OPTIMIZING NUMBER OF HIDDEN NEURONS IN NEURAL NETWORKS

Yinyin Liu, Janusz A. Starzyk, Zhen Zhu
School of Electrical Engineering and Computer Science
Ohio University
Stocker Center, Athens, OH 45701, USA
starzyk@bobcat.ent.ohiou.edu

ABSTRACT
In this paper, a novel and effective criterion based on the estimation of the signal-to-noise-ratio figure (SNRF) is proposed to optimize the number of hidden neurons in neural networks to avoid overfitting in the function approximation. SNRF can quantitatively measure the useful information left unlearned so that overfitting can be automatically detected from the training error only without use of a separate validation set. It is illustrated by optimizing the number of hidden neurons in a multi-layer perceptron (MLP) using benchmark datasets. The criterion can be further utilized in the optimization of other parameters of neural networks when overfitting needs to be considered.

KEY WORDS
Neural network, network optimization, function approximation, overfitting, signal-to-noise ratio figure

1. Introduction

Neural networks (NN) have been shown to be universal approximators [1] [2] [3] and are widely used in function approximation. Optimizing the number of hidden neurons to use without a pre-set target for accuracy is one of the major challenges for neural networks, usually referred to as the bias/variance dilemma [4]. Using excessive hidden neurons will cause overfitting, which means that the neural networks over-estimate the complexity of the target problem. It greatly degrades generalization capability, which leads to significant deviation in predictions. In this sense, determining the proper number of hidden neurons to prevent overfitting is critical in function approximation using NN. There are various approaches to build the network in a constructive or destructive way, but the common methods to determine whether a certain number of hidden neurons is optimal are cross-validation and early-stopping [5][6]. In these methods, the available data are divided into two independent sets: a training set and a validation or testing set. Only the training set participates in the neural network learning, and the testing set is used to compute testing error, which approximates the generalization error. The performance of a function approximation during training and testing is measured respectively by training error ε_{train} and testing error ε_{test}. Once the testing performance stops improving as the number of hidden neurons continues to increase, it is possible that the training has begun to fit the noise in the training data, and overfitting occurs. Therefore, during this process, the stopping criterion is set so that, when the testing set error ε_{test} starts to increase, or equivalently when training error ε_{train} and testing error ε_{test} start to diverge, optimal value of the number of hidden neurons is assumed to have been reached.

However, in cross-validation and early stopping, the use of the stopping criterion based on ε_{test} is not straightforward and requires definite answers to several questions. For example, how does one determine the size of the training and testing sets in predicting generalization error using ε_{test} [7]? Although ε_{test} provides an estimate of generalization error, is it necessary that the generalization error increases as soon as ε_{test} begins to increase? As ε_{test} varies with the number of basis functions, and it has many local minima, which of these local minima indicates the occurrence of overfitting? [5][8] Both methods require omission of the testing set in the training stage, which is a significant waste of the precious available data. It is desired to have the ability to examine the training error without using a testing set and automatically recognize the occurrence of overfitting. In this paper, a signal-to-noise ratio figure (SNRF) is defined to measure the goodness of fit using ε_{train} and to develop a novel stopping criterion in constructively building NN.

The organization of this paper is as follows. In Section 2, the definition and estimation method of SNRF is presented and the SNRF-based criterion is developed. This criterion is validated using simulation on real benchmark datasets and compared with cross-validation results in Section 3. Finally, features of the proposed stopping criterion are discussed in Section 4.

2. Estimation of signal-to-noise ratio figure

Without *a priori* knowledge of the noise characteristics of the training data, it is assumed that the training data may come with White Gaussian Noise (WGN) at an unknown level. 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, comes from two possible sources: the approximation error due to the limited learning ability or inaccuracy in approximation using the given number of hidden neurons in the network, and the WGN in the training data. A critical question is whether there is still useful signal information left to be learned in the error signal that produces the approximation error. If there is, based on the assumption that the 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 signal energy level over the noise energy level is defined as SNRF. The SNRF can be pre-calculated for the WGN. The comparison of SNRF of the error signal with that of WGN determines whether WGN dominates in the error signal. If it does, there is little useful information left in the error signal, and the approximation error cannot be reduced anymore.

