Minimum Description Length Criterion for Modeling of Chaotic Attractors With Multilayer Perceptron Networks

Zhao Yi and Michael Small, Member, IEEE

Abstract—Overfitting has long been recognized as a problem endemic to models with a large number of parameters. The usual method of avoiding this problem in neural networks is to avoid fitting the data too precisely, and this technique cannot determine the exact model size directly. In this paper, we describe an alternative, information theoretic criterion to determine the number of neurons in the optimal model. When applied to the time series prediction problem we find that models which minimize the description length (DL) of the data, both generalize well and accurately capture the underlying dynamics. We illustrate our method with several computational and experimental examples.

Index Terms—Backpropagation neural networks, description length (DL), false nearest neighbors (FNN), model size, nonlinear curvefitting.

I. INTRODUCTION

The biological motivation for artificial neural networks is a massive highly connected array of nonlinear excitatory “neurons.” It is natural to build artificial neural networks with a fairly large number of neurons. This creates a statistically ill-posed problem as the number of model parameters can easily exceed the number of available data. Therefore, it is very easy for the resultant model to overfit. However, the critical issue in developing a neural network is generalizability of that network: the network should make small prediction errors on the training set and also respond properly to novel inputs.

Elementary discussion of overfitting was represented in [1]. Geman, Bienenstock, and Doursat illustrated a rigorous approach based on the bias/variance tradeoff to improve network training, and underfitting produces excessive bias in the outputs, whereas overfitting produces excessive variance [2].

The most common methods to avoid statistical overfitting are: early stopping [3] and statistical regularization techniques, such as weights decaying [4] and Bayesian learning [5]. For early stopping, there is one limitation: we must be careful not to choose a training algorithm that converges too rapidly. With this method, the choice of the validation set is also difficult. The validation set should be representative of all points in the training set. The weights decaying method involves modifying the performance function to

\[\text{mse}_{\text{reg}} = \text{mse} + (1 - \mu)\text{msw}\]

where \(\text{mse}\) is the mean sum of squares of the network errors, \(\text{msw}\) is the mean sum of squares of the network weights and biases, and \(\mu\) is the performance ratio. The problem with this method is that it is difficult to select the optimum value for \(\mu\). If \(\mu\) is too large, the networks may overfit; if \(\mu\) is too small, the network will not adequately fit the training set.

For Bayesian learning, the optimal regularization parameter, \(\mu\) is determined in an automated fashion [5]. Another feature of Bayesian learning is that it provides a measure of how many network parameters are effectively used by the network, i.e., how many parameters are not required or have little significance to the networks. Hence, using this method we usually build a larger network with many “wasteful” parameters. The smaller optimal network cannot be determined by this method.

A common feature of all these methods is that they optimize parameters of the known neural network but they cannot determine the optimal number of neurons (model size) of the network for a specific application. In this paper, we take an alternative approach, which estimates exactly how many neurons of the feedforward multilayer neural network are required for time series prediction. We note that when mentioning neural networks used in our experiments we mean that they are feedforward multilayer neural networks.

The criterion is an approximation to the minimum description length (MDL) [6]. The MDL principle rooted in the theory of algorithmic complexity [7] and mainly developed by Rissanen with a series of papers starting with [8], who proposed the issue of model selection as a problem in data compression. Several modifications to the MDL for improving its performance have been proposed [9]–[12] and [13], [14].

There are other typical methods applicable to model selection. Akaike proposed an information criterion (AIC) based on a weighted function of the fit of a maximum log-likelihood model and the number of independent parameters adjusted to maximize the likelihood [15]. The motivation of AIC and its assumptions was the subject of some discussion in [8], [16], and [17]. From a practical point, the AIC tends to overfit the data [18], [19]. Wallace et al. developed the minimum message length (MML) [20]. As in the MDL, MML chooses the hypothesis minimizing the code-length of the data but the codes are quite different from those in MDL. The Bayesian information criterion (BIC), also known as Schwarz’s information criterion (SIC) is

Manuscript received January 2, 2005; revised April 26, 2005. This work was supported in part by Hong Kong Polytechnic University under Research Grant A-PE46 and by the Hong Kong University Grant Council under Grant G-YW55. This paper was recommended by Associate Editor C.-T. Lin.

The authors are with the Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: zhao.yi@polyu.edu.hk).

