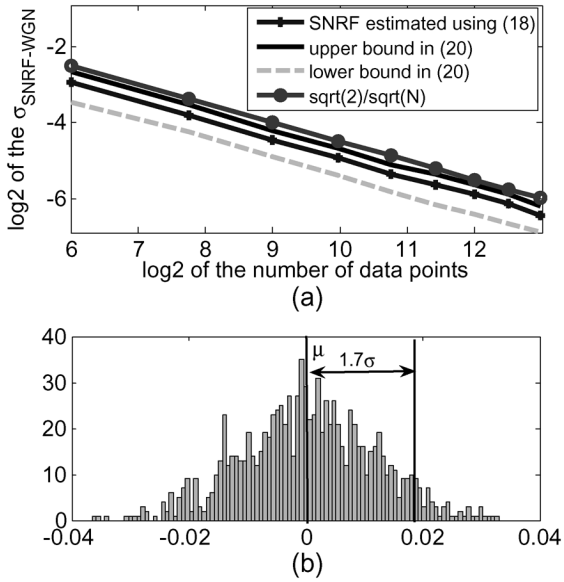

Fig. 3. (a) Standard deviation of $\mathrm{SNRF}_{\mathrm{WGN}}$ in a 3-D case with $M = 3$. (b) Histogram of $\mathrm{SNRF}_{\mathrm{WGN}}$ for 8000 samples in a 3-D case with $M = 3$.

E. Multidimensional SNRF-Based Stopping Criterion

Notice that the estimation of $\sigma_{\mathrm{SNRF\_WGN}}(N)$ and $\mu_{\mathrm{SNRF\_WGN}}(N)$ using (22) and (23) is no longer a function of the number of samples in the neighborhood or problem dimensionality. Such simplification yields a universal detection threshold. $\sigma_{\mathrm{SNRF\_WGN}}(N)$ for $M = 3$ in a 3-D case from a 1000-run Monte Carlo simulation is shown in the logarithmic scale in Fig. 3(a). The distances among WGN samples are randomly generated. The estimated $\sigma_{\mathrm{SNRF\_WGN}}(N)$ in (22) is consistent with an upper bound of $\sqrt{2}/\sqrt{N}$, and the bounds developed in (20) are validated.

Fig. 3(b) shows the histogram of $\mathrm{SNRF}_{\mathrm{WGN}}$ for 8000 samples in the 3-D case with $M = 3$. We note that the threshold of the significance level $p = 5\%$ can be approximated by the average value plus 1.7 times the standard deviations. With $N = 8000$ the threshold is calculated as $\mu + 1.7\sigma = 0 + 1.7 \times 0.0115 = 0.0195$. If not all the samples are independent, central limit theorem does not apply and the distribution of $\mathrm{SNRF}_{\mathrm{WGN}}$ is not Gaussian. In such case, the upper estimate of the standard deviation in (22) is used. The threshold can be experimentally established as the average value plus 1.2 times upper estimate of the standard deviation, to achieve the 5% significance level. Note that this result coincides with (14) for $M = 1$

$$th_{\mathrm{SNRF\_WGN}}(N) = \mu_{\mathrm{SNRF\_WGN}}(N) + 1.2 \cdot \sigma_{\mathrm{SNRF\_WGN}}(N)$$
$\sigma_{\mathrm{SNRF\_WGN}}(N)$ as the approximated upper limit in (22). (24)

While using $M > 1$ can improve estimation of the signal level by greater noise filtering when a large number of training samples is available, we did not observe a significant change in the detection threshold levels, comparing to $M = 1$. Thus, using $M = 1$ is preferred for computing efficiency even in multidimensional cases, when the number of training data is small.

In summary, a method for estimating the SNRF of the error signal has been demonstrated. By comparing SNRF_e with $\mathrm{SNRF}_{\mathrm{WGN}}$, we are able to develop the optimized approximation algorithm (OAA) as discussed in Section III. The threshold for the OAA stopping criterion is determined from the estimate of $\mathrm{SNRF}_{\mathrm{WGN}}$, and can be applied to problems of an arbitrary number of samples and dimensions.

III. Optimized Approximation Algorithm

Using SNRF, we can estimate the signal level and the noise level for the error signal and then determine the amount of useful signal information left unlearned. When there is no information left, the learning process must be stopped, and the optimal approximation has been obtained without overfitting. Otherwise, the target parameter has to be increased to improve the learning and reduce the approximation error. The following procedure describes the basic steps of the OAA for the optimization of a given parameter of the NNs.

- Step 1) Assume that an unknown function F, with input space $X \in \Re^D$, is described by N training samples as $F(x_i) = u_i$, $(i = 1, 2, \ldots N)$.
- Step 2) The signal detection threshold is precalculated for the given number of samples N based on $th_{\mathrm{SNRF\_WGN}}(N) = 1.7/\sqrt{N}$.
- Step 3) Select B as the initial value for the target parameter.
- Step 4) Use the MLP (or other learning models) to obtain the approximated function $\hat{F}(x_i) = a_i$ $(i = 1, 2, \ldots N)$.
- Step 5) Calculate the error signal $e_i = u_i - a_i$, $(i = 1, 2, \ldots N)$.
- Step 6) Determine SNRF of the error signal e_i, SNRF_e. For a 1-D problem, use (8); for a multidimensional problem, use (18).
- Step 7) Stop if the SNRF_e is less than $th_{\mathrm{SNRF\_WGN}}$, or if B exceeds its maximum value. Otherwise, increment B and repeat Steps 4)–7).
- Step 8) If SNRF_e is equal to or less than $th_{\mathrm{SNRF\_WGN}}$, $\hat{F}$ is the optimized approximation.

IV. Simulation and Discussion

An MLP is used as an example learning system to demonstrate the use of the proposed OAA. The MLP contains the input layer and the output layer with linear transfer functions and hidden layers with nonlinear transfer functions in the middle. OAA with SNRF-based stopping criterion will be tested in two aspects, optimization of the number of hidden neurons and optimization of the number of learning epochs, using synthetic data sets and benchmark data sets. First, the synthetic data sets are studied because we know the true target function so that real generalization error $\varepsilon_{\mathrm{gen}}$ can be calculated and the results provide a visual insight to the problem and its proposed solution. Subsequently, the benchmark data sets provide justification for the use of OAA in practical applications.

