5

The Chemical Oxygen Demand Modelling Based on a Dynamic Structure Neural Network

Junfei Qiao, Qili Chen and Honggui Han
Beijing University of Technology, China

1. Introduction

Wastewater treatment process aims at achieve the purpose of purification by degradation of organic matter in water. To ensure the effluent water quality, some indicators should be measured, including chemical oxygen demand (COD), Biochemical oxygen demand (BOD), etc. Through the prediction on effluent indicators can provide effective guidance for the operation of wastewater treatment plant.

Wastewater treatment process itself is a nonlinear, time-delay process with complex reactions. Thus, when using traditional mathematical model there is often a lack of accuracy, large amount of calculation and lack of flexibility in system simulation, while the prediction based on neural network model can effectively eliminate these disadvantages because of its learning mechanism. Nowadays, applying neural network in wastewater treatment process has become a research hotpot, and some breakthroughs were achieved in terms of algorithms or modelling.

zhu, et al. used MLP model to reduce the data dimension, then used the time-delay neural network to predict the effluent BOD online. Chang, et al. reduced data dimension through principal component analysis(PCA), then used extracted system inherent characteristics from data by fuzzy C clustering, at last, TSK-type fuzzy inference system was used to predict the effluent COD. Chai et al. proposed a activated sludge process mechanism model based on hierarchical neural network, connecting the mechanism model and neural network in cascade way and with the neural network identifying the reaction rate of nonlinear components in activated sludge process model to predict the effluent COD.

The above evidences show that artificial neural networks can directly establish the model according to the input / output data without prior knowledge of the condition object, and has strong online correction ability. For the process with a large amount of data information which can not be described by rules or formulas, the artificial neural network shows great flexibility and adaptability which is ideal for wastewater treatment systems. However, these network models have the same shortcomings that the network structure would no longer able to modify after finalized in the early stage of designing. For the different cases of wastewater treatment process, the re-design of neural network prediction model is necessary. To solve this problem, meet the needs of the object by dynamically adjusting the neural network structure is an available approach.

Huang et al. proposed a simple sequential learning algorithm called the "RBF growing and pruning algorithm" (GAP-RBF), which was later developed into the GGAP-RBF algorithm.

The GAP-RBF and GGAP-RBF methods use the "significance" of a hidden node to decide whether to add new nodes or reduce the number of redundant nodes. The "significance" of the node is also linked to the learning accuracy. However, since these algorithms are on-line procedures, they do not optimize the network over all past training data. Moreover, network initialization of the GAP-RBF algorithm requires a priori knowledge of the data which may not be available.

This chapter presented a repair algorithm for the design of a Radial Basis Function (RBF) neural network. The proposed repair RBF (RRBF) algorithm starts from a single prototype randomly initialized in the feature space. The algorithm has two main phases: an architecture learning phase and a parameter adjustment phase. The architecture learning phase uses a repair strategy based on a sensitivity analysis (SA) of the network's output to judge when and where hidden nodes should be added to the network. New nodes are added to repair the architecture when the prototype does not meet the requirements. The parameter adjustment phase uses an adjustment strategy where the capabilities of the network are improved by modifying all the weights. The algorithm is shown to be effective by approximating a non-linear function, so it is used to model the key parameter, chemical oxygen demand (COD) in the waste water treatment process. The results of simulation show that the algorithm provides an efficient solution to problems.

The chapter is organized as follows. Section2 introduces the methods and problems of the modelling the key parameter, chemical oxygen demand (COD) in the waste water treatment process, and gives the methods of measuring COD. Section3 gives a short description of the RBF neural network and the SA output model; briefly analyses the repair method which is used to add hidden nodes and describes the algorithm which adjusts the parameters; the proposed method is benchmarked against some well-known dynamic RBF algorithms. In order to demonstrate the superior performance of the proposed RRBF neural network, the algorithm is applied to approximating a nonlinear function. Section4 finishes the Soft measurement technique for COD in the waste water treatment process. The conclusion and Future work are given in Sec. 5.

2. Wastewater treatment process

2.1 The problem in COD measurement

Wastewater treatment plants are complex nonlinear systems, subject to large disturbances, where different physical (such as settling) and biological phenomena are taking place. Many models have been proposed in the literature for wastewater treatment process but their evaluation and comparison are difficult. To ensure the good condition and effluent quality during wastewater treatment operation process, the key parameters should be obtained in time. On the one hand, wastewater treatment aims on reducing the environmental pollution, which requires detecting the effluent COD, BOD, TN, TP etc. according to related national effluent standard; on the other hand, the normal operation and control implementation of each wastewater treatment link depends on real-time detection of controlled variables. Among these effluent parameters, COD, which indirectly represents the water organic pollution degree by DO consumption through microbiology metabolism, is an important index accords with the practical self-purification situation and the routes of most wastewater treatment processes. Therefore, the detection of COD is significant while some

inevitable defects exist by using conventional approaches like COD on-line analyzer: the process of COD is complicated, the time spent on detecting will greatly lags to the operating time in practice and the results can't reflect the real situation in time [1]; COD on-line analyzer is too expensive to extend. Thus, as a cost-effective tool for replacing expensive on-line sensor, the research on soft-sensing became active since 1990s. Combining soft-sensing with neural network to predict key parameters and guide, neural networks are wildly applied in process modelling and prediction. Chang, et al. reduced data dimension through principal component analysis(PCA), then used extracted system inherent characteristics from data by fuzzy C clustering, at last, TSK-type fuzzy inference system was used to predict the effluent COD. In this section, the soft-sensing technology applied in wastewater treatment process will be introduced.

2.2 Soft-sensing technology

Soft sensors have been reported to supplement online instrument measurements for process monitoring and control. Both model-based and data-driven soft sensors have been developed (B. Lin et al., 2007). As solution for the above problems, the core of soft-sensing is mathematical modelling for the objects. To obtain optimal estimation for primary variables and set up a soft-sensing model which is suitable for wastewater treatment process, selecting appropriate instrumental variables according to wastewater treatment characteristics is a must. Principal component analysis can reduce the dimension and correlation of process variables (W. Ran et al., 2004), data preprocessing can obtain the correct data. So the design steps for soft-sensing are as follows:

a. Data preprocessing

Influenced by precision and reliability of measuring instrument and measuring environment, the measuring errors are inevitable. Firstly, the ones with different magnitude from others were deleted. Then the measuring samples should be handled by data normalization using the following:

$$p_{ij}^{*} = \frac{p_{ij} - \overline{p}_j}{s_{jj}} \tag{1}$$

Where i is the number of samples, j is the component of samples, p_{ij} is the j th component of the i th sample, $\overline{p}_j$ is the mean of the j th component of the samples, S_{jj} is the standard error of the variable p_j.

b. Principal component analysis

Wastewater treatment process is complicated, contains many variables. And there exists quite linear correlation among those measuring variables and data.