Digital Object Identifier 10.1109/TCSI.2005.858321
equivalent to the MDL criterion [21], [17], [8]. Recently, Xu developed the Bayesian Ying Yang (BYY) information criterion for model selection [22]–[24]. It is clear that MDL criterion is related to other well-known model selection criterion. The AIC and BIC perform best for linear models; for nonlinear models, description length (DL) style information criteria are better. Our choice of the MDL, in particular, is based on our familiarity with it and the fact that it is robust and relatively easy to estimate. The MDL has the great advantage of relatively small computational costs.

There have been relatively few attempts at applying description length to model selection of neural networks for time series prediction. Judd and Mees have successfully applied DL to selection for radial basis function networks [13]. More recently, Small and Tse generalized this result to neural network architectures [14]. Our current work is a significant generalization of [14] in the three major areas.

1) Each neural network in [14] is built by the expansion of the previous (smaller) model, i.e., the larger new neural network is formed by adding one new candidate neuron (basis function) to the old model. In the current work all the neural networks are built independently, which guarantee that there is no any connection among them. Although it is never possible to find the global MDL, by allowing models of different sizes to be built independently of one another, MDL is achieved in a broader search of parameter space. In [14] models are built by successively expanding the previous (smaller) model: in doing so, one searches a much more restricted area of parameter space.

2) In this paper, we count in the contribution to the DL of not only linear parameters of the neural network, but also the nonlinear parameters of it. The penalty for model parameters \(M(k)\) then involves a variable representing the effective number of parameters within one neuron. However, in [14] the authors ignore the nonlinear parameters in a neuron, and, hence, \(M(k)\) is only a rough approximation.

3) The specific training algorithm used in [14] is borrowed from statistical approximation theory and restricted to those models (there is no flexibility in choosing fitting techniques). Therefore, it is not possible to systematically compare the model performance to standard neural network techniques. Networks in this paper can be implemented with the existing training algorithms, such as basic gradient descent, conjugate gradient algorithm, and Levenberg-Marquardt algorithm. The MDL criterion is applicable to select the model size of neural networks trained by the existing training algorithms. So it provides a general methodology for model selection. In other words the current MDL criterion is independent of training algorithms and robust. This allows us, for the first time, to present a fair comprehension of MDL, neural networks, and standard neural network fitting techniques.

In Section II, we review the MDL criterion and modeling algorithms we employ. Section III presents a systematic study of the performance of this method both with computational and experimental time series data.

II. MDL Criterior

If observed data are independent and identically distributed random numbers then there is no good predictive model, and the most compact description of this data is simply to describe the observations themselves. If observed data are (for example) a realization of some autoregressive process, then the best (most compact) description of these data is to describe the initial few observations, the autoregressive structure, and the further stochastic perturbations (that is, the model prediction errors). Finally, if the observed data represent a deterministic system (the chaotic Ikeda map for example) then the most compact description of these data is to describe the initial values, the parameters of the Ikeda map, and small corrections. Note that for chaotic systems and certainly most real world data, the perfect model cannot make perfect predictions indefinitely, therefore, we must admit the necessity of model prediction errors.

A. How to Compute DL

The basic principle of MDL is to estimate both the cost of specifying the model parameters and the associated model prediction errors. Let \(z\) be data of the model; \(\Lambda\) represents the vector of all the model parameters. In [13] the DL of the data with respect to a particular model \(\Lambda\), is expressed as

\[
L(z, \Lambda) = L(z|\Lambda) + I(\Lambda),
\]

(1)

\(L(z|\Lambda)\) represents the DL of model prediction errors; \(I(\Lambda)\) is the description length of the model. Let \(E(k)\) be the cost of describing the model prediction errors and \(M(k)\) be the cost of describing the model parameters. The DL of the data with respect to this model is then given by the sum\(^1\)

\[
D(k) = M(k) + E(k),
\]

(2)

Intuitively, typical behavior of \(E(k)\) and \(M(k)\) is that if the model size \(k\) increases \(E(k)\) increases and \(M(k)\) decreases. The MDL principle states that the optimal model is the one for which \(D(k)\) is minimal.

