A hybrid algorithm based on neural network for DO concentration control

Ran Zhen[1,a],Liang Wang[2,b*], Xueli Wu[3], Xiaojing Wu[3] , Chao Si[3], Han Bai[3]

[1] Department of Electrical Engineering, Hebei University of Science and Technology, 26 Yuxiang Street, Shijiazhuang, 050018, P R China

[2]Department of Electrical Engineering, Hebei University of Science and Technology, 26 Yuxiang Street, Shijiazhuang, 050018, P R China

[3]Department of Electrical Engineering, Hebei University of Science and Technology, 26 Yuxiang Street, Shijiazhuang, 050018, P R China

[a]E-mail: zhenranzr@126.com,[b]E-mail: wangliang19890126@163.com

Key Words: nonlinear systems, neural networks, self-organizing map, Hybrid learning, dissolved oxygen concentration control.

Abstract. The control of the dissolved oxygen concentration in an aerobic reactor is one of the most important and challenging tasks, because of its strong nonlinearities and large uncertain dynamics. In this paper a hybrid algorithm is used to approach this nonlinear dynamic system using feedforward neural network to solve the DO concentration control problem. This hybrid algorithm uses different learning algorithm separately. The weights connecting the input and hidden layers are firstly adjusted by a self-organized learning procedure, while the weights between hidden and output layers are trained by supervised learning algorithm, such as a gradient descent method. The simulation examples are provided to demonstrate the efficiency of the approach compared with radial basis function neural network.

Introduction

Due to the complexity of the physical, chemical and biological phenomena, the performance of the process is largely dependent on environmental and operational conditions. Wastewater treatment processes are difficult to control, because of the large disturbances in flow and load ， the different physical and biological phenomena which can take place. Several extensive surveys of the activated sludge process control using simulation can be found in the literature [1,2]. Such kind of bioprocess is well-known that 1) it is a complex system for the large nonlinearities, strongly coupled variables and large uncertain dynamics, and 2) most variables in the biological reaction are unmeasurable, which are essential for the control design [3,4]. So we can see that the dissolved oxygen model belongs to a class of complex nonlinear dynamic systems. It is now well know that the nonlinear dynamical systems can exhibit extremely complex dynamic behavior, for this reason it becomes necessary to use intelligent control techniques. More recently, neural networks, fuzzy logic and genetic algorithm have been applied successful to problems of control and identification of nonlinear systems for several domains of application [5,6]. For example[7], propose adaptive Neural Network decentralized backstepping output-feedback control for nonlinear large scale systems with time delays. In [8],robust adaptive Neural Networks control was developed for a class of uncertain MIMO nonlinear systems with input nonlinearities. A neural network, one of the most popular intelligent computation approaches, has the capability to approximate any desired nonlinear function to an arbitrary degree of accuracy [9]. One of the most commonly used method to train FNN is based on Radial Basis Function (RBF) algorithm. Compared to the BP neural network and other types of neural network, RBF neural network not only can approximate any nonlinear model, but also there is no problem of a local minimum, so it learns faster and more accurate. Another advantage of the RBF neural network is that there is no need to give a number of nodes in the hidden layer. Although the learning method has achieved many successes in the neural control of nonlinear systems. In [10], a novel radial basis function (RBF) neural network is proposed and applied successively for online stable identification and control of nonlinear discrete-time systems. In [11], the regularization RBF neural networks for nonlinear system modeling. In [12], a PID control combined with Radial Basis Function (RBF) neural network was proposed for course control of ship steering. In [13], all uncertain dynamics of the wastewater treatment are approximated by using radial basis function (RBF) neural networks (NNs). But RBF neural network also has its drawbacks, such as the structure of the network, the center vector of the basis function, initial weights are not well identified, in many cases we select them randomly, the centers vector of the network which is constructed by randomly selected is difficult to achieve a satisfactory level, another drawbacks is that as RBF is belongs to the scope of BP network, so it can not avoid the problems of slow convergence. Training speed is significantly increased by using various second orders algorithm [14,15].Many other algorithms with the emphasis on hybrid techniques have been developed to speed up the training iteration of feedforward neural network[16,17] .