In all the simulation examples, when OAA is tested in optimization of the number of hidden neurons, the least squared learning method (LSM) proposed in [36] as initialization method will be used as training method in this paper. In LSM, the adaptation of weights in MLP is based on the least squared calculation so that the learning performance is only affected by

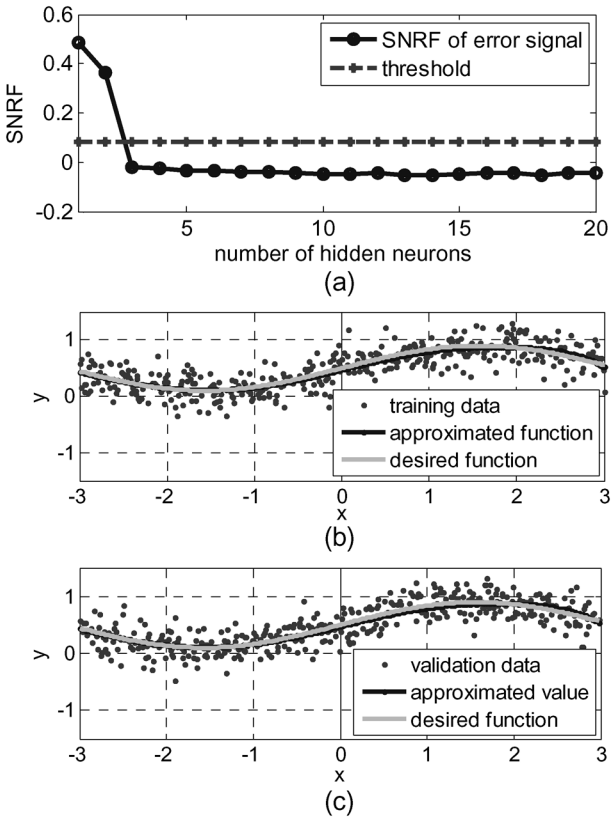

Fig. 4. Simulation I: optimization of number of hidden neurons. (a) SNRF of the error signal and the threshold. (b) Training performance. (c) Validation performance.

the number of hidden neurons representing the number of basis functions without concerning the number of iterations.

In addition, in all the simulation examples, when the number of learning epochs is optimized, an MLP with preset structure is trained using the BP method, implemented using the MATLAB NNs toolbox. The SNRF-based criterion in OAA will determine when to stop the learning to avoid overtraining (overfitting).

It is expected that when the SNRF-based criterion recognizes overfitting, either $\varepsilon_{\text{train}}$ and $\varepsilon_{\text{valid}}$ will start to diverge from each other, or $\varepsilon_{\text{valid}}$ will reach a minimum. Such observation will help to prove the effectiveness of the OAA with the SNRF-based stopping criterion. The results, including the stopping points and corresponding $\varepsilon_{\text{train}}$, $\varepsilon_{\text{valid}}$, and ε_{gen} (for synthetic data) from OAA will be compared with those from four other classes of stopping criteria described in [27] and [30]. Specific criteria used in the comparison are denoted as follows: AIC [30], $\text{GL}_1 \sim \text{GL}_5$ (generalization loss with thresholds $1 \sim 5$ [27]), $\text{PQ}_{0.5} \sim \text{PQ}_3$ (generalization loss over training progress with thresholds $0.5 \sim 3$ [27]), and $\text{UP}_2 \sim \text{UP}_8$ (the number of successive increases in the generalization error [27]). To calculate the AIC for MLP, the number of free parameters is equal to the overall number of weights and the bias.

Simulation I: 1-D Function Approximation

First, the desired function to be approximated is $y = 0.4 \sin x + 0.5$, which is the same target function as used in [18]. A four-layered MLP is used as the learning prototype with the number of hidden neurons to be optimized. The number

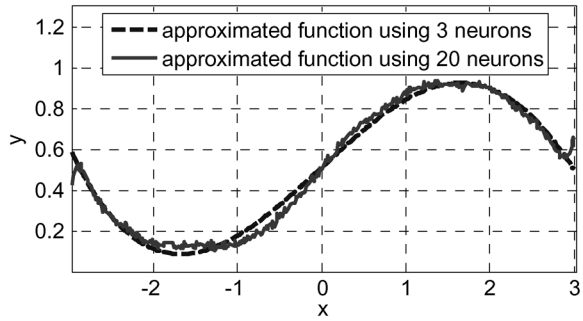

Fig. 5. Comparison of approximated function using three and 20 neurons.

TABLE I
SIMULATION I: RESULT COMPARISON FOR OPTIMIZING NUMBER OF NEURONS

Stopping Criteria	Optimum number of hidden neurons	Training error ε_{train}	Validation error ε_{valid}	Generalization error ε_{gen}
SNRF	3	0.11278	0.10559	0.0022411
AIC	3	0.11278	0.10559	0.0022411
GL_1	10	0.11084	0.10623	0.0024636
GL_2	15	0.10895	0.10779	0.0038922
GL_3	17	0.10802	0.10994	0.0057212
GL_5	18	0.10701	0.11148	0.0091781
$\text{PQ}_{0.5}$	10	0.11084	0.10623	0.0024636
$\text{PQ}_{0.75}$	10	0.11084	0.10623	0.0024636
PQ_1	11	0.11075	0.10549	0.0025474
PQ_2	14	0.1095	0.10677	0.0042547
PQ_3	14	0.1095	0.10677	0.0042547
UP_2	7	0.11123	0.10528	0.0027053
UP_3	7	0.11123	0.10528	0.0027053
UP_4	9	0.11112	0.10566	0.002306
UP_6	13	0.10955	0.10594	0.0032101
UP_8	17	0.10802	0.10994	0.0057212

of hidden neurons in these two hidden layers is set equal in the following simulation. The training and validation data sets, containing 200 samples each, are randomly sampled from the input space, and the outputs are subjected to WGN with a standard deviation of 0.2.

Simulation results show that SNRF_e goes below the threshold when the number of hidden neurons on each layer is more than three for the four-layered MLP, as can be seen from Fig. 4(a). As shown in Fig. 4(b), the approximated function for the training data obtained from the MLP with size 1-3-3-1 approximates the target function well. At the same time, it makes reasonable predictions on the unseen validation data, as shown in Fig. 4(c). Although the $\varepsilon_{\text{valid}}$ produced by MLP with 20 neurons is only 6% higher than that by three neurons, the MLP with 20 neurons seriously overestimates the complexity of the problem and the overfitting definitely shows up, as in the comparison in Fig. 5.

The results from different kinds of stopping criteria are compared in Table I. Among all the stopping criteria, SNRF-based stopping criterion suggests the minimum structure that can efficiently handle the target problem and yield the minimum generalization error, which corresponds to possibly the best generalization ability.