Let \(\{y_i\}_{i=1}^N\) be a time series of \(N\) measurements; let \(f(y_{i-1}, y_{i-2}, \ldots, y_{i-d}; \Lambda_k)\) be a scalar function of \(d\) variables that is completely described by the \(k\) parameters \(\Lambda_k = (\lambda_1, \lambda_2, \ldots, \lambda_k)\). Define the prediction error \(e_i\) by \(e_i = f(y_{i-1}, y_{i-2}, \ldots, y_{i-d}; \Lambda_k) - y_i\). For any \(\Lambda_k\) the DL of the model \(f(\cdot; \Lambda_k)\) is given by the DL of the \(k\) parameters \(\Lambda_k\) [14]

\[
M(k) = L(\Lambda_k) = \sum_{i=1}^k \ln \frac{\gamma}{\delta_j}
\]

(3)

where \(\gamma\) is a constant and represents the number of bits required in the exponent of the floating point representation. For example, \(\gamma = 32\) is more than adequate for nearly all purposes, and smaller values can be chosen if desired [13]. In (3), \(\delta_j\) is

\(^1\)Calculation of \(M(k)\) and \(E(k)\) depends on the specific encoding selected for the model and for rational numbers. We use the optimal encoding of floating point numbers described by Rissanen [6].
interpreted as the optimal precision of the parameters $\lambda_j$ and $(\delta_0, \delta_1, \ldots, \delta_k)$ are defined as the solution of

$$Q = D_{\lambda_0} \lambda_k E(k).$$

where $Q$ is the second derivative of $E(k)$ with respect to the model parameters $\lambda_k$, and $(\cdot)_j$ denotes the $j$th element of the vector $(\cdot)$ [13].

Rissanen [6] has shown that $E(k)$ is the negative logarithm of the likelihood of the errors $e = \{e_t\}_{t=1}^N$ under the assumed distribution of those errors

$$E(k) = -\ln \text{Prob}(e|\lambda_k)$$

where

$$\text{Prob}(e|\lambda_k) = \frac{1}{\sqrt{2\pi}^N} \exp \left( -\frac{e^T e}{2N} \right).$$

For the general case of an unknown distribution of errors, the situation is rather complicated. One has to calculate the deviation of the errors $\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} e_i^2$. However, assuming that the model errors $e$ are Gaussian distributed with mean zero and standard deviation $\sigma$ we get

$$E(k) = \frac{N}{2} + \ln \left( \frac{2\pi}{N} \right)^{N/2} + \ln \left( \sum_{i=1}^{N} e_i^2 \right)^{N/2}.$$  

In principle we may compute DL by solving (4) to yield the precision $\delta_j$, substituting it to (3) and (7) to calculate the DL of the model $M(k)$, and calculate the DL of the model prediction errors $E(k)$. Note that for large $k$ a computational bottleneck results from ensuring that the matrix (5) yields a solution to (4) [14]. Substituting (7) to (5) we obtain

$$Q = D_{\lambda_0} \lambda_k \left( \frac{N}{2} + \ln \left( \frac{2\pi}{N} \right)^{N/2} + \ln \left( \sum_{i=1}^{N} e_i^2 \right)^{N/2} \right)$$

$$= D_{\lambda_0} \lambda_k \left( \ln \left( \sum_{i=1}^{N} e_i^2 \right)^{N/2} \right)$$

$$= D_{\lambda_0} \lambda_k \left( \frac{N}{2} \ln(e^T e) \right)$$

$$= \frac{N}{2} D_{\lambda_0} \lambda_k (\ln((f(y; \lambda_k) - y)^T(f(y; \lambda_k) - y)))$$

where $y = \{y_t\}_{t=1}^N$.

The above equation reveals that if $f(y; \lambda_k)$ is a linear function, the computation of $Q$ is straightforward, but when it is nonlinear the computation of $Q$ becomes considerably more complex and the solution of (4) becomes substantially more difficult. In [25] Small and Judd found that this full nonlinear approximation to DL was realizable (for radial basis models), but it was also rather slow, and provided marginal benefit.

In this paper, we restrict our interest to multilayer neural networks with a single hidden layer, sigmoid activation functions, and linear output. For other neural network structures, the treatment is equivalent. Hence, for inputs $(x_{t-1}, x_{t-2}, \ldots, x_{t-d})$ the model $f$ can be mathematically expressed as

$$f(x_{t-1}, x_{t-2}, \ldots, x_{t-d}) = b_0 + \sum_{i=1}^{k} v_i \phi \left( \sum_{j=1}^{d} \omega_{ij} x_{t-j} + b_i \right)$$

where $v_i, \omega_{ij}, b_i \in \mathbb{R}$, $k$ is the model size and $d$ is the number of inputs. The model parameters $\Lambda_k$ are, therefore, $\{b_0, b_k, v_i, \omega_{ij} | i = 1, \ldots, kj = 1, \ldots, d\}$. Of these parameters, the weights $v_i$ and the offset $b_0$ are all linear, the remaining parameters $\omega_{ij}$ and $b_i(i = 1, \ldots, k, j = 1, \ldots d)$ are nonlinear. For multilayer networks, $\phi$ is usually selected to be a bounded monotonically increasing function. We choose the tan-sigmoid transfer function $\phi(x) = 1 - e^{2x}/1 + e^{2x}$.