In this paper, we used a new alternative hybrid algorithm[18,19].This algorithm could be split up into two stages. In the first stage, a structure learning which includes adding hidden neurons is used to determine the network size. This learning algorithm starts with a single training pattern and a single hidden layer neuron. When a new pattern is presented to the net, the distance between the input and the weight vector is computed. If the distance is smaller than a prespecified distance threshold, the new input pattern belongs to this neuron, otherwise, this new pattern become a new neuron of hidden layer. This procedure is repeated until all patterns are presented. The network can automatically create the neurons of the hidden layer and their initial weights. This approach for constructing the self organizing map is proposed as a dynamic version of the Kohonen self organizing map in order to overcome the weakness of the need for user defined static map structure of SOM. In the second stage, network parameters are adjusted using a recently proposed approach [20]. This learning procedure uses different learning algorithms separately. The weights between input and hidden layers are firstly adjusted by a self organized learning procedure [21], while the weights connected hidden and output layers are trained by a supervised learning algorithm, such as a gradient descent method[22]. The algorithm will improve the convergence speed for iteration.

The paper is organized as follows: In Section 2, we present the Activated Sludge system model. In Section 3,description of the learning algorithm. In Section 4, simulation results and comparisons with the RBF are given. Finally, in Section 5, we present the main conclusions.

Activated sludge system model

On the basis of the international water association established activated sludge model no. 1 and the law of material balance: (input) - (output) = (cumulants)-(reaction), we can receive a link to sewage aeration activated sludge process mathematical mode.

A mathematical model is obtained based on the following assumptions :

(1) The microorganisms’ growth rate is larger than their death rate and obeys the Monod law.

(2) No biochemical reactions take place inside the settling tank; biomass in the sedimentation tank is negligible.

(3) The inflow stream contains no biomass; complete settling is achieved, hence the sludge wastage is restricted to a waste stream.

The model is then given by

$$\begin{cases} \dfrac{\mathrm{d}X}{\mathrm{d}t} = \dfrac{Q}{V} X_{\,i} - \dfrac{Q_w}{V} CX + \hat{m} \dfrac{S}{k_s + S} \dfrac{o}{k_o + o} X - k_d X \\ \dfrac{\mathrm{d}S}{\mathrm{d}t} = \dfrac{\hat{m}}{Y_{NH}} \dfrac{S}{k_s + S} \dfrac{o}{k_o + o} X + \dfrac{Q}{V}(S_i - S) \\ \dfrac{\mathrm{d}O}{\mathrm{d}t} = -\dfrac{1 - f \cdot f_x Y_{NH}}{f Y_{NH}} \hat{m} \dfrac{S}{k_s + S} \dfrac{o}{k_o + o} X - f_x k_d X + u \end{cases} \tag{1}$$

where X is the biomass concentration in the aeration tank, X_i is the inflow biomass concentration, S is the reactor substrate con-centration, S_i is the substrate concentration contained in inflow, O is the oxygen concentration in reactor tank, $\hat{m}$ is the maximal specific growth rate, k_s is the half-velocity constant, substrate concentration at one-half the maximal growth rate, k_o is the oxygen half-saturation coefficient for heterotrophic biomass, k_d is the endogenous decay coefficient, Q is the inflow, V is the reactor volume, Q_w is the wastage flow, C is concentration factor in the sedimentation tank, Y is the yield coefficient, Y_{NH} is the observed yield coefficient, f is a factor, which correlates the substrate with oxygen demand, f_x is the consumption factor, and u is the oxygen transfer rate.

In practice, the parameters $k_s \ll S$ and $k_o \ll O$, so the system dynamics in (1) can be rewritten as

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \tag{2}$$

where $A = \begin{bmatrix} \hat{m} - k_d - \frac{Q_w}{V}C & 0 & d \\ -\frac{1}{Y_{NH}}\hat{m} & -\frac{Q}{V} & 0 \\ -\frac{1 - ff_xY_{NH} - ff_xY_{NH}k_d}{fY_{NH}}\hat{m} & 0 & d \end{bmatrix}$, $B = [0 \quad 0 \quad 1]^{\mathrm{T}}$, $C = [0 \quad 0 \quad 1]$, $D = [0 \quad 0 \quad 0]$,

where d is a impulse coefficient which is settings for dissolved oxygen. It is proposed to strengthen the biological effect of the dissolved oxygen in the reaction process.