Therefore, in the process of optimizing number of hidden neurons of NN, one may start with a network with a small number of hidden neurons and keep adding hidden neurons until SNRF indicates overfitting. The estimation of SNRF will be first illustrated using a one-dimensional function approximation problem, followed by a general discussion of multi-dimensional problems.

2.1. SNRF estimation for a one-dimensional function approximation

Assume that in a one-dimensional function approximation problem, training data are uniformly sampled from the input space $X \subset \Re^1$ with additive noise at an unknown level. An approximation $\hat{F}$ is obtained using NN with a certain number of hidden neurons. The error signal e contains a noise component denoted by n, and an approximation error component, which is the useful signal left unlearned and therefore denoted by s.

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

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 the signal and noise components.

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

This energy can be approximated 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 that 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 the approximated function 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 circularly shifted version of the s. Due to the nature of WGN, noise of a sample is independent of noise on 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 circularly shifted replica of n_{i-1}. Since the noise component is independent of the signal component, the correlation of e_i with its shifted copy e_{i-1} approximates the signal energy, as shown in (6).

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

The difference between the autocorrelation with no time shift in (3) and $C(e_i, e_{i-1})$ in (6) 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 signal level to noise level, defined as the SNRF of the error signal, is obtained:

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

In order to detect the existence of useful signal in e, the SNRF of e has to be compared with SNRF of WGN estimated using the same number of samples. When there is no signal in e, the SNRF of WGN is calculated as,

$$\begin{aligned} SNRF_{WGN} &= \frac{C(n_i, n_{i-1})}{C(n_i, n_i) - C(n_i, n_{i-1})} \\ &= \frac{C(m_i, m_{i-1})}{C(m_i, m_i) - C(m_i, m_{i-1})} \end{aligned}. \tag{9}$$

It is observed that the $SNRF_{WGN}$ is in fact independent of the noise level β, which means that estimation on WGN with unit standard deviation can obtain the general characterization of $SNRF_{WGN}$. $SNRF_{WGN}$ represents the signal to noise ratio estimated using a limited number of samples, which is a random value related to the number of samples N. Its average value $\mu_{SNRF\_WGN}(N)$ can be obtained from,

$$\mu_{SNRF\_WGN}(N) = \mu\left[\frac{C(m_i, m_{i-1})}{C(m_i, m_i) - C(m_i, m_{i-1})}\right] \tag{10}$$

Since $C(m_i, m_i) >> C(m_i, m_{i-1})$,

$$\mu_{SNRF\_WGN}(N) \approx \mu\left[\frac{C(m_i, m_{i-1})}{C(m_i, m_i)}\right] \approx \frac{\mu[C(m_i, m_{i-1})]}{N} = 0. \tag{11}$$

The standard deviation $\sigma_{SNRF\_WGN}(N)$ can be estimated using Monte-Carlo simulation. In Fig. 1, $\sigma_{SNRF\_WGN}(N)$ from a 10000-run Monte-Carlo simulation is shown in the logarithmic scale. It can be approximated using a linear function of the number of samples N, also shown in Fig. 1. The linear fit is done to fit better for larger values of N, considering the sample numbers available in real-world training datasets. The linear fit is then converted back into the linear scale, and the $\sigma_{SNRF\_WGN}(N)$ can be found using

$$\sigma_{SNRF\_WGN}(N) = \frac{1}{\sqrt{N}}. \tag{12}$$

Since the samples of $SNRF_{WGN}$ are statistically independent, by the central limit theorem, if N is large enough, the distributions of $SNRF_{WGN}$ are Gaussian with mean μ_{SNRF_WGN} and standard deviation σ_{SNRF_WGN}.

The stopping criterion 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 one must stop increasing number of hidden neurons to avoid overfitting.