Create a matrix $p = [p_1, p_2, \cdots, p_m]$, which composed of process variables and divided by columns, to calculate its covariance matrix S. The characteristic roots of S are listed as $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_m \geq 0$, of which the corresponding unit orthogonal Eigenvectors are composed of matrix $L = [L_1, L_2, \cdots, L_m]$ (Load Matrix) Dividing x into the exterior product of principal components' sub-matrix T and Load Matrix L, added residual term E, as follows:

$$x = TL^T + E = T_1 L_1{}^T + T_2 L_2{}^T + \cdots + T_n L_n{}^T + E$$

Then calculate m (minimum required number of principal components) which make cumulative variance contribution rate $\eta = \sum_{i=1}^{n} \lambda_i / \sum_{j=1}^{m} \lambda_j > 90\%$, select relevant n principal components in T. Under the precondition $\eta > 90\%$, four instrumental variables which mostly influenced effluent COD were chosen by analyzing the relation between principal components and instrumental variables, they are SS, pH, oil and NH_3-N.

c. Establishment of soft-sensing model

The well-known mathematical modelling and neural networks methods have limitations to incorporate the key process characteristics at the wastewater treatment plants which are complex, non-stationary, temporal correlation, and nonlinear systems (M. H. Kim et al., 2009). To build soft-sensing model of key water quality parameters, some systematic methods of neural networks modelling. Wang, W. (W. Wang & M. Ren, 2002) used BP neural network to predict BOD and COD etc. key parameters by modelling wastewater process; Guclu, D (D. Guclu & S. Dursun, 2008) used an artificial neural network to implement the prediction of effluent COD concentrations. The common characteristic of these techniques is that the feed forward neural network was used to model the wastewater treatment process (J. B. Zhu et al., 1998). Used MLP model to reduce the data dimension, then used the time-delay neural network to predict the effluent BOD online. Chai et al. (T.Y. Chai et al., 2009) proposed a activated sludge process mechanism model based on hierarchical neural network, connecting the mechanism model and neural network in cascade way and with the neural network identifying the reaction rate of nonlinear components in activated sludge process model to predict the effluent COD. Chen et al. (Q. L. Chen et al., 2010) proposed a recurrent neural network model to identifying the BOD by modelling wastewater process.

Researches show that artificial neural network is able to model the wastewater treatment process. However, wastewater treatment process is a highly nonlinear and state-varying dynamic process; the network's dynamic performance will vary according to different network structure because of the single and immutable mapping ability of fixed-structure neural network. Therefore, a self-constructing neural network (J. F. Qiao & H. G. Hang, 2010) which combined RBF neural networks and principal component analysis technology will be presented in section 3. The result of principal component analysis efficiently included in the key modelling information of the wastewater treatment process. The above four variables are used as the input of RBF model while effluent COD as the output, then we choose the proper number of neurons in the hidden layer and train the network by learning algorithm. This neural network would adjust network structure according to the complexity of wastewater treatment process, which would significantly improve the soft sensor's performance.

3. A Self-constructing RBF neural network

This section will introduce a self-constructing RBF neural network which based on a repair algorithm. Cell replacement therapy is emerging as a novel method for restoration of the defective tissues by repairing the inactive cells. This innovative strategy has attracted considerable attention to the human embryonic stem cell recently (Mathur A et al., 2004). Similarly, it is well known that the connecting cells of the biological networks are repaired in the neural systems (Noriaki Suetake & Eiji Uchino, 2007). A Radial Basis Function (RBF)

neural network is a simple neural system model, which is often applied to machine learning problems, because it can approximate non-linear mappings directly from input patterns (S.S. Panda et al., 2008; Xabier Barandiaran & Alvaro Moreno, 2008). In theory, all RBF network topologies should be able to learn any given task to some level of competency. In reality, however, a given topology can be both a bottleneck and a constraint on a system. If the size of the network is chosen incorrectly it becomes moribund, rendering the results meaning less (A. Esposito et al., 2000; Wang, X. X. et al., 2004). Consequently a lot of research has focused on the difficult problem of determining the optimal size of a RBF neural network (S. Chen et al., 1991 & 1996). The size of the input and output layers in a RBF neural network is fixed – only the size of the hidden layer can be modified. Two main methods, "growing" and "pruning", have been developed to dynamically change the size of a network. Esposito et al. (A. Esposito et al., 2000) have proposed a growing method based on an evolutionary optimization strategy. However, this method has a number of drawbacks: it requires a large amount of processing power; the convergence time is very long, and the convergence is sometimes premature. To reduce the amount of computational time an a priori clustering method has been proposed (X. X. Wang et al., 2004). Unfortunately the proposed algorithm is intrinsically flawed limiting its usefulness (N. Y. Liang&G. B. Huang, 2008). Orr (M. J. L. Orr, 1995) has proposed a regularized forward selection (RFS) algorithm, based on subset selection (S. Chen et al., 1996), which combines forward subset selection and zero-order regularization. However, the subset selection method has several major disadvantages, the worst of which is that in order to increase the chance of obtaining a satisfactory RBF network; it has to use a very large set of candidate RBF nodes with different centers and radiuses. There are a number of other growing strategies (K. Li, J. Peng & G. W. Irwin, 2005; A. L. I. Oliveira et al., 2006 ; J. Gonzalez et al., 2003), but in all these methods the criterion for determining growth suffers from a lack of objectivity. Many of them are also time-consuming, sensitive to the input data, and do not consider the effect of the RBF output. Yingwei et al. (L. Yingwei et al., 1998) have proposed a pruning strategy, based on the relative contribution of each hidden node to the overall network output, which aims to reduce the complexity of the RBF neural network. In theory the size of the final neural network obtained by this method is minimal. Other methods for pruning RBF neural networks have been proposed by Salmer' on et al., (M. Salmer' on et al., 2001) Rojas et al., (I. Rojas et al.,2002) and Hao et al. (P. Hao & J. Chiang, 2006). A major problem with pruning methods is that they often require more computational time than growing methods. In fact, there are also quite a large number of parameters or variables that need to be preset; the training data needs to be stored and re-used for pruning purposes. A promising alternative is to combine growing and pruning methods together. Huang et al. (G. B. Huang et al., 2004) proposed a simple sequential learning algorithm called the "RBF growing and pruning algorithm"(GAP-RBF), which was later developed into the GGAP-RBF algorithm (G. B. Huang et al., 2004; Q. Meng & M. Lee, 2008). The GAP-RBF and GGAP-RBF methods use the "significance" of a hidden node to decide whether to add new nodes or reduce the number of redundant nodes. The "significance" of the node is also linked to the learning accuracy. However, since these algorithms are on-line procedures, they do not optimize the network over all past training data. Moreover, network initialization of the GAP-RBF algorithm requires a priori knowledge of the data which may not be available. Lian et al. (J. M. Lian et al., 2008) have proposed a self-organizing RBF neural network (SORBF) for real-time approximation of continuous-time dynamic systems. The aim of the SORBF network is to develop an algorithm that can be used in real time processes. However, the authors do not

investigate the failure cases for the algorithm. How to choose a suitable criterion to design RBF neural network architecture remains an open question.
Section 3 presents a new RBF neural network design method which is called the "repair RBF neural network algorithm" (RRBF). RRBF performs simultaneous network architecture design and parameter optimization within an integrated analytic framework. This approach has two technical advantages. The first advantage is that the method is not dependent on the input data: when the sensitivity analysis (SA) of the RBF output indicates that the prototype is not suitable, the RBF architecture is repaired. The second advantage is that the criterion used to determine whether the network should be repaired or not is more objective than the criterion used in other similar time-based methods (e.g. RFS and GAF-RBF): it is based on the sensitivity analysis (SA) value which calculates the contribution from hidden nodes over a given time period (t + 1, t + 2, . . . , t + m, where m is the period). The learning process starts by randomly initializing a single prototype in the feature space; then, the prototypes undergoes adaptive repair until the most appropriate number of prototypes is reached.