In [18], the same target function is approximated using an MLP with size 1-13-1. It was demonstrated that the overfitting problem can be mitigated to some degree by using additive

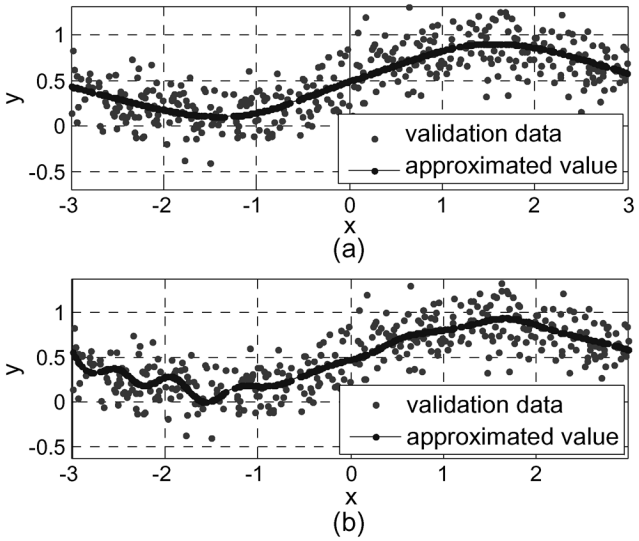

Fig. 6. Approximated function using (a) ten and (b) 200 learning epochs.

TABLE II
SIMULATION I: RESULT COMPARISON FOR OPTIMIZING NUMBER OF LEARNING EPOCHS

Stopping Criteria	Optimum number of learning epochs	Training error ε_{train}	Validation error ε_{valid}	Generalization error ε_{gen}
SNRF	10	0.1086	0.0987	0.00053
AIC	N/A	N/A	N/A	N/A
GL1	180	0.1064	0.1003	0.0073
GL2	Incomplete	Incomplete	Incomplete	Incomplete
GL3	Incomplete	Incomplete	Incomplete	Incomplete
GL5	Incomplete	Incomplete	Incomplete	Incomplete
PQ0.5	30	0.1076	0.0989	0.0026
PQ0.75	30	0.1076	0.0989	0.0026
PQ1	30	0.1076	0.0989	0.0026
PQ2	130	0.1076	0.0989	0.0026
PQ3	130	0.1076	0.0989	0.0026
UP2	50	0.1066	0.0993	0.0065
UP3	60	0.1074	0.0991	0.0037
UP4	90	0.1078	0.0988	0.0012
UP6	150	0.1075	0.0992	0.0029
UP8	160	0.1073	0.0990	0.0034

noise to expand the sparse data set [18]. However, without optimizing the network structure, the approximated function still deviates from the desired function. Using the proposed OAA, the SNRF-based stopping criterion shows that the optimal number of hidden neurons for this three-layered MLP is five.

With such 1-5-1 MLP, the number of learning epochs of the BP algorithm can be optimized using SNRF-based stopping criterion in OAA. It suggests stopping the training after ten epochs. The approximated function after ten epochs is compared with that after 200 epochs in Fig. 6, which shows that large number of learning epochs induces overfitting and the SNRF-based stopping criterion is able to stop the learning process at the optimum point.

The results of optimizing the number of learning epochs from different kinds of stopping criteria are compared in Table II. SNRF-based stopping criterion suggests stopping the training with minimum number of learning epochs in this case and shows minimum generalization error. Notice that because the network structure does not change during the process, the AIC stopping criterion can not be applied and will be denoted as "N/A" in the result tables. Some of the stopping criteria, including GL_2, GL_3, and GL_5, have not been met even with the maximum number of learning epochs and will be denoted as "incomplete" in the result tables.

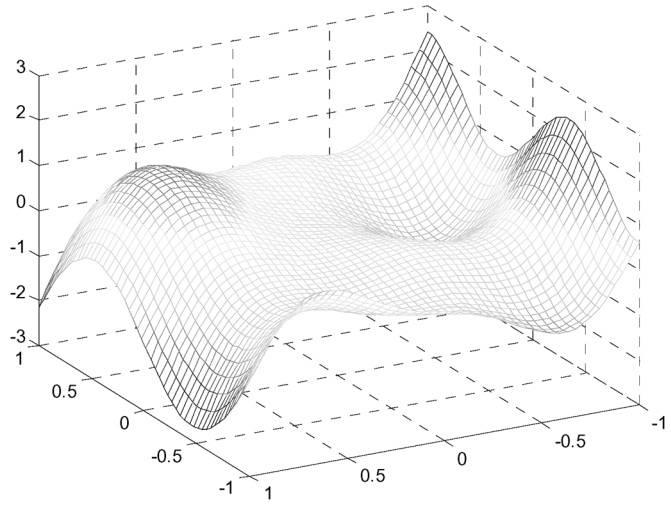

Fig. 7. Multidimensional function to be approximated.vsk 6pt

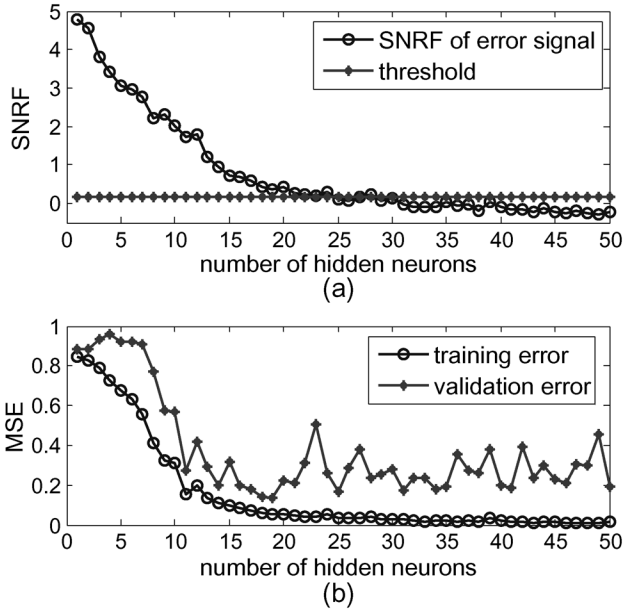

Fig. 8. Simulation II: optimization of number of hidden neurons. (a) SNRF of the error signal and the threshold. (b) Training and validation errors.

Simulation II: 2-D Function Approximation

A function $y = x_2^2 + \sin(3x_2) + 2x_1^2 \sin(4x_1) + x_1 \sin(4x_2)$ is used as the target function to illustrate a multidimensional case, as shown in Fig. 7. Data points are randomly sampled adding WGN with a standard deviation of 0.1 to produce training and validation data sets, each containing 100 samples.