This process is a multi-input and single-output model, it belongs to nonlinear dynamic system, So we use the follow method to control the DO concentration.

Description of the learning algorithm

Without loss of generality, a single hidden layer FNN is used in this paper. It is a three-layer FNN controller based on a hybrid algorithm. Bypass weights from the input layer to the output layer are used, As the process described above is a MISO system, the FNN is also a MISO. There are n units in the input layer, m units in the hidden layer, and one unit in the output layer. The wastewater treatment system is used as the control object and the DO concentration is used as the control variable. The hybrid algorithms are described as follows.

Preliminary. The model used in this work is a feedforward neural network. The network structure consists on an input layer, a hidden layer and an output layer.

The output of each neuron in the hidden layer is as follows:

$$o_j = \frac{1}{1 + \exp(-\mathrm{net}_j)}. \tag{3}$$

$$net_j = \sum_i w_{ji} \cdot x_i - q_j, \mathrm{i} = 1,2,3,...,n, \mathrm{j} = 1,2,...,\mathrm{L}. \tag{4}$$

where o_j is the output of jth neuron in the hidden layer, x_i is the ith input variable, $x = [\mathrm{X,S,O}]$, w_{ji} is the connection weight between the ith neuron in the input layer and the jth neuron in the hidden layer, q_j

is the threshold of the jth neuron in the hidden layer, n and L are the numbers of neurons in the input and hidden layers, respectively. The output of each neuron in the output layer is as follows:

$$\hat{y}_k = \frac{1}{1+\exp(-\mathrm{net}_k)}. \tag{5}$$

$$net_k = \sum_j w_{kj} \cdot o_j - q_k, \mathrm{k} = 1,2,3,...,\mathrm{m}. \tag{6}$$

where w_{kj} is the connection weight between the jth neuron in the hidden layer and the kth neuron in the output layer, q_k is the threshold of the kth neuron in the output layer, m is the numbers of neurons in the output layer.

$$E = \frac{1}{2}(\hat{\mathrm{y}}_k - \mathrm{y}_d). \tag{7}$$

where $\hat{\mathrm{y}}_k$ and y_d are the desired and the current outputs of the network. And as the process in (2) only has one output, so k=1. The FNN with at least one hidden layer has the capability to approximate any desired nonlinear function to an arbitrary degree of accuracy [23].

Training algorithm. We set this network with three layers: an input layer, a hidden layer and an output layer. we use a FN-based hybrid training algorithm which combines unsupervised and supervised training in one procedure[18]. The specifics of the process followed in this algorithm is as follows:

First stage: constructing the self-organized layer

The distance threshold r controls neuron insertion and, therefore, determines implicitly the number of neurons in the self organizing map $r \in [0,1]$. The procedure can be described in the following six steps.

Step 1a: Set the number of neuron in the hidden layer to 1, and initialize its weights with the first input vector.

Step 2a: Present an input vector x to the FNN.

Step 3a: Calculate the distance between the input vector x and the weight vector w_n of neuron n. The distance measurement can be defined as:

$$D(x-w_n) = \mathbf{P}\, x - w_n \,\mathbf{P}. \tag{8}$$

where $\mathbf{P}\,.\,\mathbf{P}$ denotes the Euclidean distance metric.

Step 4a: If $D(x-w_n)$ **f** r , the neuron is not close enough to the input vector, then a new neuron is added in the network (n = n + 1).The connection weight vector of the newly generated neuron, w_{n+1}, is then initialized to be x (w_{n+1}= x), else, go back to Step 2a and repeat until all inputs have been presented.

Step 5a: Initialize the weights of the output layer at small random values.

Step 6a: Go to the second stage.

Second stage: the procedure of training algorithm

Step 1b: Present an input vector x to the FNN.

Step 2b: The winner neuron b is selected by evaluation of the distance measure between the input and the neuron weights[19]:

$$\mathbf{P}\, x - w_b \,\mathbf{P} = \min \mathbf{P}\, \mathrm{x} - w_b \,\mathbf{P}. \tag{9}$$

The distance measure can be any distance norm, in the practice mostly the Euclidean one is used.