The function $\phi(x)$ is approximately linear in the region of interest, so we suppose that the precision of the nonlinear parameters is similar to that of the linear one in this region. We thus employ the linear parameters, the weights $v_i$, to calculate the precision of the model $\delta_j$, which makes it computationally feasible to solve (4) and (5).

One may often expect that not all the parameters are significant for a particular neuron and some of parameters are effectively “irrelevant” to the DL calculation. To account for the contribution of all linear and nonlinear parameters to $M(k)$ we define $n_p(i)$ as the effective number of the parameters of the $i$th neuron contributed to the DL of the neural network. In [14] for the $i$th neuron $n_p(i)$ is just equal to 1 since this neuron contains one linear parameter and a few of nonlinear parameters. They merely calculated the contribution of the linear parameter to $M(k)$.

Thus, we have

$$M(k) = \sum_{i=1}^{k} n_p(i) \ln \frac{\gamma}{\delta_i}$$

where $\delta_j$ is the relative precision of the weight $v_i \in \{v_1, v_2, \ldots, v_k\}$.

In general, $n_p(i)$ will be a variable with respect to different neurons but in order to make the problem tractable we make one further approximation. We consider that $n_p(i)$ is fixed for all $i$ and then replace $n_p(i)$ with $n_p$. For (9) there exists one parameter $n_p$, which make the following

$$n_p \sum_{i=1}^{k} \ln \frac{\gamma}{\delta_i} = \sum_{i=1}^{k} n_p(i) \ln \frac{\gamma}{\delta_i}.$$  

However, the exact value of $n_p$ is difficult to calculate and we, therefore, approximate $n_p(i)$ by using the embedding dimension estimated with false nearest neighbors (FNN) [26].
Thus, we have

\[ M(k) \approx \hat{h}_p \sum_{i=1}^{k} \ln \frac{\gamma}{\delta_i}, \quad (10) \]

In (3), we take \( \hat{h}_p = 1 \), but for our neural networks \( \hat{h}_p \in (1, d+2) \). The motivation for this approximation is described in the following section.

This approximation is necessary to make the calculation of DL computationally feasible. We have not found a way to analytically evaluate the accuracy of the approximation. Probably it is not possible to evaluate the accuracy of this approximation in general. For specific cases, the accuracy of this approximation can be quantified indirectly by evaluating the performance of the resultant models.

For nonlinear optimization problems (such as fitting a nonlinear model to time series data), it is not possible to guarantee a global optimal solution without performing an exhaustive search. For a continuous (and, in fact, fairly high-dimensional) parameter space, one can only ever achieve a local model. We cannot consider that the models estimated are necessarily the best especially comparing with other more complicated models, but we can conclude that this technique can estimate the optimal one from the available neural networks, and provide improved modeling results.

B. FNN

For a deterministic system, we can establish a phase space for the system such that specifying a point in this space specifies the state of the system, and vice versa. We, therefore, have to convert the data into phase vectors. This is the problem of phase space reconstruction [27] which is technically obtained by the delay time and embedding dimension. Vectors in the embedding space are formed from time delayed values of the scalar measurements \( y(n) \) [26]:

\[ y(n) = (x(n), x(n + \tau), \ldots, x(n + (m - 1)\tau)) \]

where \( m \) is called the embedding dimension, \( \tau \) is the delay time. Note that in this paper for the purpose of modeling the lag of the input vector \( \tau \) is equal to one and the number of input vectors \( m \) is approximately the length of one orbit (i.e., pseudo period). We also expect that \( m >> d_e \), the embedding dimension.

Embedding dimension \( d_e \) represents the dimension of the phase space required to specify a point in that phase space, i.e., it represents the effective number of inputs for a model. According to (8) the value of the parameters in each neuron, i.e., the effective number of parameters, is closely related to the effective number of inputs. So the main point to consider here is that we try to determine how many coordinates of a high-dimensional embedding space provide significant useful information.

FNN is widely used to determine the number of degrees of freedom required to unambiguously unfold the dynamics, i.e., the FNN embedding dimension ensures that trajectories in the embedded phase space do not cross. Conversely, an embedding dimension larger than that suggested by FNN introduces additional redundancy: that is, trajectories that should be close are also sparsely distributed. When we consider the model building problem it has the same requirement that the parametric representation of coordinates in phase space should be sufficient to distinguish between points that represent disparate trajectories, but should still allow similar trajectories to be close to one another.