The OAA is applied to optimize the number of hidden neurons of a four-layered MLP. $SNRF_e$ falls below the threshold

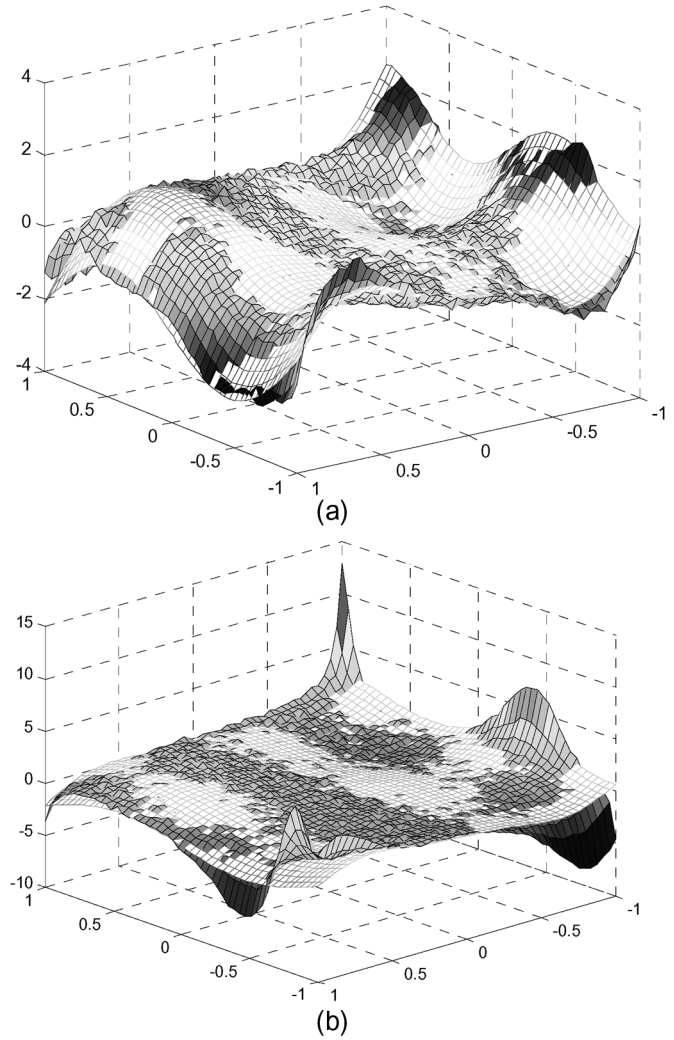

Fig. 9. Approximated function using (a) 2-25-25-1 and (b) 2-35-35-1 MLPs.

when the number of hidden neurons exceeds 25 as shown in Fig. 8. It may be seen that the validation error has many local minima located in the range from 25 to 35 neurons. In this case, it would be difficult to exactly determine where overfitting begins by using $\varepsilon_{\text{valid}}$. Using such 2-25-25-1 MLP as a function approximator, the approximated function in the given input space replicates the desired function well, as in Fig. 9(a). However, using 35 hidden neurons, the approximated function has significant deviations from the target function at the unseen data, which is illustrated in Fig. 9(b). The function surface indicates obvious overfitting. The optimal network size with 25 neuron optimum is correctly predicted by the OAA.

The optimization results based on different stopping criteria are compared in Table III. In this case, other methods stop too early resulting in larger generalization errors.

Subsequently, OAA was used in a three-layered MLP with size 2-25-1 to find proper number of learning epochs, and the results are compared with others methods in Table IV. Again, we can see that the proposed SNRF criterion yields an optimum number of the training epochs with the smallest validation and generalization errors.

TABLE III
SIMULATION II: RESULT COMPARISON FOR OPTIMIZING NUMBER OF HIDDEN NEURONS

Stopping Criteria	Optimum number of hidden neurons	Training error ε_{train}	Validation error ε_{valid}	Generalization error ε_{gen}
SNRF	25	0.023222	0.22471	0.065502
AIC	1	0.85216	0.87783	0.89908
GL1	6	0.64278	0.82437	0.64931
GL2	6	0.64278	0.82437	0.64931
GL3	6	0.64278	0.82437	0.64931
GL5	16	0.072757	0.41385	0.15182
PQ0.5	6	0.64278	0.82437	0.64931
PQ0.75	16	0.072757	0.41385	0.15182
PQ1	16	0.072757	0.41385	0.15182
PQ2	16	0.072757	0.41385	0.15182
PQ3	16	0.072757	0.41385	0.15182
UP2	6	0.64278	0.82437	0.64931
UP3	16	0.072757	0.41385	0.15182
UP4	16	0.072757	0.41385	0.15182
UP6	16	0.072757	0.41385	0.15182
UP8	17	0.055236	0.19659	0.11702

TABLE IV
SIMULATION II: RESULT COMPARISON FOR OPTIMIZING NUMBER OF LEARNING EPOCHS

Stopping Criteria	Optimum number of learning epochs	Training error ε_{train}	Validation error ε_{valid}	Generalization error ε_{gen}
SNRF	21	0.017228	0.072691	0.02101
AIC	N/A	N/A	N/A	N/A
GL1	41	0.015332	0.07962	0.024886
GL2	41	0.015332	0.07962	0.024886
GL3	41	0.015332	0.07962	0.024886
GL5	41	0.015332	0.07962	0.024886
PQ0.5	41	0.015332	0.07962	0.024886
PQ0.75	41	0.015332	0.07962	0.024886
PQ1	41	0.015332	0.07962	0.024886
PQ2	61	0.014128	0.13411	0.031172
PQ3	101	0.012752	0.16083	0.039461
UP2	41	0.015332	0.07962	0.024886
UP3	61	0.014128	0.13411	0.031172
UP4	101	0.012752	0.16083	0.039461
UP6	131	0.0097924	0.42093	0.095002
UP8	171	0.0041367	0.78034	0.17466

Simulation III: Mackey–Glass Data Set

The Mackey–Glass data is a time–series data set obtained from a physiological system [37]. In the first test on Mackey–Glass data, MLP is used to predict the eight sample based on the preceding seven samples, assuming that every eight sample in the time series is a function of previous seven samples. The training and validation sets contain 500 and 293 samples, respectively. As shown in Fig. 10(a), it is predicted with 95% probability that overfitting occurs when the four-layered network has more than two neurons on hidden layers. This prediction is confirmed by the increase of $\varepsilon_{\text{valid}}$ and divergence between $\varepsilon_{\text{train}}$ and $\varepsilon_{\text{valid}}$, as shown in Fig. 10(b).