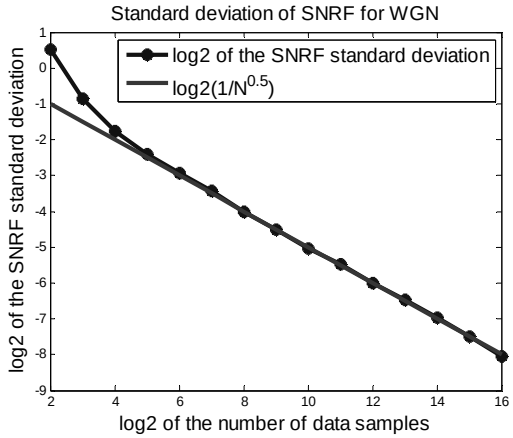


Fig. 1. Standard deviation of SNRF for WGN in one-dimensional case

It is obtained from statistical simulation that the p = 5% significance level [9] can be approximated by the average value plus 1.7 times standard deviations for an arbitrary N. Hence, the stopping criterion can be defined as a $SNRF_e$ smaller than the threshold determined by (13):

$$th_{SNRF\_WGN}(N) = \mu_{SNRF\_WGN}(N) + 1.7\sigma_{SNRF\_WGN}(N). \quad (13)$$

Notice such threshold can achieve 5% significance level for Gaussian process, which proves that the distributions of $SNRF_{WGN}$ are Gaussian with mean μ_{SNRF_WGN} and standard deviation σ_{SNRF_WGN}.

The above discussion have been developed based on the assumption that e could be treated as a signal with evenly-spaced samples. In a general one-dimensional function approximation problem, the input samples may be unevenly spaced. Yet, E_{s+n} , E_s and E_n can still be roughly approximated using (3), (6) and (7) respectively. Thus, the $SNRF_e$ can be estimated using (8) and the overfitting is determined by comparison of $SNRF_e$ with the threshold in (13).

2.2. SNRF estimation for multi-dimensional function approximation

In a multi-dimensional function approximation problem, the training data are usually randomly sampled from the input space $X \subset \Re^M$. They are often not evenly spaced and the distances among samples are random. The method used to estimate SNRF in the one-dimensional case cannot be directly applied to the multi-dimensional problem. However, we could still assume that variation of s in each coordinate is slow compared to the average sampling distance. Thus the same principle of signal and noise level estimation using correlation may be used.

The signal level at each sample e_p (p=1, 2,...N) can be estimated through the correlation with its nearest neighbours. Since s changes slowly in all directions, the continuous function can be locally approximated around e_p with an M-dimensional hyper-plane. Such a hyper-plane can be uniquely determined using M+1 points, including e_p and its M neighbors with the shortest Euclidean distances. Correlation of e_p with these M points provides an estimate of the signal level at e_p. If an error signal contains only the signal component, the points in the neighbourhood are expected to have correlated values. On the contrary, the values of points from WGN are uncorrelated. Therefore, by calculating the correlation, the signal and noise levels within the neighborhood of e_p can be estimated. The signal level at e_p is computed using a weighted combination of the products of e_p values with each of its neighbors, e_{pi} (i=1, 2,...M). The combination can be simplified by averaging the M products, if the distances between e_p and e_{pi} (i=1, 2,...M) are equal. This, however, cannot be guaranteed in a multi-dimensional problem. Since the samples of e_i are assumed to be spatially correlated, the distance between each two samples affects the level of correlation between them, and the distances can be used to calculate weight values. In an M-dimensional space, the weights are obtained based on the Euclidean distance to the power of M, and normalized, as given by (14), where d_{pi} represents the Euclidean distance between e_p and e_{pi}.

$$d_{pi} = \left| e_p - e_{pi} \right|, \; w_{pi} = \frac{\frac{1}{d_{pi}^M}}{\sum_{i=1}^{M} \frac{1}{d_{pi}^M}} \begin{pmatrix} p = 1,2,...N \\ i = 1,2,...N \end{pmatrix} \quad (14)$$