Several authors have found that good representation of dynamics depends crucially on appropriate embedding and reconstruction: one simply cannot build a good model without considering the embedding dimension at the same time [28]. Alternatively, one found that the best embedding for the purpose of modeling depends on the construction of the model [29]. So using FNN to choose \( \hat{h}_p \) is equivalent to making the correct choice of embedding reconstruction parameters \( (d_e) \) of the modeling problem.

According to the preceding discussion the minimum embedding dimension represents the maximal effective number of model parameters which is defined as \( \hat{h}_p \), in Section II-A.

Kenne et al. [26] described the idea of the FNN algorithm as follows. Each point \( \vec{R}_i = (x(n), x(n + \tau), \ldots, x(n + (m - 1)\tau)) \) in the time series looks for its \( r \)-th nearest neighbor \( \vec{R}_j \) in a \( m \)-dimensional space. Calculate their distance

\[ R^2_{m}(n, r) = \sum_{k=0}^{m-1} [x_i(n + k\tau) - x_j(n + k\tau)]^2. \]

When embedding dimension increases from \( m \) to \( m + 1 \), one calculates the distance between \( \vec{R}_i \) and the same \( r \)-th nearest neighbors

\[ R^2_{m+1}(n, r) = R^2_{m}(n, r) + [x_i(n + m\tau) - x_j(n + m\tau)]^2. \]

The criterion for designating a neighbor as false is that

\[ \left( \frac{R^2_{m+1}(n, r) - R^2_{m}(n, r)}{R^2_{m}(n, r)} \right)^{1/2} = \left| \frac{x_i(n + m\tau) - x_j(n + m\tau)}{R_{m}(n, r)} \right| > R_{tol} \]

where \( R_{tol} \) is a given heuristic threshold [26].

For \( R_{tol} = 5 \), the FNN are clearly identified in our application. The determination of which neighbors are true and which are false is usually quite insensitive to the threshold one uses once the number of data is sufficient to nicely populate the attractor [30].

According to the usual criterion, one decides how many points can be counted as FNN. The percentage of FNN decreases with increasing embedding dimension. If one applies FNN to clean data from a chaotic system, one usually expects that the percentage of FNN will drop from nearly 100% in dimension one to strictly zero when \( m \) is reached. Furthermore, it will remain zero from then on, since once the attractor is unfolded, it remains unfolded [30]. If the signal is contaminated by noise, the noise may be sufficient to always produce some FNN. So, the percentage of FNN will not reach zero, but remain approximately stable after the minimal embedding dimension.

III. APPLICATIONS AND EXAMPLES

In this section, we present the application of this algorithm to two test systems (Section III-A) and two experimental data
sets (Section III-B). The test systems we select are the Rössler system and the Ikeda map, both with the addition of dynamic noise. We then describe the application of this method to experimental recordings of the chaotic laser data utilized in the 1992 Santa Fe time series competition [31] and electrocardiogram (ECG) data [32], [33].

When we calculate DL for each example we met a problem that DL curves fluctuate dramatically and it is very likely to affect our estimate of the minimum point. Since the different neural networks are constructed independently in the current work the perturbations in structural parameters (e.g., associated with estimation noise) of each network are not related to that of successive networks. This correspondingly results in fluctuation of DL curves but the potential changing tendency estimated by DL curves still exists. The curve fitting procedure is intended to smooth this noise and provide a more accurate estimate of the actual minimum. Our idea to apply nonlinear fitting can be explained in the following three steps.

1) We first define a function, which takes a value $k$ (the number of neurons) and parameter vector to estimate DL. In (7) since $N$ is the number of measurements the first and second terms are constant, and the form of $E(k)$ is consistent with that of a decreasing exponent function. So we use $ae^{-bk}(a > 0, b > 0)$ to describe $E(k)$. Referring to (10), $n_p$ and $\gamma$ are constant; $\delta_i$ is the relative precision of the weights of the $i$th neuron, and for different neurons it is close to a constant. So $M(k)$ can be regarded appropriately as the linear function about the number of neurons $k$. Thus define $ck + d(c > 0)$ to describe $M(k)$. Actually $d$ is also required to compensate for an arbitrary constant which is missing from the computation of DL in (9) and (10). So the defined function is

$$F(a, b, c, d; k) = ae^{-bk} + ck + d \quad (11)$$

where $k$ is the number of neurons; $a, b, c,$ and $d$ are the required coefficients.