The results of optimizing number of neurons from different stopping criteria are compared in Table V. The SNRF-based OAA suggests the minimum structure and yields the smallest

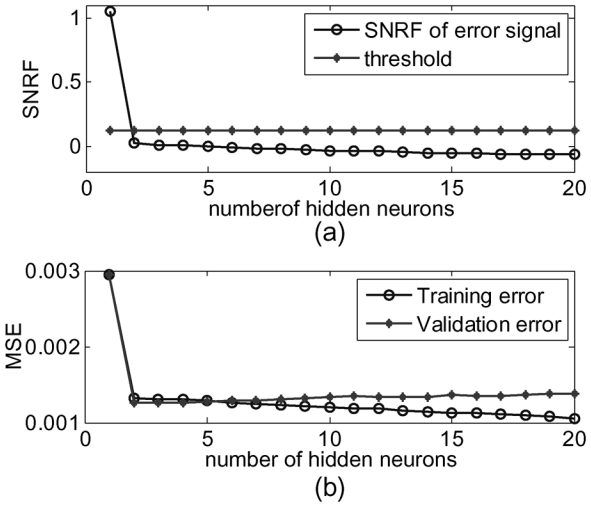

Fig. 10. Simulation III: optimization of number of hidden neurons. (a) SNRF of the error signal and the threshold. (b) Training and validation errors.

TABLE V
SIMULATION III: RESULT COMPARISON FOR OPTIMIZING NUMBER OF HIDDEN NEURONS

Stopping Criteria	Optimum number of hidden neurons	Training error ε_{train}	Validation error ε_{valid}
SNRF	2	0.0013266	0.0012672
AIC	2	0.0013266	0.0012672
GL1	5	0.0012875	0.0012756
GL2	6	0.0012653	0.001288
GL3	7	0.0012485	0.0012981
GL5	9	0.0012159	0.0013267
PQ0.5	7	0.0012485	0.0012981
PQ0.75	9	0.0012159	0.0013267
PQ1	10	0.0012011	0.0013348
PQ2	16	0.0011279	0.001352
PQ3	16	0.0011279	0.001352
UP2	5	0.0012875	0.0012756
UP3	7	0.0012485	0.0012981
UP4	9	0.0012159	0.0013267
UP6	14	0.0011433	0.0013401
UP8	18	0.0010993	0.0013702

test error, which outperforms other criteria on the generalization ability.

Using a three-layered 7-2-1 MLP, the results of optimizing number of learning epochs are compared in Table VI.

In another test on Mackey–Glass data, MLP is used to predict the sample $x[t+50]$ from the earlier points $x[t]$, $x[t-6]$, $x[t-12]$, and $x[t-18]$. The number of neurons varies from 1 to 281, with the increment of 20. For a three-layered MLP, SNRF_e becomes lower than the threshold because the number of the hidden neurons is larger than 181. The approximated sequence using such 4-181-1 MLP is compared with the original sequence in Fig. 11. The results from different stopping criteria are compared in Table VII. The SNRF-based OAA yields the smallest validation error, which outperforms others on the generalization ability. For time–series prediction problem, the discontinuous sampling in the second test gives a much more difficult function to approximate comparing to the one in the first test. The $\varepsilon_{\text{train}}$ and $\varepsilon_{\text{valid}}$ fall very slowly after 61 neurons. Due to the

TABLE VI
SIMULATION III: RESULT COMPARISON FOR OPTIMIZING NUMBER OF LEARNING EPOCHS

Stopping Criteria	Optimum number of learning epochs	Training error ε_{train}	Validation error ε_{valid}
SNRF	36	4.9556e-5	6.0117e-5
AIC	N/A	N/A	N/A
GL1	4	0.012513	0.013471
GL2	4	0.012513	0.013471
GL3	4	0.012513	0.013471
GL5	4	0.012513	0.013471
PQ0.5	5	0.032976	0.040495
PQ0.75	5	0.032976	0.040495
PQ1	5	0.032976	0.040495
PQ2	5	0.032976	0.040495
PQ3	5	0.032976	0.040495
UP2	5	0.032976	0.040495
UP3	9	0.00014627	0.00015228
UP4	9	0.00014627	0.00015228
UP6	26	0.00010707	0.00011101
UP8	46	6.1688e-5	6.3639e-5

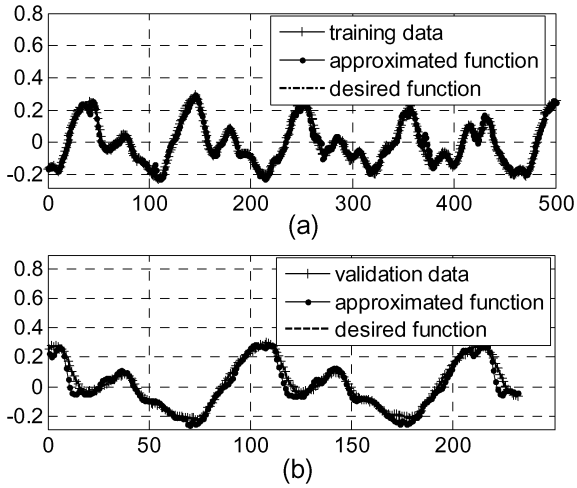

Fig. 11. Approximated Mackey–Glass sequences using 4-18-1 MLP. (a) Training performance. (b) Validation performance.

slow decrease, in order to meet the SNRF stopping criterion, it takes many more hidden neurons in MLP than in the previous test. As shown in Table VII, all other stopping criteria, except for AIC, suggest the same or even higher number of hidden neurons as obtained in this test. The results of optimizing number of learning epochs for 4-181-1 MLP are compared in Table VIII.