Thus, the signal level at e_p is estimated by the normalized weighted sum as

$$E_{sp} = \sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi} \;\; (p = 1,2,...N), \quad (15)$$

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

$$E_s = \sum_{p=1}^{N} E_{sp}. \quad (16)$$

Again, as in (3), the autocorrelation of e_i estimates signal plus noise level:

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

Then the noise level is estimated using

$$E_n = E_{s+n} - E_s = \sum_{i=1}^{N} e_i^2 - \sum_{p=1}^{N} \sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi}. \quad (18)$$

Finally, the $SNRF_e$ in an M-dimensional input space is computed as

$$SNRF_e = \frac{E_s}{E_n} = \frac{\sum_{p=1}^{N}\sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi}}{\sum_{i=1}^{N} e_i^2 - \sum_{p=1}^{N}\sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi}} . \quad (19)$$

Notice that when applied to one-dimensional cases ($M=1$), (19) is identical to (8), the $SNRF_e$ model particularly derived for the one-dimensional case.

When there is no signal, $SNRF_{WGN}$ is estimated using (19) 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. Although the values of w_{pi} in (14) cannot be predicted for the general characterization of $SNRF_{WGN}$, which makes it difficult to evaluate its standard deviation, the upper and lower bounds of $\sigma\left[\sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi}\right]$ can be determined as,

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

The lower bound is reached when the w_{pi} have equal values (i.e., uniform sampling distance). It gets closer to the upper bound in problems with large dimensionality. For a dataset with large dimensionality and non-uniformly spaced samples, the lower bound would be hard to reach. Therefore, the detection threshold can be calculated using the upper bound, which tends to be more sensitive to the occurrence of overfitting, with a greater possibility of a premature stop.

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_1 and p_2 are the closest neighbors to each others, $e_p \cdot e_{pi}$ is calculated twice. The estimation of $\sigma\left[\sum_{p=1}^{N}\sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi}\right]$ is thus different from the calculation based on the summation of independent random variables. In the worst case, all the terms in $\sum_{p=1}^{N}\sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi}$ may appear twice, which can be used to estimate the upper bound of standard deviation for the numerator of (19):

$$\sigma\left[\sum_{p=1}^{N}\sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi}\right] \leq \sqrt{2N} , \quad (21)$$

from which we have,

$$\sigma_{SNRF\_WGN}(N) \leq \frac{\sqrt{2}}{\sqrt{N}} . \quad (22)$$

Also, the average of $SNRF_{WGN}$ is estimated as,

$$\mu_{SNRF\_WGN}(N) = \mu\left[\frac{\sum_{p=1}^{N}\sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi}}{\sum_{i=1}^{N} e_i^2 - \sum_{p=1}^{N}\sum_{i=1}^{M} w_{pi} \cdot e_p \cdot e_{pi}}\right] \approx 0 \quad (23)$$

Notice that the estimation of $\sigma_{SNRF\_WGN}(N)$ and $\mu_{SNRF\_WGN}(N)$ using (22) and (23) has been simplified and is no longer a function of the problem dimensionality. Such simplification makes it more convenient to construct a detection threshold universal to any number of dimensions.