The above function is the empirical approximation of the true tendency estimated by the DL curve.

2) According to the least-square sense, $\min_{a,b,c,d} \sum_{i=1}^{k} (F(a,b,c,d; i) - D(i))^2$, we solve (11) and obtain the vector $(a, b, c, d)$.

3) Introduce $a, b, c, d$, which is achieved in step (II), and determine the solution of $k$ which make (11) minimal.

Thus the fitted DL curve provides a smooth estimation of DL and aims to better catch the true minimum. When expending the additional effort of nonlinear curvefitting we find that the estimate of $k$ is much more robust, especially for complicated noisy experimental data.

A. Computational Experiments

1) The Rössler System: The first computational simulation we consider is a reconstruction of the Rössler system with dynamic noise. The equations of the Rössler system are [34]

$$\begin{align*}
x(t) &= -y(t) - z(t) \\
y(t) &= x(t) + ax(t) \\
z(t) &= b + z(t) * [x(t) - c]
\end{align*} \quad (12)$$

Fig. 1. DL (solid line) of dynamic noise data of the Rössler system (top panel) has the minimum point at five (i.e., five neurons), and the fitted curve (broken line) attains the minimum at the same point. In the bottom panel solid line is mse of training set and dotted line is that of testing data.

with parameters $a = 0.15, b = 0.20, c = 10.00$, and we take the sampling time $t_s$ to be 0.5 s.

By dynamic noise we mean that system noise is added to the $x$-component data prior to prediction of the succeeding state at 0.5-s intervals, i.e., we integrated the ODE (12) with the step size 0.5 and then use the integrated results as initial data for the next step. The standard deviation of the noise is set at 9% of the standard deviation of the data. We generate 2000 points of this system of which 1600 points are selected as a training set for the neural network and the rest are used as testing data. We calculate DL of every network constructed with from 1 to 20 neurons, i.e., $D(1), D(2), \ldots, D(20))$. Results are shown in Fig. 1 with mse of these networks.

Referring to Fig. 1(a) we find both the DL curve and fitted curve attain the same minimum point, which denotes that for this application the optimal number of neurons estimated is five. Note that mse of the testing set in Fig. 1(b) starts increasing at the fourth point. This reflects the fact that for the large networks
the errors between testing data and the prediction of those become large gradually and these networks tend to overfit.

We choose four networks with different numbers of neurons to perform free-run prediction for the testing set. The $x$-component data is converted into three vectors, $x(t), x(t + 3)$, and $x(t + 5)(t \in [1, 396])$ to construct 3-D figures shown in Fig. 2.

The network that consists of five neurons can capture the dynamics of the Rössler system very well but with more neurons the network is apt to overfit.

2) The Ikeda Map: The second computational simulation is the reconstruction of the Ikeda map with dynamic noise. We add the dynamic noise to the Ikeda Map in the same way. The standard deviation of the noise is set at 30% of the standard deviation of the data. We generate 1000 points of this system, of which 600 points are selected as a training set for the neural network and the rest are used as testing data. Fig. 3 presents DL and mse of 600 observations of Ikeda training set and 400 observations of testing data from one to twenty neurons.

Although the original DL curve in Fig. 3(a) attains the minimum, it fluctuates somewhat dramatically, which we do not expect. However, the fitted curve is smooth and attains the same minimum point as the DL curve. Furthermore, we find this result is also in agreement with the content of Fig. 3(b). Mean square error of testing set in Fig. 3(b) begins to increase at the fourth point. This means that the neural network with more than four neurons exhibits larger error on testing data and, therefore, overfits the data.

In this experiment, we choose the data of $x$-component data to execute free-run prediction for the testing set. Fig. 4 presents four Ikeda maps, of which $x$-column and $y$-column represent two vectors $x(t)$ and $x(t + 1)(t \in [1, 396])$, respectively.

We can observe the network that is made up of three neurons can predict the best results among the four Ikeda maps. The attractor in Fig. 4(b) is almost identical to that of the true Ikeda map. Whereas with fewer neurons, the network cannot fit either the training set or testing set well, and with more neurons the...
neural network is apt to overfit: the attractors in Fig. 4(d) are extremely "noisy." So, DL can estimate the correct number of neurons of the neural network (the optimal network) for the reconstruction of the Ikeda map.