3.1 Previous and related works

a. RBF neural network

A standard RBF neural network consists of three layers: an input layer, a hidden layer, and an output layer. Fig. 1 shows a schematic diagram of the RBF network. As the nodes in the input layer represent the variables from input space and the nodes in the outer layer represent the desired response, the number of nodes in the input and output layers is configured in advance. A learning algorithm uses the defined optimization criteria to minimize the error between the actual response and the desired response.

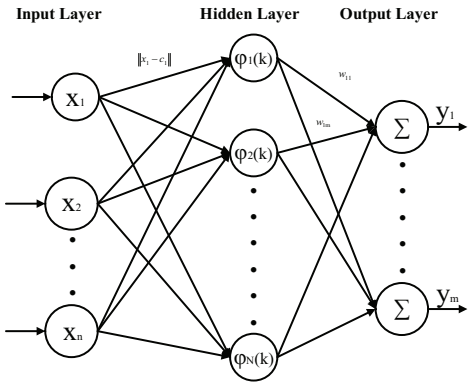


Fig. 1. Schematic diagram of RBF neural network (RBFNN)

As depicted in Fig. 1, the r -th output node of the RBF network can be expressed as follows:

$$y_i = \sum_{k=1}^{p} \phi_k(\|x - c_k\|) \cdot w_{ik}, \ i = 1,2,\cdots,m \tag{2}$$

where $x = [x_1, x_2, \cdots, x_n]^T$ is an input value; n is the number of input node; c_k is the k -th center node in the hidden layer, $k = 1,2,\cdots,p$, and p is the number of hidden nodes;

$\|x - c_k\|$ denotes the Euclidean distance between c_k and x; $\phi_k(\bullet)$ is the nonlinear transfer function of the *k*-th center; w_{ik} is the weighting value between the *k*-th center and the *i*-th output node; and m is the number of output nodes.

Equation (2) reveals that the output of the network is computed as a weighted sum of the hidden layer outputs. The nonlinear output of the hidden layer are described as $\phi_k(\bullet)$, which are radial symmetrical. Here the function chosen for this neural network is Gaussian function, and the description is shown as follows:

$$\phi(x) = e^{(-\frac{(x-v)^2}{\delta^2})} \tag{3}$$

where v and δ are the parameters of position and width of the centers. The activation functions commonly used for the classification and regression problems are in the Gaussian functions because they are continuous and differentiable; they provide softer output and improve interpolation capabilities. The significant parameters to design an RBFNN for solving problems are shown as:

1. The RBF position of the centers v;
2. The width δ of the centers;
3. The weights w;
4. The number of the hidden nodes p;

Based on 1), 2) and 4), the initializations of the centers are very important, if an incorrect initialization of the centers is performed, the approximation error could be increased. The reason is that during the execution of a local search algorithm to make a fine tuning of the centers and the radius, there is a possibility of falling into a bad local minimum. An on-line self-organizing algorithm is used for selecting the centers of the RBFNN in this chapter. This algorithm can add new centers to repair the RBFNN, which solves the pre-set problem of the conventional RBFNN. Meanwhile, the width of the centers is very important, if the width is not appropriate for the RBF, the training time will be heavy. When the RBFNN selects correct centers, the parameters of the widths and the weights will be adjusted at the same time.

b. The Sensitivity Analysis of Model Output (SAMO)

A thorough description of sensitivity analysis methods can be found in (Andrea Saltelli et al., 2006). The most common SA is sampling-based. There are several steps to conduct SA. The following steps can be identified as (the details can be found in (Andrea Saltelli et al., 2006)):

Step 1. Define the model, its input factors and output variable.
Step 2. Assign probability density functions or ranges of the variation to each input factor.
Step 3. Generate an input matrix through sampling design.
Step 4. Evaluate the output.
Step 5. Assess the influences or relative importance of each input factor on the output variable.

At Step 4), an empirical probability distribution for the output can be created which may lead to a first step of uncertainty analysis. After quantifying the variation of the output, SA consists in apportioning the variance of the output according to the input factors. The representation of the results can be described as the contribution to the input that describes the variance of the output into the percentages that each factor is accounting for. In this way,

the variance decomposition may allow the identification of the most influential factors. Then the SA should present analyses over the full range of plausible values of key parameters and their interactions, to assess how the change in response impacts changes in key parameters.
Sensitivity analysis (SA) is an available tool (J. Cariboni et al., 2007; A. Saltelli et al., 2000) which may be used to study the behavior of a system, or a model, and to ascertain the contribution ratio of the outputs depending on each or some of the input parameters. Among the SA methods, quite often they are identified almost as a mathematical definition, with a differentiation of the output respecting to the input. For this reason, quantitative measure of sensitivity, such as the EFAST method (A. Saltelli et al., 1999) is described as follows:

$$S_h = \frac{Var_h[E(Y|Z_h = \alpha_h)]}{Var(Y)} \tag{4}$$

where, Z_h denotes an input factor, $h = 1, 2, \cdots, p$, and Z_h represents the output connecting value of the hidden nodes in the RBFNN in this chapter. Y is the model response, $E(Y|Z_h = \alpha_h)$ is the expectation of Y conditional on a fixed value α_h of Z_h and the variance Var_h is taken over all the possible value of Z_h . The ratio S_h represents the main effect. It is called the first-order index in the SA terminology. Thus, the main effect of a factor represents the average effect of that factor on the response or conversely these methods allow the computation of that fraction of the variance of a given model output which is due to each input factor. In addition to the computation of the EFAST method which also provided an estimation of the total sensitivity index ST_h . The total effect includes the main effect as well as all the interaction terms involving that factor. The total effect is defined by:

$$ST_h = \frac{Amount\ of\ \mathrm{mod}el\ response\ \mathrm{var}iance\ Z_h}{Model\ response\ \mathrm{var}iance} \tag{5}$$

The model is additive when the response is nonlinear but interactions are negligible. In that case, the main effects are the suitable indexes for the sensitivity analysis of model output (Philippe Lauret et al., 2006).
EFAST is based on the Fourier decomposition of the variance in the frequency domain. This method is especially suited for a quantitative model independent global SA. The computational cost of this method is the number of model evaluations required and is a function of the number of input factors and the complexity of the model. The ever-increasing power of computers tends to make these global methods affordable for a large class of models. Among the SA methods, the total sensitivity index is undoubtedly the best guide to quantitatively rank the factors by order of importance. Indeed, even if this occurs rarely, interaction effects on a model response may dominate the main effects. So, whether the interaction effects are taken into account or not, the analysis may result in a different ranking of the factors' importance.