Step 3b:Calculate

$$e(\mathrm{t}) = e_0 \exp(-\frac{t}{T}). \tag{10}$$

where e_0 is its initial value, T is the total number of iterations and t is the current iteration.

Step 4b: Calculate

$$s\,(\mathrm{t}) = s_0 (\frac{s_f}{s_0})^{\frac{t}{T}}. \tag{11}$$

where s_0 and s_f control the initial and final values of neighbourhood width.

Step 5b: A Gaussian neighbourhood function is calculated as follows:

$$h_{bi}(\mathrm{t}) = e(\mathrm{t}) \exp(-\frac{(\mathrm{b}-\mathrm{i})^2}{2s^2}), h_{bi}(\mathrm{t}) \in [0,1]. \tag{12}$$

where b denotes the index of the winner neuron and i denotes the index of any neuron. The factors $e(\mathrm{t})$ and $s\,(\mathrm{t})$ are the learning rate factor and the neighbourhood width factor, respectively. They are decreasing functions of time. The neighborhood function usually $h_{bi}(\mathrm{t})$ is equal to 1 for the winner neuron and decrease with the distance of the neurons from the winner.

Step 6b: The weights of the winner and its neighbouring neurons are modified as follows:

$$\Delta w_{ji} = h_{bi}(\mathrm{t})(\mathrm{x} - w_{ji}(\mathrm{t})). \tag{13}$$

The weights leaving from the winner neuron and its neighbours are modified based on the gradient descent method [24] :

$$\Delta w_{kj}(\mathrm{t}) = -a\,(\mathrm{t})(\mathrm{y}_d - \hat{y}_k)\hat{y}_k(1-\hat{y}_k)\,\mathrm{o}_j + b\,(\mathrm{t})\Delta w_{kj}(\mathrm{t}-1). \tag{14}$$

where $a\,(\mathrm{t})$ is the learning rate; $b\,(\mathrm{t})$ is the momentum rate; y_d and $\hat{y}_k$ are the desired and actual outputs, respectively; o_j is the output of jth neuron in the hidden layer.

Step 7b: The algorithm is stopped when the value of the error function (7) has become sufficiently small, otherwise go back to *step1b*.

It is not absolutely necessary to execute the first stage at every iteration. It is sufficient to initialize the hidden and the output layers.

The hybrid algorithm above updates only parameters of the selected hidden nodes after every input–output case.

When a new input vector is presented to the neural network, the adjustment of weights does not damage the acquired performance of the neural network. In fact, if this example is different from the other, it will activate another neuron and only its weight will be adjusted. The network preserves its plasticity, because it can always react to unknown inputs and its stability since the network performances are not damaged by the new examples.

The proposed algorithm uses the basic concepts of self organization map but has a dynamic structure. It also uses the distance threshold parameter to generate the hidden layer neurons. This process improves the initial input layer weights and the adaptive neurons of the hidden layer.

Simulation result

In this section, simulation results are presented in order to evaluate the performance of the hybrid algorithm and to compare it with RBF algorithms.

To verification the effectiveness of this new control method, we use the parameters for the proposed algorithm are [18]: e_0=0.5, a =0.15, b =0.15, s_0=2, s_f = 0.5. We applied the new algorithm to the control of dissolved oxygen in the wastewater treatment system. The initial value of x = [150 200 1.5]. The parameters in formula (2) are chosen as $\hat{m}$ = 2, Y_{NH} = 0.7, C =2, k_d = 0.06,

f = 0.66, f_x = 1.41, k_o =2, k_s = 79 and set the Do concentration 2mg/L. The simulation curve as Fig 1.