Simulation IV: Building Data Set

The building data set [38] is used to predict the hourly consumption of electric energy in a building based on 14 inputs including the time of the day, outside temperature, outside air humidity, solar radiation, wind speed, etc. The data set is subject to an unknown level of noise. It is observed that overfitting will start to occur when a four-layered MLP has more than 19 hidden neurons on each hidden layer. By comparing the first 100 samples from the given sequence with the approximated sequences in Fig. 12(a), it is observed that the approximated sequence basically follows the variation of the electric consumptions. After the given sequence is sorted in the ascending order of electrical energy consumption, and the approximated sequence is

TABLE VII
SIMULATION III: RESULT COMPARISON FOR OPTIMIZING NUMBER OF HIDDEN NEURONS

Stopping Criteria	Optimum number of hidden neurons	Training error ε_{train}	Validation error ε_{valid}
SNRF	181	0.033741	0.11173
AIC	61	0.10435	0.15502
GL1	181	0.033741	0.11173
GL2	261	0.017577	0.13185
GL3	261	0.017577	0.13185
GL5	261	0.017577	0.13185
PQ0.5	281	0.015165	0.14431
PQ0.75	281	0.015165	0.14431
PQ1	Incomplete	Incomplete	Incomplete
PQ2	Incomplete	Incomplete	Incomplete
PQ3	Incomplete	Incomplete	Incomplete
UP2	181	0.033741	0.11173
UP3	181	0.033741	0.11173
UP4	181	0.033741	0.11173
UP6	261	0.017577	0.13185
UP8	Incomplete	Incomplete	Incomplete

TABLE VIII
SIMULATION III: RESULT COMPARISON FOR OPTIMIZING NUMBER OF LEARNING EPOCHS

Stopping Criteria	Optimum number of learning epochs	Training error ε_{train}	Validation error ε_{valid}
SNRF	601	0.019179	0.042483
AIC	N/A	N/A	N/A
GL1	201	0.054362	0.084417
GL2	201	0.054362	0.084417
GL3	201	0.054362	0.084417
GL5	201	0.054362	0.084417
PQ0.5	401	0.031217	0.056719
PQ0.75	401	0.031217	0.056719
PQ1	401	0.031217	0.056719
PQ2	651	0.019212	0.049268
PQ3	651	0.019212	0.049268
UP2	251	0.03988	0.06154
UP3	401	0.031217	0.056719
UP4	401	0.031217	0.056719
UP6	651	0.019212	0.049268
UP8	Incomplete	Incomplete	Incomplete

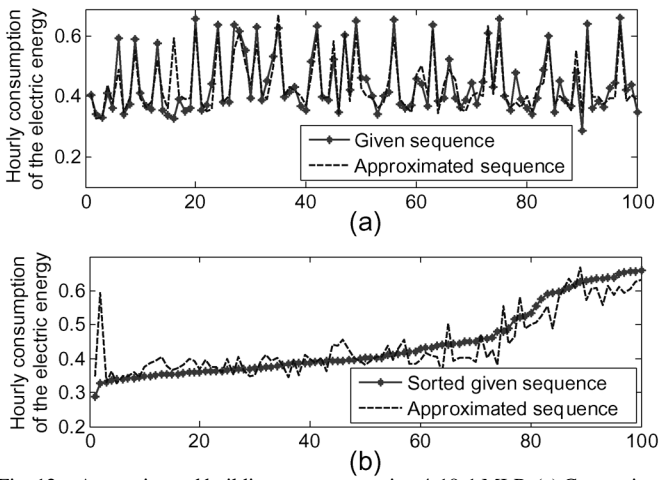

Fig. 12. Approximated building sequences using 4-19-1 MLP. (a) Comparison of the first 100 samples. (b) Sorted sequences.

reordered according to given sequence's order, the correlation between them is clearly observed as shown in Fig. 12(b).

TABLE IX
SIMULATION IV: RESULT COMPARISON FOR OPTIMIZING NUMBER OF HIDDEN NEURONS

Stopping Criteria	Optimum number of hidden neurons	Training error ε_{train}	Validation error ε_{valid}
SNRF	19	0.014129	0.016549
AIC	1	0.023338	0.025003
GL1	7	0.015403	0.02066
GL2	7	0.015403	0.02066
GL3	7	0.015403	0.02066
GL5	13	0.014457	0.018729
PQ0.5	13	0.014457	0.018729
PQ0.75	13	0.014457	0.018729
PQ1	13	0.014457	0.018729
PQ2	31	0.011374	0.020279
PQ3	31	0.011374	0.020279
UP2	22	0.013063	0.017127
UP3	22	0.013063	0.017127
UP4	25	0.012357	0.018323
UP6	40	0.0099095	0.021296
UP8	49	0.0092831	0.022648

TABLE X
SIMULATION IV: RESULT COMPARISON FOR OPTIMIZING NUMBER OF LEARNING EPOCHS

Stopping Criteria	Optimum number of learning epochs	Training error ε_{train}	Validation error ε_{valid}
SNRF	5	0.0067004	0.019159
AIC	N/A	N/A	N/A
GL1	4	0.54801	0.52314
GL2	4	0.54801	0.52314
GL3	4	0.54801	0.52314
GL5	4	0.54801	0.52314
PQ0.5	26	0.0028364	0.035643
PQ0.75	26	0.0028364	0.035643
PQ1	26	0.0028364	0.035643
PQ2	26	0.0028364	0.035643
PQ3	26	0.0028364	0.035643
UP2	7	0.0061748	0.021296
UP3	7	0.0061748	0.021296
UP4	9	0.004131	0.035239
UP6	21	0.0028356	0.052924
UP8	41	0.00075926	0.074638

Optimum numbers of neurons for a four-layered MLP determined by different stopping criteria are compared in Table IX. Subsequently, OAA and other criteria are used to optimize the number of learning epochs for a three-layerd MLP with size 14-19-1, and the results are compared in Table X.

Simulation V: Puma Robot Arm Dynamics Data Set

Another multidimensional benchmark data set OAA is applied to is generated from a simulation of the dynamics of a Unimation Puma 560 robot arm [39]. The task in this problem is to predict the angular acceleration of the robot arm's links from eight inputs including angular positions of three joints, angular velocities of three joints, and torques of two joints of the robot arm. Various numbers of neurons (from one to 100 with a step size of three) are used in the MLP and the optimum number of hidden neurons is determined using OAA. The SNRF_e is compared with threshold, as shown in Fig. 13(a), which indicates that overfitting starts to occur when the number of neurons is 46.

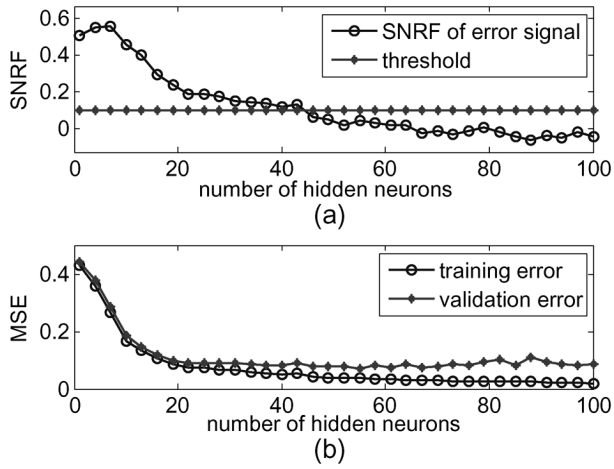

Fig. 13. Simulation V: optimization of number of hidden neurons. (a) SNRF of the error signal and the threshold. (b) Training and validation errors.