3.2 The repair method for selecting hidden nodes of RBFNN

Considering the intrinsic structure of RBFNN, if the RBFNN consists of assured inputs, certain hidden nodes and outputs, it can only adjust the weights v , δ and w which are in the hidden layer or connecting the hidden nodes and the output nodes. In this chapter, the

single response relationship between the hidden neurons and the output of the RBF is discussed. We state that the relevance of a hidden node is related to its influence on the RBF response. This is the key idea of the method proposed to determine the optimal architecture for the RBF. The parameters w of the hidden nodes are the input values of the repair algorithm based on the sensitivity analysis (SA). And this SA is based on the Fourier decomposition of the variance in the frequency domain.

a. Selecting Hidden Nodes

A generic model is assumed to describe an RBF neural network system. The model is represented by a mapping f (a deterministic or stochastic function) which relates the inputs domain to the output space:

$$Y = f(Z_1, Z_2, \cdots, Z_p) = \sum_{i=1}^{p} \beta_i Z_i \tag{6}$$

The input factors $(Z_1, Z_2, \cdots, Z_p)$ are supposed to be the variables described by the parameters w of the RBF hidden nodes, $Z_1 = w_{1\bullet}, Z_2 = w_{2\bullet}, \cdots, Z_p = w_{p\bullet}$, p is the number of the nodes in the output layer; $w_{1\bullet} = [w_{11}, w_{12}, \cdots, w_{1m}]^T$, m is the number of the nodes in the output layer. Y is taken to be a scalar even in the application we shall consider each output variable in turn. Based on the EFAST method, the polynomial expansion can be described again. The range of the factor Z_h is $\left[a_q, b_q\right]$, so the Z_h performances as follows:

$$Z_h^{(q)} = (b_q + a_q/2) + (b_q - a_q/2)\sin(\omega_h s^{(q)}) \tag{7}$$

where, $s^{(q)} = 2\pi n/N$, ω_h is the frequency, q is the simulation number, N is the total simulation number. If $Z_h^{(q)}$ is straightforward to note $z_h^{(q)} = \sin(\omega_h s^{(q)})$ and that the formula (6) should be as:

$$\begin{aligned} Y^{(n)} = Y_0 &+ \sum_{i=1}^{p} \beta_i \sin(\omega_i s^{(q)}) \\ &+ \sum_{i=1}^{p}\sum_{j=1}^{p} \beta_{ij} \sin(\omega_i s^{(q)})\sin(\omega_j s^{(q)}) \\ &+ \cdots \end{aligned} \tag{8}$$

Based on this formula, the linear effect of Z_h corresponds to the Fourier amplitude at the fundamental frequency ω_h. In EFAST, each input factor Z_h is related to a frequency ω_h and a set of suitable defined parametric equations:

$$Z_h(s) = G_h(\sin(\omega_h s)) \qquad h = 1, 2, \cdots, p \tag{9}$$

The equations allow each factor to vary in a given range, as the new parameter s is varied. They define a curve which explores the input factors' space systematically. As s varies, all the factors oscillate at the corresponding driving frequency ω_h and their range is systematically explored. For the EFAST method, a parametric representation of the form is often used.

$$Z_h(s) = \frac{1}{2} + \frac{1}{\pi}\arcsin(\sin(\omega_h s)) \tag{10}$$

This transformation allows a better coverage of the factors' apace since it generates samples that are uniformly distributed in the range [0, 1].
If the range of variation of the factor Z_h is $[a, b]$, the parametric representation of the form should be:

$$Z_h(s) = \frac{b+a}{2} + \frac{b-a}{\pi}\arcsin(\sin(\omega_h s)) \tag{11}$$

As each factor Z_h oscillates periodically between $[a, b]$ at the corresponding frequency ω_h, the model output Y exhibits different periodicities that result from the combination of the different frequencies $\omega_{i=1,\cdots,p}$, whatever the model f is. Just for Fourier amplitudes, the p-factor model can be described as follows:

$$f(s) = f(Z_1(s), Z_2(s), \cdots, Z_p(s)) \tag{12}$$

where, the range of s is $[-\pi, \pi]$, so the expanded in a Fourier series of the form:

$$f(s) = \sum_{j=-\infty}^{\infty} (A_j \cos(\omega_j s) + B_j \sin(\omega_j s)) \tag{13}$$

where, the Fourier coefficients are defined as:

$$A_j = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(s)\cos(\omega_j s)ds \,,\; B_j = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(s)\sin(\omega_j s)ds \,;$$ the range of s is $[-\pi, \pi]$.

Therefore, N equally spaced sample points are required to perform the Fourier analysis. N represents the sample size and coincides with the number of model evaluations. Based on the Fourier translation, the variance D_y can be computed as:

$$D_y = Var(Y) = 2\sum_{k=1}^{+\infty}(A_k^2 + B_k^2) \tag{14}$$

The portion of the variance of Y explained by Z_h alone is:

$$D_y = Var_{Z_h}[E(Y|Z_h)] = 2\sum_{k=1}^{+\infty}(A_{k\omega_h}^2 + B_{k\omega_h}^2) \tag{15}$$

where, $A_{k\omega_h}$ and $B_{k\omega_h}$ denote the Fourier coefficients for the fundamental frequency and its higher harmonics $k\omega_h$. Then the expansion of the main effect is given by:

$$S_h = \frac{Var_{Z_h}[E(Y|Z_h)]}{Var(Y)} = \frac{2\sum_{k=1}^{+\infty} A_{k\omega_h}^2 + B_{k\omega_h}^2}{Var(Y)} \tag{16}$$

We stated above that in order to evaluate the main effect of Z_h, one must calculate the Fourier coefficients at the fundamental frequency ω_h and all the harmonics.
Because there is M interference factor (usually set to 4 or 6 in the SA community). So that the first-order sensitivity index is approximated by:

$$S_h = \frac{Var_{Z_h}[E(Y|Z_h)]}{Var(Y)} = \frac{2\sum_{k=1}^{M} A_{k\omega_h}^2 + B_{k\omega_h}^2}{Var(Y)} \tag{17}$$

The number of simulation runs represents the sampling frequency and, to meet the Nyquist criterion, it must equal to:

$$N = 2M\omega_{\max} + 1\,,\ (\omega_{\max} = \max(\omega_i)) \tag{18}$$

The variance $Var(Y)$ can be evaluated in the frequency domain through the following relationship:

$$Var(Y) = 2\sum_{w=1}^{\frac{(N-1)}{2}} (A_\omega^2 + B_\omega^2) \tag{19}$$

A_ω and B_ω denote the Fourier coefficients at frequency ω. In fact, based on this analysis the total sensitivity index ST_h can be shown as:

$$ST_h = \frac{\sum_{\omega=M\max(\omega_h)+1}^{\frac{(N-1)}{2}} (A_\omega^2 + B_\omega^2)}{\sum_{\omega=1}^{\frac{(N-1)}{2}} (A_\omega^2 + B_\omega^2)} \tag{20}$$