3) **Comparative Simulation of the Rössler System and Ikeda Map:** In the preceding subsections we demonstrated that the method of DL can determine the optimal networks for two specific applications. However, we were not sure whether alternative methods such as a Bayesian learning algorithm or early stopping can train the networks to attain good or even better results. To address this question we consider further experiments in which we adopt these standard methods to avoid the overfitting of the selected networks for the same applications.

As mentioned in Section I, the Bayesian learning algorithm provides a measure of how many network parameters (weights and biases) are being effectively used by the network. This effective number should remain approximately the same, no matter how large the total number of parameters in the network becomes. This assumes that the network has been trained for a sufficient number of iterations to ensure convergence [35]. So we apply the Bayesian learning algorithm to train large networks whose neurons are between eighteen and thirty. For the Rössler system, we utilize the same 1600 point training set and 400 point testing set; for the Ikeda map the same 600 point training set and 400 point testing set are used. We also ensure sufficient iterations during training neural networks. Typical results are shown in Fig. 5.

Referring to these figures, we can observe that for either the Ikeda map or Rössler system the large networks can capture the basic underlying dynamics. Without Bayesian learning algorithm these large networks are very likely to overfit. We conclude that the Bayesian learning algorithm improves the generalization of the networks. However, comparing with Fig. 2(b) and Fig. 4(b) the optimal networks estimated by DL can capture the dynamics more precisely for these two applications, which indicates that the optimal networks possess better generalizability than the networks trained with Bayesian learning algorithm.

For the method of early stopping, we also repeat the experiments and construct the attractors of the free-run prediction, but these attractors are considerably poorer. One of the primary reasons is that we do not know how to choose the validation set which should be representative of all points in the training set. Based on our experiences, correct choice of the validation set is vital but very difficult to achieve.

In the following, we apply this implementation of the information theoretic method to two kinds of experimental data.

**B. Experimental Data**

Experimental data, such as of the chaotic laser data and ECG data that are natural phenomena are more complicated than the above computational data. These experimental data are more difficult to predict and model accurately. So we apply this information theoretic method to data from two practical systems: experimental recordings of the chaotic laser data utilized in the 1992 Santa Fe time series competition and data of ECG measurements, both of which are the focus of considerable attempts to model dynamics.
1) Chaotic Laser Data: We select the 1000 data points, of which 600 points are used as training set and the rest are the testing data. DL and mse of every network constructed with from 1 to 20 neurons are shown in Fig. 6.

We note that mse of testing data still start to increase at certain point (the fourth point), but comparing to the former two mse pictures in Fig. 1(b) and Fig. 3(b) the increasing trend is not very obvious. Fig. 7 presents free-run predictions of 400 points of the laser data predicted by networks with 4, 7, 9, and 15 neurons.

Based on a qualitative comparison of free-run dynamics we found that the network with seven neurons can capture the underlying dynamics of chaotic laser data best, but the network with four neurons fails to predict the testing data. Networks with more neurons are inclined to overfit for the new data. Consequently, the information theoretic method can decide the correct number of neurons for the first experimental data.

It is not surprising that the network with nine neurons can obtain relatively good prediction, because DL curves can be viewed as a guide for a range to select neurons, including the minimum. It suggests that networks with neurons in this range can provide good fitting for the data with high probability, and the network with the number of neurons corresponding to the minimum of the DL curve (fitted curve) can provide adequate generalization and capture the dynamics of time series most exactly.

2) ECG Data: It has been shown that ECG measurement during induced ventricular fibrillation (VF) in pigs is consistent with a chaotic dynamical system [36]. ECG data (during sinus rhythm) was collected by a unique data collection facility established in the coronary care unit (CCU) of the Royal Infirmary of Edinburgh [32], [33].

From 3000 data points, we utilize 2550 points to build networks and the other 450 to test them. Note that among the previous three simulation examples, every FNN curve will drop to zero and remain at zero, but for ECG data the process is similar to that of contaminated signal (i.e., it does not reach zero). So,
the original ECG data is preprocessed by a low-pass Chebyshev Type II filter with normalized cutoff frequency 40 Hz in order to remove any noise at 50 Hz, and we then apply FNN to calculate $\hat{h}_p$. Fig. 8 describes DL and mse for this application.

The DL curve suggests the optimal number of neurons is 16. But the fitted curve estimates that the optimal number of neurons is eight. And in this experiment the mse of testing data cannot help us decide which networks are most susceptible to overfitting. So we deliberately select free-run prediction of the network with eight neurons and the networks with 16 neurons, 3 neurons, and 22 neurons for comparison, as shown in Fig. 9.