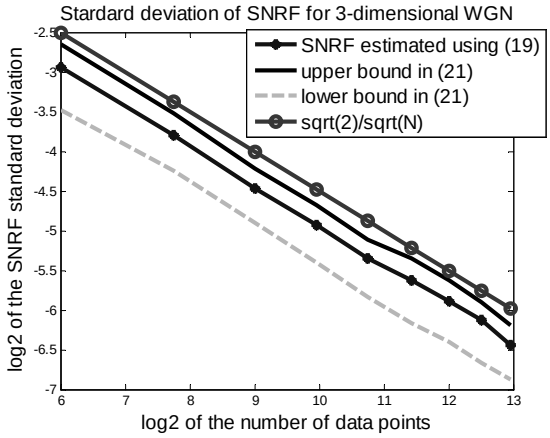


Fig. 2. Standard deviation of SNRF of WGN in a three-dimensional case

$\sigma_{SNRF\_WGN}(N)$ for a three-dimensional WGN from a 1000-run Monte-Carlo simulation is shown in the logarithmic scale in Fig. 2. The estimated $\sigma_{SNRF\_WGN}(N)$ in (22) is consistent with an upper bound of $\frac{\sqrt{2}}{\sqrt{N}}$, and the bounds developed in (20) are validated. Consider not all the samples are independent, central limit theorem does not apply and the distribution of $SNRF_{WGN}$ is not Gaussian. The threshold can be experimentally established as the average value plus 1.2 times the standard deviation, to achieve the 5% significance level, as in (24).

$$th_{SNRF\_WGN}(N) = \mu_{SNRF\_WGN}(N) + 1.2 \cdot \sigma_{SNRF\_WGN}(N) \quad (24)$$

The threshold calculation defined in (24) also can be applied to the one-dimensional case and is very close to the approximation of one-dimensional functions modeled in Section 2.1.

In summary, using SNRF, we can estimate the signal level and noise level for the error signal and then quantitatively determine the amount of useful signal information left unlearned. When there is no information left, the learning process with certain number of hidden neurons can be stopped, and the optimum approximation has been obtained without overfitting. Otherwise, more hidden neurons have to be used to improve the learning. Although we assumed that the approximated signal may be noisy, the method works equally well when there is no noise in the input data. The noise characteristics simply serve as a reference for developing the stopping criterion.

3. Simulation and discussion

The proposed SNRF-based criterion on optimizing the number of hidden neurons was tested on a 4-layer MLP in the following simulations. The 4-layer MLP contains an input layer, 2 hidden layers (with equal number of hidden neurons) with nonlinear transfer functions, and an output layer with linear transfer functions. The training algorithm is the linear least-squares learning method [10] so that the number of iterations does not affect the learning performance.

As the number of hidden neurons increases, the amount of signal information left in the error signal decreases, and the $SNRF_e$ is expected to decrease as well. When the $SNRF_e$ becomes lower than the threshold as more neurons are added, the error signal shows the characteristic of the WGN and overfitting starts to occur. At this point, one should stop increasing the size of the hidden layer, and the optimized approximation without overfitting using MLP is obtained. Meanwhile, the cross-validation method is applied to the same dataset, which utilizes the training error ε_{train} and the testing error ε_{test} measured in the normalized MSE [7]. It is expected that when the SNRF-based criterion recognizes overfitting, either ε_{train} and ε_{test} will start to diverge from each other, or ε_{test} will reach a minimum. This minimum is employed in the cross-validation method as the "early-stopping" point [7]. As stated in Section 1, the recognition of overfitting by ε_{test} is not accurate because ε_{test} is only an approximation of the generalization error and has local minima as parameters of the network vary. Hence, the cross-validation results helps to prove the effectiveness of the SNRF-based criterion, and the obtained error signals are shown to demonstrate the quality of fit.

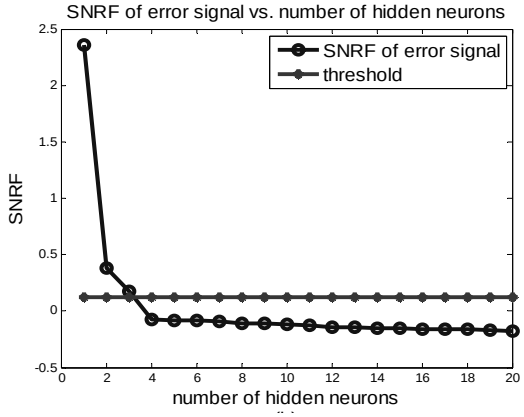
3.1. Mackey-glass dataset

The Mackey-glass data is a time series data set obtained from a physiological system [11] with an unknown level of noise. In this test, MLP is used to predict each 8th sample based on every 7 samples, assuming that every 8th sample in the time series is a function of previous 7 samples. The target function and the error signal, *e,* are a continuous one-dimensional, time-domain functions. This allows us to use the one-dimensional SNRF and threshold measures to determine the optimum number of hidden neurons.