The advantages of the ST_h make the judgments in this chapter much better than the S_h. Based on the former analysis of ST_h, the main steps of the proposed approach for selecting hidden nodes in the RBF are as table 1. For the proposed total sensitivity index, some points have to be highlighted. First, usually, the sensitivity index for the ratio of the inputs is dependent on the current time $(t+m)$. But the total sensitivity index can calculate the contribution of period of time $(t, t+1, \cdots, t+m)$ which is more objective then the others based on the current time $(t+m)$. Second, the computation of total sensitivity index occurs when the FNN has been trained into a minimum of the error function or when over fitting begins. But this proposed approach is not necessary. In other words, computation of total sensitivity index starts when the FNN has been trained for some epochs in this chapter.

b. Adding New Nodes

As the active nodes in the hidden layer are found by the SA, the new nodes will be inserted to repair the prototype of the RBFNN. The parameters v, δ and w of the new nodes are given as follows (assuming the h-th node is the active node and only a new node will be inserted):

1. The position of the centers v ; in order to speed up the convergence, the center of the new inserted node is given by the Nearest Neighbor Interpolation theory, and the position of the center is: $v = \frac{1}{2} v_h + v_{nearest}$. $v_{nearest}$ is the nearest node of the active node.
2. The width δ of the centers; finding out the minimal value of δ within the existing nodes, and the width of the new node is: $\delta = \min(\delta_i)$.
3. The weights w ; in order to speed up the convergence and keep the least error of the next output of the RBF. The weights w of the new node is: $w = 0.5 \times w_{act}$, w_{act} is the weights of the active node.

By adding a new node to the hidden layer, the number of the nodes in hidden layer will be updated, the centers, width and the weights should be updated for the whole RBFNN. Since the new node is inserted based on the activity of the node for the output, the new node can ensure the activity for the whole network.

①For each factor Z_h , finding its minimal and maximal values a_h and b_h .
②Setting the interference factor to M and choose the number of simulation runs N .
③Computing the frequency $\omega_h = (N-1)/2M$ to be assigned to the factor Z_h .
④Just considering the output of the hidden nodes in the RBFNN, the factors are varied according to the curve definition; computing the total sensitivity index ST_h of the factor Z_h based on formula (20).
⑤Computing the percentage contribution $ST_h \Big/ \sum_{i=1}^{p} ST_i$ of the output of each hidden node.
⑥If the percentage contribution $ST_h \Big/ \sum_{i=1}^{p} ST_i$ larger than ε_1 , the node h is the active node and a new node should be inserted with relation to the active node.
⑦Repeating ①-⑥ until all of the existing hidden nodes are considered.

Table 1. Procedure of selecting hidden nodes.

c. Parameters adjusting

After adding new nodes to the RBFNN, the number of the centers is confirmed timely. Then the whole parameters of the RBFNN will be adjusted. In fact, these parameters adjusting relate to the final capabilities of the RBFNN directly.

Considering the training process of the RBF, researchers have put forward many methods to adjust the parameters of the RBFNN. The parameter adjusting algorithm is based on the mean squared error (MSE) in this chapter:

$$E(t) = \frac{1}{T} \sum_{t=1}^{T} (y_d(t) - y(t))^2 \tag{21}$$

Where, T is the total number of the samples, $y_d(t)$ is the expected output of the t step, $y(t)$ is the practical output of the t step. The goal of this method is to reach $E(t) < \varepsilon$ by learning. ε is the expected stable error. The details of the adjusting process are:

4. The weights w ;

$$w_{ij}(t+1) = w_{ij}(t) - \eta_1 \frac{\partial E(t)}{\partial w_{ij}(t)} \tag{22}$$

$i = 1,2,\cdots,p$; $j = 1,2,\cdots,m$.

η_1 is a plus constant, and it is less than 1.

5. The width δ of the centers;

$$\delta_i(t+1) = \delta_i(t) - \eta_2 \frac{\partial E(t)}{\partial \delta_i(t)} \quad i = 1,2,\cdots,N ; \tag{23}$$

η_2 is a plus constant, and it is less than 1.

6. The position v of the centers;

In section B, the position v of the new inserting nodes has been discussed, and the position v of the other centers will be discussed here.

$$v_i(t+1) = \begin{cases} v_i(t) - \eta_3(P(k) - v_i(t)) & if\ v_i\ is\ active \\ v_i(t) - \eta_4 \dfrac{\partial E(t)}{\partial v_i(t)} & others \end{cases} \qquad i = 1,2,\cdots,p \tag{24}$$

η_3, η_4 are the plus constants, which are less than 1; p is the whole number of the hidden nodes after adding new nodes. This new algorithm will continue to circulate until reach the stable error.

d. Repair RBFNN

Choose the parameters M , N , ε_1 , η_1 , η_2 , η_3 , η_4 ; initialize the centers v , The width δ , The weights w and the number of the hidden nodes p . In each sampling period, the main steps of the repair RBFNN algorithm are shown as table 2:

Training a given RBF for some epochs.
Finding out the active nodes in the hidden layer and go to step ; if there is no active node, go to step.
Adding new nodes to the RBFNN.
Adjusting the parameters of the RBFNN and updating the whole value of the parameters.
Repeating the step - , Stopping computing until the RRBF achieves the expected stable MSE.

Table 2. Procedure of RRBF neural network

In this proposed repair RBF algorithm, two points need to be highlighted. First, usually, the former dynamic RBFNN used the clustering methods or based on the information matrix,

these methods required heavy computation. The SA method used in this chapter is based on the Fourier translation, and then the percentage contribution of the hidden neurons is computed in a quantitative way. This repair does not have to judge for the structure every step, therefore gives a completely satisfactory method for growing the hidden nodes. Second, the common SA method is based on the quantitative and qualitative methods and between local and global techniques. The SA method used in this chapter is global, and the percentage contribution of the hidden neurons is direct related to the RBF output.

3.3 Simulations

To demonstrate the effectiveness of the proposed algorithm, three examples are discussed in this chapter: nonlinear function approximation, dynamic system identification. The results are compared with other algorithms such as SGP-RBF (M. J. Er & S. Wu, 2002), and GAP-RBF (P. Hao & J. Chiang, 2006).

a. Tracking Nonlinear Function

Consider a common nonlinear function which was also used in (Gang Lengl et al., 2004) to demonstrate the effect of the algorithms:

$$y = 1.1 \times (1 - x_1 + 2x_2{}^2) \times e^{(-x_3{}^2/2)} \tag{25}$$

where there are 3 continuous attributes x_i ($i = 1,2,3$). And the data set $\{x_i; y\}$ is generated by the equation (25), and x_i satisfies the uniform distribution U [0, 10]. For each trial, the size of training samples is 200, and the size of testing samples is 200.