TABLE XI
SIMULATION V: RESULT COMPARISON FOR OPTIMIZING NUMBER OF HIDDEN NEURONS

Stopping Criteria	Optimum number of hidden neurons	Training error ε_{train}	Validation error ε_{valid}
SNRF	46	0.043562	0.077474
AIC	1	0.43152	0.44099
GL1	28	0.066768	0.091784
GL2	40	0.049517	0.083847
GL3	40	0.049517	0.083847
GL5	43	0.052393	0.089354
PQ0.5	61	0.034858	0.07574
PQ0.75	73	0.027735	0.084617
PQ1	73	0.027735	0.084617
PQ2	79	0.027958	0.094146
PQ3	85	0.024118	0.082011
UP2	28	0.066768	0.091784
UP3	28	0.066768	0.091784
UP4	28	0.066768	0.091784
UP6	43	0.052393	0.089354
UP8	52	0.038588	0.078167

Note that $\varepsilon_{\text{valid}}$ has many local minima, as seen in Fig. 13(b), and using a local minimum of $\varepsilon_{\text{valid}}$ as a stopping criterion would be ambiguous. The optimization results based on different stopping criterion are compared in Table XI. With an MLP of size 8-46-1, OAA can be used to find proper number of learning epochs and the results are compared with others in Table XII.

In summary, for all tested data sets, the SNRF quantitatively identified overfitting and helped to find the proper structure or the number of training epochs for effective NN learning for a given problem. In the presented simulations, SNRF-based criterion correctly recognizes overfitting, and through analysis of numerical results, we observe that at the obtained optimum point, $\varepsilon_{\text{train}}$ and $\varepsilon_{\text{valid}}$ start to diverge from each other, and $\varepsilon_{\text{valid}}$ reaches its minimum.

In most simulation cases, OAA suggests the minimum structure or minimum length of the training period unlike other stopping criteria. In the few cases it does not, OAA still delivers better generalization in the sense of the smallest $\varepsilon_{\text{valid}}$. In many stopping criteria, variation of $\varepsilon_{\text{valid}}$ is one of the measures used

TABLE XII
SIMULATION V: RESULT COMPARISON FOR OPTIMIZING NUMBER OF LEARNING EPOCHS

Stopping Criteria	Optimum number of learning epochs	Training error ε_{train}	Validation error ε_{valid}
SNRF	4	0.039814	0.077506
AIC	N/A	N/A	N/A
GL1	7	0.032243	0.10845
GL2	7	0.032243	0.10845
GL3	7	0.032243	0.10845
GL5	7	0.032243	0.10845
PQ0.5	11	0.0027257	0.14612
PQ0.75	26	1.0016e-011	0.16009
PQ1	101	1.4087e-018	0.18978
PQ2	101	1.4087e-018	0.18978
PQ3	101	1.4087e-018	0.18978
UP2	7	0.032243	0.10845
UP3	7	0.032243	0.10845
UP4	10	0.0038718	0.11284
UP6	26	1.0016e-011	0.16009
UP8	51	1.3981e-016	0.21248

to determine possibility of overfitting rather than providing quantified evaluation of the goodness-of-fit accomplished by SNRF. To meet the quantified stopping criterion, it may take slightly more hidden neurons or learning epochs for the SNRF to fall below the threshold than in some other criteria. However, in all the cases, the network optimized with OAA outperforms all the other stopping criteria by providing optimized generalization ability.

V. CONCLUSION

In this paper, an optimized approximation algorithm is proposed to solve the problem of overfitting in function approximation applications using NNs. The OAA utilizes a quantitative stopping criterion based on the SNRF. This algorithm can automatically detect overfitting based on the training errors only. The algorithm has been validated for optimization of the number of hidden neurons for MLP and the number of iterations for the BP training. It can be applied to parametric optimization of any learning model or model selection for other function approximation problems.

REFERENCES

[1] S. I. Gallant, "Perceptron-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 1, no. 2, pp. 179–191, Jun. 1990.
[2] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Englewood Cliffs, NJ: Prentice-Hall, 1999.
[3] G. Lorentz, *Approximation of Functions*. New York: Holt, Rinehart and Winston, 1966.
[4] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, pp. 359–366, 1989.
[5] J. M. Zurada, *Introduction to Artificial Neural Systems*. St. Paul, MN: West, 1992.
[6] C. Xiang, S. Ding, and T. Lee, "Geometrical interpretation and architecture selection of MLP," *IEEE Trans. Neural Netw.*, vol. 16, no. 1, pp. 84–96, Jan. 2005.
[7] K. Swingler, *Applying Neural Networks: A Practical Guide*. London, U.K.: Academic, 1996.
[8] M. J. Berry and G. Linoff, *Data Mining Techniques*. New York: Wiley, 1997.
[9] Z. Boger and H. Guterman, "Knowledge extraction from artificial neural network models," in *Proc. IEEE Syst. Man Cybern. Conf.*, Orlando, FL, 1997, pp. 3030–3035.