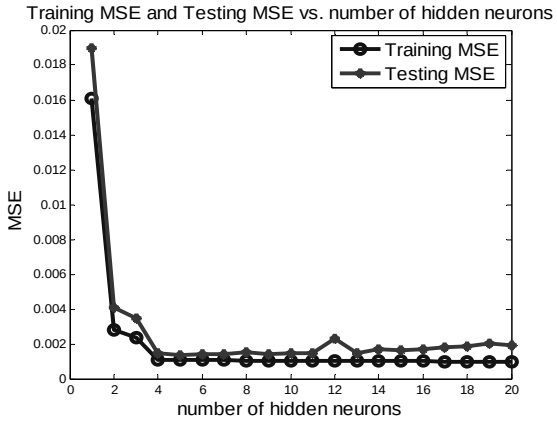
It is discovered that $SNRF_e$ becomes lower than the threshold with the number of the hidden neurons larger than or equal to 4, as can be seen in Fig.3(a). Fig. 3(b) shows that ε_{test} starts to increase and diverge from ε_{train} after the number of hidden neurons exceeds 4, as expected. The approximated sequence, using 4 hidden neurons in the MLP, almost overlaps the desired target function. The approximation error signal obtained shows the characteristic of WGN as shown in Fig.4.

3.2. Puma robot arm dynamics dataset

The dataset generated from a simulation of the dynamics of a Unimation Puma 560 robot arm [12] has 8 inputs include angular positions of 3 joints, angular velocities of 3 joints and torques of 2 joints of the robot arm. The task in this problem is to predict the angular acceleration of the robot arm's links. The dataset is subject to an unknown level of noise. In order to find the optimum number of neurons, various numbers of neurons (from 1 to 100 with a step size of 3) are used in the MLP and the optimum number of hidden neurons is determined using SNRF-based criterion.



(a) SNRF of the error signal and threshold



(b) Training and testing error

Fig. 3. Optimization of number of hidden neurons in Mackey-glass case

Randomly choose 300 samples to use as the training set and another 200 samples to use as the testing set. The $SNRF_e$ is compared with threshold, as shown in Fig. 5(a) which indicates that overfitting starts to occur when the number of neurons is 40. Note that ε_{test} has many local minima, as seen in Fig. 5(b), and using a local minimum of ε_{test} as a stopping criterion would be ambiguous in this case. Using a 6th order polynomial fit to ε_{test}, we can see that the testing error starts to diverge from training

error at 40 neurons, which is very close to the prediction from SNRF based criterion.

In summary, the proposed SNRF-based criterion is tested and verified on one-dimensional and multi-dimensional datasets. It proves that the SNRF is able to identify the overfitting from the training errors only.