The real output at time k is $y(k)$, the required value at time k is $y_d(k)$, the error at time k will be $e(k) = y_d(k) - y(k)$. The inputs of this SORBF are given as: $P(k) = (x_1(k), x_2(k), x_3(k))$. The training MSE for tracking is 0.001, the initial radius of every hidden node is 0.1; the initial weight of every hidden node is randomly given in the interval [0, 1]. The initial value of M and N are $M = 4$ and $N = 5000$. There are two initial nodes in the hidden layer.

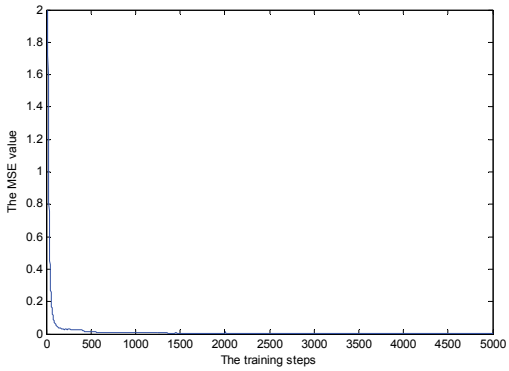


Fig. 2. The error results in the tracking process

Fig 2 shows 5000 steps of the error values in the training process, the error values show that when the new node is inserted to the hidden layer, the error values will shake. However, the error values can be convergence quick after adding new nodes; Fig 3 shows the dynamic

number of the nodes in the tracking process; Fig 4 shows the width δ of the centers after training; Fig 5 shows the RBF position of the centers v after training.

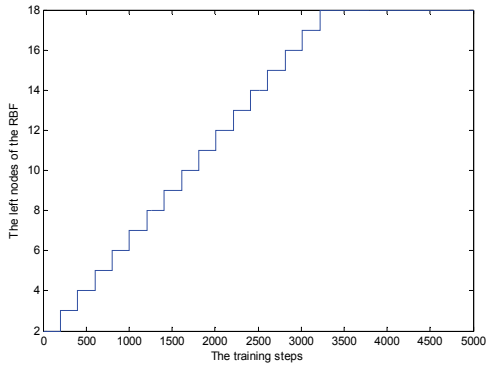


Fig. 3. The number of the nodes in the tracking process

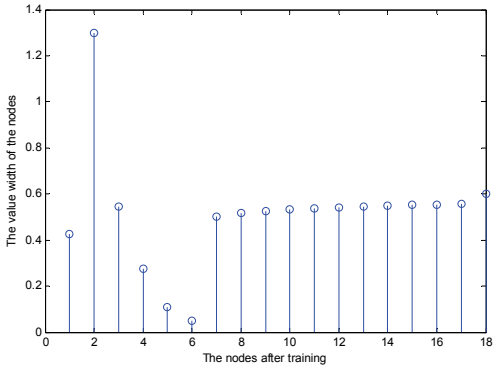


Fig. 4. The δ values of the left nodes after training

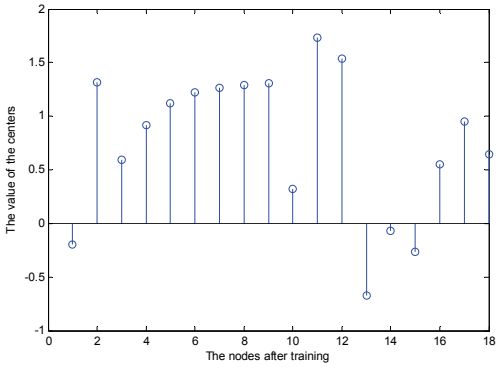


Fig. 5. The v values of the left nodes after training

The final results of all the algorithms are shown in Table 3. The compared values are: training time, test error, left nodes after training.

Algorithm	CPU Time(s)	Training Error	Testing Error	No. of Neurons
SGP-RBF	42.32	0.001	0.0086	19
GAP-RBF	26.86	0.001	0.0031	16
RRBF	22.67	0.001	0.0025	18

Table 3. The performances comparison of the different algorithms

Based on table 3, the RRBF is faster and more accurate than SGP-RBF and GAP-RBF. The structure of this RRBF is simpler than the SGP-RBF; the memory space is fewer owing to the simple structure. The nodes in the RRBF are more than the GAP-RBF, but the algorithm is faster and more accurate than GAP-RBF. The results prove that this RRBF performs better than the former two algorithms and it has better ability to model the nonliear systems.

4. Experimental results of COD soft-sensing

The widespread popularity of neural networks in many fields is mainly due to their ability to approximate complex nonlinear mappings directly from the input samples. They can solve many problems especially in modelling which are difficult to handle using classical parametric techniques. Since the activated sludge wastewater treatment process is a high nonlinear and complicated system, it is very suitable to be modeled by neural network. In the experiment, we select the most important influent water quality parameters COD, Mixed Liquor Suspended Solid (MLSS), pH, Oil and NH3-N as the research variables, where COD measures the total amount of oxygen that can be combined with the chemical compounds (organic and inorganic) in the water. For it is an extremely important value to reveal the total amount of pollution in water and is easily to acquire, it is widely used in the wastewater treatment process. MLSS is a measure of dry solids concentration in mg/l in mixed liquor in an aeration tank. PH shows the degree of acid or alkali in the influent water, Oil is the content of oil contamination, and NH3-N delegates the content of nutritious contamination in the influent water. The output vector of the network is the value of COD of the water flow out of the system after the treatment. By the way, the data used in the experiment are from a small wastewater treatment factory.

The model used the data of SS, pH, oil and NH3-N as inputs to estimate the settled sewage COD. Thus, the data set used to develop the model consisted of 100 samples of the model inputs and settled sewage COD and 100 samples for testing. Hence, the model has been developed to determine if acceptable estimates can be produced from a limited amount of relatively inexpensive and readily available information. Note also that for each set of input data the model is estimating the corresponding COD. It is used on-line then each model would give an instantaneous estimation of the COD that could be expected if standard laboratory tests were performed. Each model input was scaled to lie in the range 0 to 1 to reduce the effects of widely differing magnitudes of input data that could lead to a biased model. The model output was scaled in an identical fashion. The error measures for this model are 3mg/L confidence limits. The results are shown Figs. 6–11.

Fig.6 gives the training results of COD; Fig.7 describes the error value of the trained results which are less than 3mg/L; Fig.8 shows the left nodes in the hidden layer after training; Fig.9 describes the error value in the training process; Fig.10 shows the predictions results of COD; Fig.11 shows the error value of the predicting process.

Based on the results, this RRBF is able to be used for the COD measurement on-line. The results demonstrate that the COD trends in the settled sewage at the wastewater treatment could be predicted with acceptable accuracy using SS, pH, Oil and NH_3-N data as model inputs. This approach is relatively straightforward to implement on-line, and could offer real-time predictions of COD. It is concluded that this is a significant feature of this approach since COD is the more commonly used and readily understood measure.