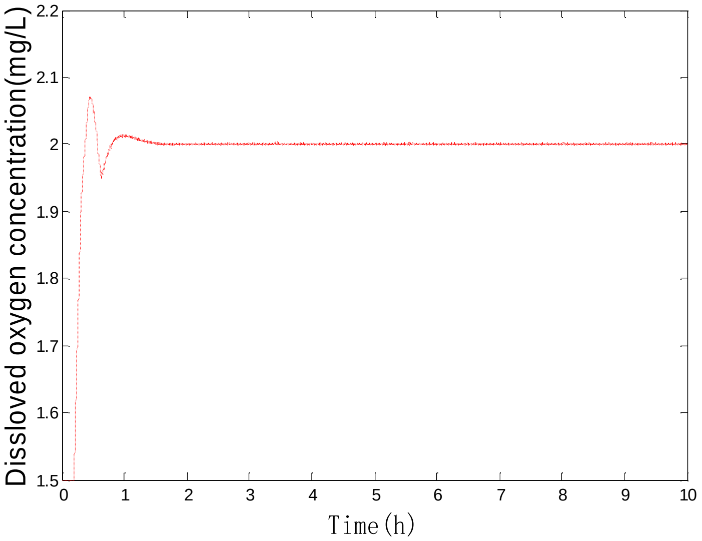

Fig.1 The control of dissloved oxygen concentration

It can be seen that the curve has overshoot, rising fast and after a transitional period the steady-state error become zero and the dissolved oxygen concentration has reach the set value in the limited time. So the hybrid algorithm is feasible and effective for the dissolved oxygen concentration control.

We compared the RBF which is used to approach unknown nonlinear function in [13]. We set the training goal of the neural network to 0, twenty hidden neurons, one output and the learning rate is 0.1. The Gaussian function is used as radial basis function, the initial weights of RBF is $w = 0$. The training result of RBF and the hybrid are shown in Fig 2 and Fig 3, respectively.

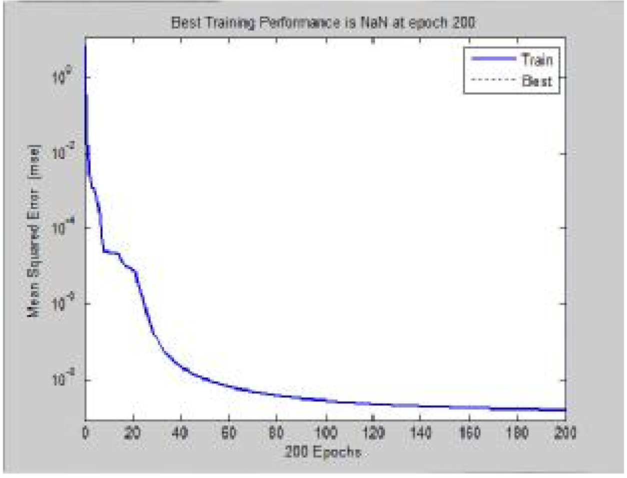

Fig.2 The evolution of iteration for RBF

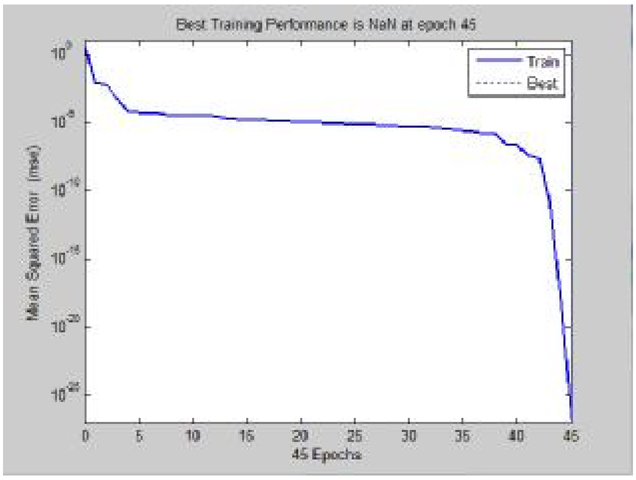

Fig.3 The evolution of iteration for hybrid

As it can be seen, in the curve the best line is not reveal, because the goal which we set is 0. The Fig.2 shows that after about 200 times iteration the error become stabilize, but in Fig.3 we can see that after 45 times iteration the curve is close to 0, it is more accurate. At the Fig 2, when the line become stabilize, we can see that the MSE is about 10^{-9} ,but from Fig 3, when the line reaches the same MSE the iteration is about 42 times. So we draw a conclusion that the iteration speed of the hybrid is faster than RBF under the same training goal.