[10] L. S. Camargo and T. Yoneyama, "Specification of training sets and the number of hidden neurons for multilayer perceptrons," *Neural Comput.*, vol. 13, pp. 2673–2680, 2001.
[11] S. Lawrence, C. L. Giles, and A. C. Tsoi, "What size neural network gives optimal generalization? Convergence properties of backpropagation," Inst. Adv. Comput. Studies, Univ. Maryland, College Park, MD, Tech. Rep. UMIACS-TR-96-22 and CS-TR-3617, Jun. 1996.
[12] S. Lawrence, C. L. Giles, and A. C. Tsoi, "Lessons in neural network training: Overfitting may be harder than expected," in *Proc. 14th Nat. Conf. Artif. Intell.*, 1997, pp. 540–545.
[13] G. Huang and H. A. Babri, "Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions," *IEEE Trans. Neural Netw.*, vol. 9, no. 1, pp. 224–229, Jan. 1998.
[14] G. Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks," *IEEE. Trans. Neural Netw.*, vol. 14, no. 2, pp. 274–281, Mar. 2003.
[15] M. Sartori and P. Antsaklis, "A simple method to derive bounds on the size and to train multi-layer neural networks," *IEEE Trans. Neural Netw.*, vol. 2, no. 4, pp. 467–471, Jul. 1991.
[16] S. Tamura and M. Tateishi, "Capabilities of a four-layered feedforward neural network: Four layers versus three," *IEEE Trans. Neural Netw.*, vol. 8, no. 2, pp. 251–255, Mar. 1997.
[17] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Comput.*, vol. 4, pp. 1–58, 1992.
[18] L. Holmstrom and P. Koistinen, "Using additive noise in back-propagation training," *IEEE Trans. Neural Netw.*, vol. 3, no. 1, pp. 24–38, Jan. 1992.
[19] G. N. Karystinos and D. A. Pados, "On overfitting, generalization, and randomly expanded training sets," *IEEE Trans. Neural Netw.*, vol. 11, no. 5, pp. 1050–1057, Sep. 2000.
[20] C. Wang and J. C. Principe, "Training neural networks with additive noise in the desired signal," *IEEE Trans. Neural Netw.*, vol. 10, no. 6, pp. 1511–1517, Nov. 1999.
[21] E. Alpaydin, "GAL: Networks that grow when they learn and shrink when they forget," Int. Comput. Sci. Inst., Univ. California, Berkeley, CA, Tech. Rep. TR-91-032, 1991.
[22] T. Kwok and D. Yeung, "Constructive algorithms for structure learning in feedforward neural networks for regression problems," *IEEE Trans. Neural Netw.*, vol. 8, no. 3, pp. 630–645, May 1997.
[23] R. Reed, "Pruning algorithms—A survey," *IEEE Trans. Neural Netw.*, vol. 4, no. 5, pp. 740–747, Sep. 1993.
[24] M. R. Frean, "The upstart algorithm: A method for constructing and training feedforward neural networks," *Neural Comput.*, vol. 2, no. 2, pp. 198–209, 1990.
[25] R. Setiono, "Feedforward neural network construction using cross validation," *Neural Comput.*, vol. 13, pp. 2865–2877, 2001.
[26] S. Amari, N. Murata, K. Muller, M. Finke, and H. H. Yang, "Asymptotic statistical theory of overtraining and cross-validation," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 985–996, Sep. 1997.
[27] L. Prechelt, "Automatic early stopping using cross validation: Quantifying the criteria," *Neural Netw.*, vol. 11, no. 4, pp. 761–777, 1998.
[28] S. Wang, J. Judd, and S. S. Venkatesh, "When to stop: On optimal stopping and effective machine size in learning," presented at the Conf. Neural Inf. Process. Syst., Denver, CO, 1993.
[29] E. J. Teoh, K. C. Tan, and C. Xiang, "Estimating the number of hidden neurons in a feedforward network using the singular value decomposition," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1623–1629, Nov. 2006.
[30] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. AC-19, no. 6, pp. 716–723, Dec. 1974.
[31] J. Rissanen, "Stochastic complexity and modeling," *Annu. Statist.*, vol. 14, no. 3, pp. 1080–1100, 1986.
[32] D. B. Forgel, "An information criterion for optimal neural network selection," *IEEE Trans. Neural Netw.*, vol. 2, no. 5, pp. 490–497, Sep. 1991.
[33] N. Murate, S. Yoshizawa, and S. Amari, "Network information criterion-determining the number of hidden units for an artificial neural network model," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 865–872, Nov. 1994.
[34] M. Sugeno and T. Yasukawa, "A fuzzy-logic based approach to qualitative modeling," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 1, pp. 7–31, Feb. 1993.
[35] E. L. Lehmann, *Testing Statistical Hypotheses*. New York: Springer-Verlag, 1997.
[36] D. Erdogmus, O. Fontenla-Romero, J. C. Principe, A. Alonso-Betanzos, and E. Castillo, "Linear-least-squares initialization of multilayer perceptrons through backpropagation of the desired response," *IEEE Trans. Neural Netw.*, vol. 16, no. 2, pp. 325–337, Mar. 2005.
[37] M. C. Mackey and L. Glass, "Oscillations and chaos in physiological control systems," *Science*, vol. 197, pp. 287–289, 1977.
[38] L. Prechelt, "PROBEN1: A set of benchmarks and benchmarking rules for neural network training algorithms," Univ. Karlsruhe, Karlsruhe, Germany, Tech. Rep. 21/94, Sep. 1994 [Online]. Available: ftp://ftp.ira.uka.de/pub/neuron/
[39] University of Toronto, "Delve Datasets," Toronto, ON, Canada [Online]. Available: http://www.cs.toronto.edu/~delve/data/datasets.html

Yinyin Liu received the B.S. degree in information science and engineering from Northeastern University, Shenyang, China, in 2001, and the M.S. degree in aerospace engineering from Old Dominion University, Norfolk, VA, in 2003. Currently, she is working towards the Ph.D. degree at Ohio University, Athens.

Since 2003, she has been working as a Research Associate in Electrical Engineering and Computer Science, Ohio University. Her research interests include machine learning systems, embodied intelligence, and neural networks.

Janusz A. Starzyk (SM'83) received the M.S. degree in applied mathematics and the Ph.D. degree in electrical engineering from Warsaw University of Technology, Warsaw, Poland, in 1971 and 1976, respectively.

From 1977 to 1981, he was an Assistant Professor at the Institute of Electronics Fundamentals, Warsaw University of Technology. From 1981 to 1983, he was a Postdoctorate Fellow and Research Engineer at McMaster University, Hamilton, ON, Canada. In 1983, he joined the Department of Electrical and Computer Engineering, Ohio University, Athens, where he is currently a Professor. He has cooperated with the National Institute of Standards and Technology. He has been a consultant to ATT Bell Laboratories, Sarnoff Research, Sverdrup Technology, Magnolia Broadband, and Magnetek Corporation. In 1991, he was a Visiting Professor at University of Florence, Florence, Italy. He was a Visiting Researcher at Redstone Arsenal–U.S. Army Test, Measurement, and Diagnostic Activity and at Wright Labs-Advanced Systems Research and ATR Technology Development. He is an author or a coauthor of over 170 refereed journal and conference papers. Since 1993, he has been the President of Artificial Neural Systems Incorporated in Ohio. His current research is in the areas of embodied intelligence, sparse hierarchically organized spatio–temporal memories, self-organizing learning machines, and neural networks.

Zhen Zhu (M'06) received the M.S. and Ph.D. degrees in electrical engineering from Ohio University, Athens, in 2002 and 2006, respectively.

Currently, he is a Senior Research Engineer at the Avionics Engineering Center, Ohio University. His research interests include software radio technology, GPS signal processing and receiver design, urban navigation, machine learning systems, and artificial intelligence.