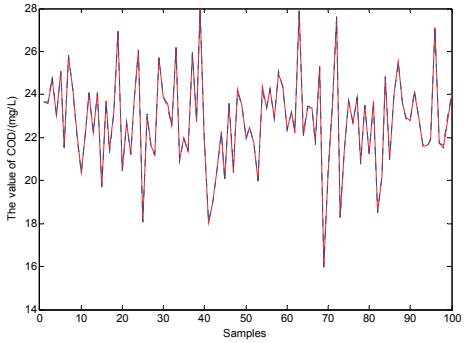


Fig. 6. The training results of COD

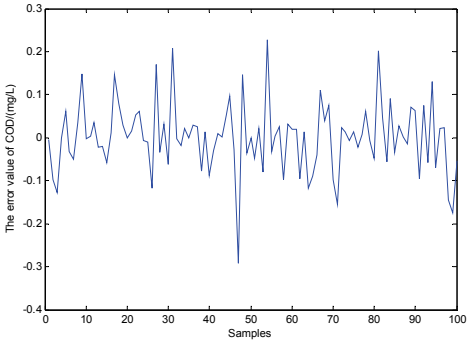


Fig. 7. The error value of the trained results

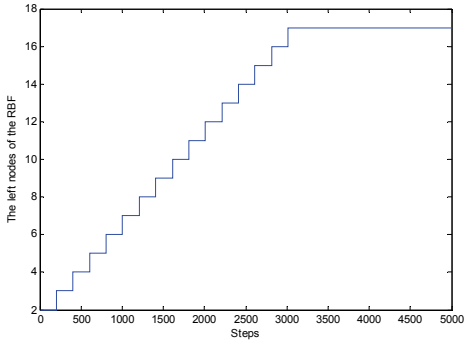


Fig. 8. The number of the nodes in the training process

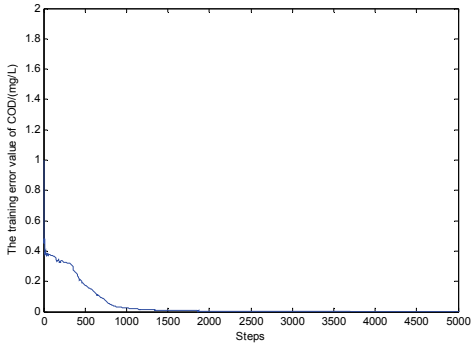


Fig. 9. The error value of the training process

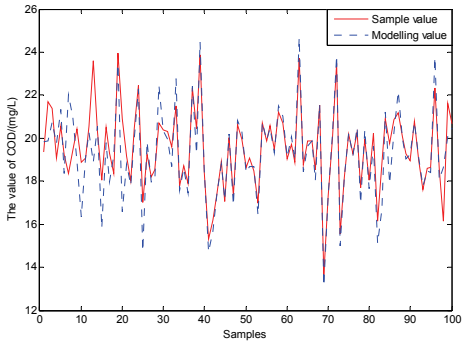


Fig. 10. The predictions results of COD

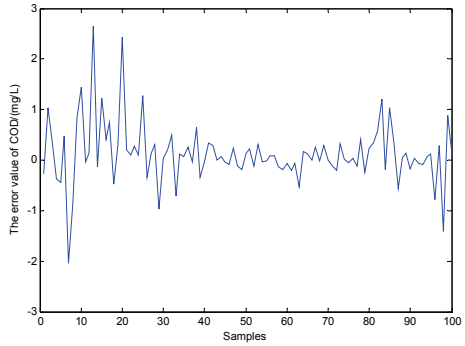


Fig. 11. The error value of the predictions results

5. Conclusion and future work

Section 3 presents a repair algorithm for the design of a RBF neural network which is called RRBF to model the COD in wastewater treatment process.
The following important points should be noted:

1. In most algorithms the criterion used to determine growth is dependent on the current time (t + m). This section, however, uses the sensitivity index, which can calculate the contributions of hidden nodes over a number of time periods (t + 1, t + 2, . . . , t + m). This is more objective than using a criterion based on the current time (t + m).
2. The criterion used to select hidden nodes is based on the SA method of the RBF output – it is independent of the input data.
3. Less computation is required because the initial weights of the new inserted nodes are utilized to calculate the repaired RBF. Simulation results show that the proposed algorithm performs well in modelling the key parameter, COD, in the wastewater treatment process. This type of RRBF based approach may potentially be used in any area where it is difficult to measure a range of variables because of the need for specialized equipment. It can, therefore, be a cost effective solution in many application areas where such measurements are needed.

The following future work is under investigation.

1. An adaptive repairing strategy which will allow the addition of hidden nodes during the training process based on the SA of the network output.
2. A pruning operation which will reduce the hidden nodes that have little contribution to the output of the RBF network is under investigation.
3. The application of the algorithm to other areas is also on-going.
4. The growing Mechanism need further improvement.

6. Reference

B. Lin, B. Recke, J. K. H. Knudsen, & S. B. Jrgensen. (2007). A systematic approach for soft sensor development, Computers and Chemical Engineering, vol. 31, 2007, pp. 419-425

W. Ran, J. Qiao & X. Ye. (2004). Soft-measuring approach to on-line predict BOD based on PCA time-delay neural network, in WCICA 2004 - Fifth World Congress on Intelligent Control and Automation, Conference Proceedings, pp. 3483-3487, June 15, 2004 - June 19, 2004, Hangzhou, China

M. H. Kim, Y. S. Kim, A. A. Prabu, and C. K. Yoo. (2009). A systematic approach to data-driven modelling and soft sensing in a full-scale plant, Water Science and Technology, vol. 60, 2009, pp. 363-370

W. Wang & M. Ren.(2002) Soft-sensing method for wastewater treatment based on BP neural network, in Proceedings of the 4th World Congress on Intelligent Control and Automation, pp. 2330-2332, June 10, 2002 - June 14, 2002, Shanghai, China

D. Guclu & S. Dursun. (2008). Amelioration of carbon removal prediction for an activated sludge process using an artificial neural network (ANN), Clean-Soil Air Water, vol. 36, 2008, pp. 781-787

J. B. Zhu, J. Zurcher, M. Rao, et al. (1998). An on-line wastewater quality predication system based on a time-delay neural network. Engineering Applications of Artificial Intelligence, 1998, 11(6): 747~758

Q. M. Cong, T.Y. Chai, Y. Wen. (2009). Modelling wastewater treatment plant via hierarchical neural networks. Control Theory & Applications，vol.26 , 2009, pp 8-14

Q. L. Chen, W. Chai, J. F. Qiao. (2010). Modeling of wastewater treatment process using recurrent nueral network, Proceedings of the 8th World Congress on Intelligent Control and Automation, pp 5872-5876, July 6-9 2010, Ji'nan, China

J. F. Qiao , H. G. Hang. (2010). A repair algorithm for radial basis function neural network and its application to chemical oxygen demand modelling, International Journal of Neural Systems, Vol. 20, No. 1, 2010, 63–74

Mathur A, Martin JF. (2004). Stem cells and repair of the heart. Lancet, vol.364, 2004, pp.183–192

S. Ghosh-Dastidar and H. Adeli. (2009). Spiking neural networks, International Journal of Neural Systems, 19(4), 2009. pp 295 – 308