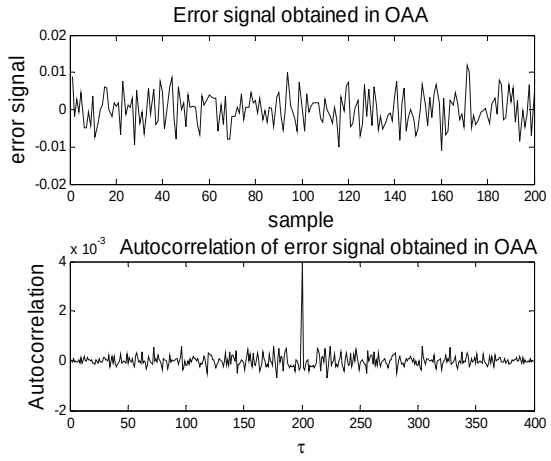
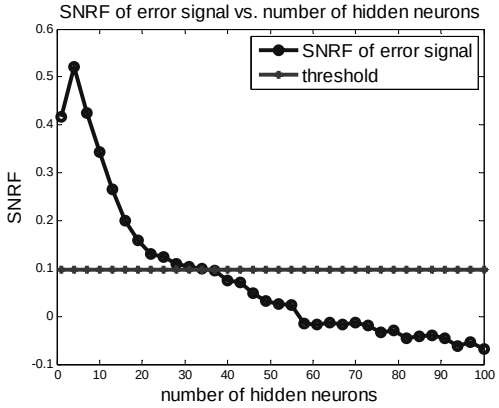
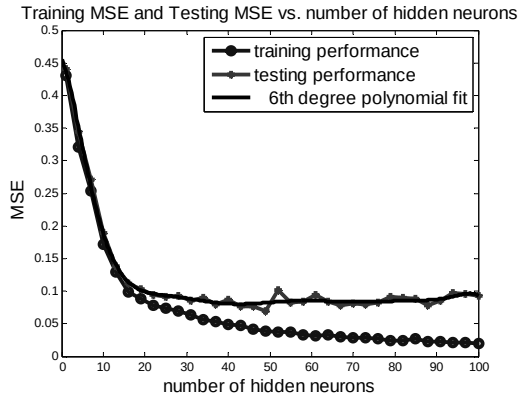


Fig. 4. Performance and error signal obtained in Mackey-glass case



(a) SNRF of the error signal and threshold



(b) Training and testing error

Fig. 5. Optimization of number of hidden neurons in Puma Robot case

4. Conclusions

In this paper, a method is proposed to optimize the number of hidden neurons in NN to avoid overfitting in function approximation. The method utilizes a quantitative criterion based on the SNRF to detect overfitting automatically using the training error only, and it does not require a separate validation or testing set. The criterion has been validated using benchmark datasets and compared with the common cross-validation method. The criterion is very easy to apply, consumes small amount of computations and is suitable for practical application. The same principle applies to the optimization of other parameters of neural networks, including the number of iterations in back propagation training to avoid overtraining or the number of hidden layers. It can be applied to parametric optimization or model selection for other function approximation problems as well.

References

[1]. S. I. Gallant, Perceptron-based learning algorithms, *IEEE Trans. on Neural Networks*, *1(2)*, 1990, 179-191.

[2]. K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, *Neural Networks*, *2(5)*, 1989, 359-366.

[3]. J. M. Zurada, *Introduction to artificial neural systems* (West Publishing Company, 1992).

[4]. S. Geman, E. Bienenstock, R. Doursat, Neural networks and the bias/variance dilemma, *Neural Computation*, *4(1)*, 1992, 1-58.

[5]. L. Prechelt. Early stopping -- but when, *Neural Networks: Tricks of the trade*, *Lecture Notes in Computer Science 1524* (Springer Verlag, Heidelberg, 1998).

[6]. R. Setiono, Feedforward neural network construction using cross validation, *Neural Computation*, *13(12)*, 2001, 2865–2877.

[7]. S. Amari, N. Murata, K. Muller, M. Finke, H. H. Yang, Asymptotic statistical theory of overtraining and cross-validation, *IEEE Trans. On Neural Networks*, *8(5)*, 1997, 985-996.

[8]. C. Wang, S. J. Judd, S. S. Venkatesh, When to stop: on optimal stopping and effective machine size in learning, *Proc. Conf. on Neural Information Processing Systems*, Colorado, 1993.

[9]. E. L. Lehmann, *Testing statistical hypotheses* (New York, Springer, 1997).

[10]. D. Erdogmus et al. Linear-least-squares initialization of multilayer perceptrons through backpropagation of the desired response, *IEEE Trans. On Neural Networks, 16(2)*, 2005, 325-337.

[11]. M. C. Mackey, L. Glass, Oscillations and chaos in physiological control systems, *Science*, *197(4300),* 1977, 287-289.

[12]. http://www.cs.toronto.edu/~delve/data/datasets.html