Networks with eight neurons can predict the testing data best, but the network with 16 neurons predicts poorly. So we deduce that the optimal number of neurons is eight, i.e., the network with eight neurons can provide adequate fitting in this experiment, and networks with more neurons, such as 16, overfit. Although referring to the DL line we may select the wrong optimal number of neurons, the fitted curve can capture the true tendency of the DL and correctly estimates the optimal number of neurons. So application of both DL and nonlinear curvefitting in selecting the model size of neural networks can estimate the correct number of neurons.

In the experiment of the laser chaotic data, we found that the DL figure actually estimates a range for selecting neurons, not only the minimum. Actually, we also observe that the networks with seven and nine neurons are likely to obtain good results.

IV. CONCLUSION

The information theoretic approach we described in this paper provides a very useful method to decide the model size

Fig. 7. Chaotic laser dynamics. Free-run prediction (solid line) and actual data (broken line) for four networks composed of 4, 7, 9, and 15 neurons, which are labeled as a, b, c, and d in Fig. 6(a).

Fig. 8. DL (solid line) and the fitted curve (broken line) of ECG data (top panel) estimate the minimal point at the sixteenth and eighth point, respectively. Solid line is the mse of training set with mse of testing data (dotted line) in the bottom panel.
networks. This algorithm estimates the number of neurons for a specific application in terms of the MDL. When calculating DL we find that the DL curves fluctuate somewhat dramatically, preventing an accurate estimation of the true DL. We adopt nonlinear curvefitting to catch true DL curves. The fitted curves always estimate the correct MDL.

We have demonstrated the application of this algorithm to four typical nonlinear time series. We note that DL can directly estimate optimal numbers of neurons for the Ikeda map and the Rössler system, as does nonlinear curvefitting. For chaotic laser data and ECG data, DL with the help of nonlinear curvefitting can estimate optimal numbers of neurons. In all experiments, networks with this optimal number of neurons can provide adequate fitting and capture the dynamics very well.

Regularization techniques optimize the parameters of the known neural network to improve the generalization of this neural network. They do not determine how many neurons of the neural network are sufficient to a specific application. However, the MDL criteria can be utilized to determine the optimal network size directly. We find that the optimal model estimated by the criterion consists of a small number of neurons. Despite this, such models generalize well (have good prediction error performance on new data) and also capture the underlying dynamics (the deterministic attractor of the model is the same, or similar, to that of the data). Furthermore, we observe that such models actually perform better than larger models built by using standard methods.

Note that the models discussed in the paper are time-invariant (i.e., stationary) systems. For time-varying systems, the unknown time-varying parameter may affect model building in an arbitrary way, and can, therefore, be modeled with an arbitrary number of models. If the instantaneous values of the time-varying parameter are known, then it can be added as an additional model parameter (an exogenous input). Modeling such processes with MDL type model selection process is possible [37].

Finally, we note that in this paper we adopt the Levenberg–Marquardt algorithm [38], which is a very general learning function. When applied to neural networks trained with different training algorithms DL will estimate different optimal model sizes. This is easily understood. The same four experiments described in the paper have been repeated to estimate model size of the neural networks with different training algorithms. We find the MDL criterion can also select optimal numbers of neurons (i.e., optimal networks) for the preceding four application. Consequently, this algorithm appears to be robust for neural networks whose training algorithms are different.

REFERENCES


Zhao Yi was born in Weihai, Shandong Province, China, in 1977. He received the B.E. and M.S. degrees, both in electronic engineering, from the China Institute of Metrology and Zhaijiang University, China, in 1999 and 2003, respectively. He is currently working toward the Ph.D. degree at Hong Kong Polytechnic University.

Since 2003, he worked as a Research Assistant with the Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hong Kong. His research interests include nonlinear system modeling, nonlinear dynamics, neural networks, and nonlinear analysis in biomedicine.

Michael Small (M’01) received the B.Sc. (Hons.) degree in pure mathematics and the Ph.D. degree in applied mathematics from the University of Western Australia, in 1994 and 1998, respectively.

Since graduating he was with the University of Western Australia, Stellenbosch University, South Africa, and Heriot-Watt University, Edinburgh, U.K. He has been with the Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hong Kong, China, since 2000 and is currently an Associate Professor. His research interests are nonlinear dynamics and chaos, nonlinear time series analysis, and nonlinear modeling. His recent work has been on the application of nonlinear mathematical methods to a diverse range of problems, including, infant respiratory patterns, cardiac arrhythmia, pulse diagnosis, digital communication systems, modeling disease epidemics, and financial time series analysis.