A. Karim and H. Adeli. (2003). Radial basis function neural network for work zone capacity and queue estimation, Journal of Transportation Engineering 129(5),2003, pp 494 – 503

S. Ghosh-Dastidar, H. Adeli and N. Dadmehr. (2008). Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection, IEEE Transactions on Biomedical Engineering, 55(2), 2008, pp 512 – 518

H. Liu, X. Wang and W. Qiang. (2007). A fast method for implicit surface reconstruction based on radial basis functions network from 3D scattered points, International Journal of Neural Systems 17(6) ,2007, pp 459 – 465

T. D. Jorgensen, B. P. Haynes and C. C. F. Norlund. (2008). Pruning artificial neural networks using neural complexity measures, International Journal of Neural Systems 18(5), 2008, pp 389 – 403

D. S. Huang and J. X. Du. (2008).A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks, IEEE Transaction on Neural Networks, 19(12), 2008, pp 2099 – 2115

N. Jin and D. R. Liu, (2008). Wavelet basis function neural networks for sequential learning, IEEE Transaction on Neural Networks, 19(3), 2008, pp 535 – 540

A. Esposito, M. Marinaro, D. Oricchio and S. Scarpetta. (2000). Approximation of continuous and discontinuous mappings by a growing neural RBFbased algorithm, Neural Networks, 13(3), 2000, pp 651 – 665

X. X. Wang, S. Chen and D. J. Brown. (2004). An approach for constructing parsimonious generalized Gaussian kernel regression models, Neurocomputing, 62, 2004, pp 441 – 457

N. Y. Liang&G. B. Huang. (2008). P. Saratchandran and N.Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Transaction on Neural Networks 17(6) 1411 – 1423

M. J. L. Orr. (1995). Regularization on the selection of radial basis function centers, Neural Computation, 7(3), 1995, pp 606 – 623

S. Chen, E. S. Chng and K. Alkadhimi. (1996). Regularized orthogonal least squares algorithm for constructing radial basis function networks, International Journal of Control, 64(5), 1996, pp 829 – 837

K. Li, J. Peng and G. W. Irwin. (2005). A fast nonlinear model identification method, IEEE Transaction on Automation Control, 50(8), 2005, pp 1211 – 1216

A. L. I. Oliveira, E. A. Medeiros, T. A. B. V. Rocha, M. E. R. Bezerra and R. C. Veras. (2006). On the influenceof parameter θ on performance of RBF neural networkstrained with the dynamic decay adjustment algorithm, International Journal of Neural Systems, 6(4), 2006, pp 271 – 281

J. Gonzalez, I. Rojas, J. Ortega. (2003). H. Pomares, F.J. Fernandez and A. F. Diaz, Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation, IEEE Transaction on Neural Networks, 14(6), 2003, pp 1478 – 1495

L. Yingwei, N. Sundararajan and P. Saratchandran. (1998). Performance evaluation of a sequental minimal radial basis function (RBF) neural network learning algorithm, IEEE Transaction on Neural Networks, 9(2), 1998, pp 308 – 318

M. Salmer' on, J. Ortega, C. G. Puntonet and A.Prieto, Improved RAN sequential prediction using orthogonal techniques, Neurocomputing 41 (2001) 153 – 172

I. Rojas, H. Pomares, J. L. Bernier, J. Ortega, B. Pino, F. J. Pelayo and A. Prieto. (2002). Time series analysis using normalized PG-RBF network with regression weights, Neurocomputing, 42, 2002, pp 267 – 285

P. Hao and J. Chiang. (2006).Pruning and model-selecting algorithms in the RBF frameworks constructed by support vector learning, International Journal of Neural Systems, 16(4), 2006, pp 283 – 293

G. B. Huang, P. Saratchandran and N. Sundararajan. (2004). An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks, IEEE Transaction on Systems, Man, and Cybernetics—Part B: Cybernetics, 34(6), 2004, pp 2284 – 2292

G. B. Huang, P. Saratchandran and N. Sundararajan. (2005). A generalized growing and pruning RBF (GGAP-RBF) neural network for nunction approximation, IEEE Transaction on Neural Networks, 16(1), 2005, pp 57 – 67

Q. Meng and M. Lee. (2008). Error-driven active learning in growing radial basis function networks for early robot learning, Neurocomputing, 71(7 – 9), 2008, pp 1449 – 1461

J. M. Lian, Y. G. Lee, S. D. Sudhoff and S. H. Zak. (2008). Self-organizing radial basis function network for real-time approximation of continuous-time dynamical systems, IEEE Transaction on Neural Networks, 19(3), 2008, pp 460 – 474

Andrea Saltelli, Marco Ratto, Stefano Tarantola, Francesca Campolongo. (2006). Sensitivity analysis practices: Strategies for model-based inference. Reliability Engi-neering and System Safety, vol.91, no.10-11, 2006, pp 1109–1125

J. Cariboni, D. Gatelli, R. Liska, A. Saltelli. (2007). The role of sensitivity analysis in ecological modelling. Ecological Modelling, vol.203, no.1-2, 2007, pp 167–182

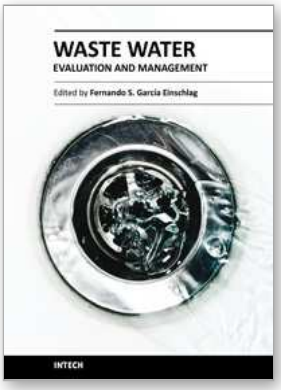
A. Saltelli, K.-S. Chan, and E. M. Scott. (2000). Sensitivity Analysis. New York: Wiley

A. Saltelli, S. Tarantola, and K.-S. Chan. (1999). A quantitative model independant method for global sensitivity analysis of model output. Technometrics, vol. 41, no. 1, 1999, pp. 39–56

Philippe Lauret, Eric Fock, and Thierry Alex Mara. (2006). A Node Pruning Algorithm Based on a Fourier Amplitude Sensitivity Test Method. IEEE Trans. Neural Networks, vol. 17, no. 2, 2006, pp 273-283

M. J. Er & S. Wu. (2002). A fast learning algorithm for parsimonious fuzzy neural systems. Fuzzy Sets and Systems, vol.126, no.33, 2002, pp 337–351

Gang Lengl, Girijesh Prasad, Thomas Martin McGinnity. (2004). An on-line algorithm for creating self-organizing fuzzy neural networks. Neural Networks, vol.17, no.10, 2004, pp 1477–1493



Waste Water - Evaluation and Management

Edited by Prof. Fernando SebastiÃ¡n GarcÃa Einschlag

ISBN 978-953-307-233-3
Hard cover, 470 pages
Publisher InTech
Published online 01, April, 2011
Published in print edition April, 2011

Fresh water resources are under serious stress throughout the globe. Water supply and water quality degradation are global concerns. Many natural water bodies receive a varied range of waste water from point and/or non point sources. Hence, there is an increasing need for better tools to asses the effects of pollution sources and prevent the contamination of aquatic ecosystems. The book covers a wide spectrum of issues related to waste water monitoring, the evaluation of waste water effect on different natural environments and the management of water resources.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821