Conclusion

A new neural control method was presented to solve the DO concentration control problem in the activated sludge process. The method is based on the concept of combining Kohonen algorithm and gradient descent method. The key ideas explored are the use of the Kohonen algorithm for training the weights of the hidden layers and gradient descent method for training the weights of the output layer. Simulation results have demonstrated the effectiveness of the proposed algorithm in controlling the dissolved oxygen concentration and improving the convergence speed for iteration.

Acknowledgments

This paper was supported by the Education Department fund of Hebei Province under Grant Z2010136, and the Natural Science fund of Hebei Province under Grant F2012208075 and F2014208119.

References

[1] B. Chachuat, N. Roche, M.A. Latifi, Optimal aeration control of industrial alter-nating activated sludge pants, Biochemical Engineering Journal 23 (2005) 277–289.

[2] B. Holenda, E. Domokos, A. Redey, J. Fazakas, Aeration optimization of a wastewater treatment plant using genetic algorithm, Optimal Control Applications and Methods 28 (2007) 191–208.

[3] Petre E, Marin C, Selisteanu D. Adaptive control strategies for a class of recycled depollution bioprocesses[J]. Journal of Control Engineering and Applied Informatics, Vol.7, No.2, 25-33, 2005.

[4] Petre E, Selişteanu D. A multivariable robust-adaptive control strategy for a recycled wastewater treatment bioprocess[J]. Chemical Engineering Science, Vol.90, No. 7, 40-50, 2013.

[5] Y. Zhang, T. Chai, H. Wang, A nonlinear control method based on ANFIS and multiple models for a class of SISO nonlinear systems and its application, IEEE Trans. Neural Netw. 22 (11) (2011) 1783–1795.

[6] Y.M. Li, S.C. Tong, Y. Liu, T. Li, Adaptive fuzzy robust output feedback control of nonlinear systems with unknown dead zones based on a small gain approach, IEEE Trans. Fuzzy Syst. 22 (1) (2014) 164–176.

[7] S.C. Tong, Y.M. Li, H. G. Zhang, Adaptive neural network decentralized back-stepping output-feedback control for nonlinear large scale systems with time delays, IEEE Trans. Neural Netw.22 (7) (2011) 1073–1086.

[8] M. Chen, S.S. Ge, B.V.E. How, Robust adaptive neural network control for a class of uncertain MIMO nonlinear systems with input nonlinearities, IEEE Trans. Neural Netw. 21(5) (2010) 796–812.

[9] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (1989) 359–366.

[10] Selami Beyhan, Musa Alci, Mixed Structured RBF Network for Direct Inverse Control of Nonlinear syetems, 9781-4244-3428-2/09/2009 IEEE.

[11] Tomohiro Ando, Sadanori Konishi, Seiya Imoto. Nonlinear regression modeling via regularized radial basis function networks[J]. Journal of Statistical Planning and Inference, 2008:1-18.

[12] Zeyu Li, Jiangqing Hu, Xingxing Huo, PID control based on RBF neural network for ship steering, Proceedings of the IEEE (2012) 1076 – 1080.

[13] Lin Mei-jin, Luo Fei, A Nonlinear Adaptive Control Approach for an Activated Sludge Process Using Neural Networks, Proceedings of the IEEE (2014) 2435 - 2440

[14] G. Lera, M. Pinzolas, Neighborhood based Levenberg Marquardt algorithm for neural network training, IEEE Trans. Neural Netw. 13 (5) (2002) 1200–1203.

[15] B.M. Wilamowski, H. Yu, Improved computation for Levenberg-Marquart,IEEE Trans. Neural Netw. 21 (6) (2010) 930–937.

[16] P. Liu, H. Li, Efficient learning algorithms for three-layer regular feedforward fuzzy neural networks, IEEE Trans. Neural Netw. 15 (3) (2004) 545–558.

[17] M. Ben Nasr, M. Chtourou, A hybrid training algorithm for feedforward neural networks, Neural Process. Lett. 24(2)(2006) 107–117.

[18] Mounir Ben Nasr, Mohamed Chtourou, Neural network control of nonlinear dynamic systems using hybrid algorithm, Applied Soft Computing 24 (2014) 423-431.

[19] Mounir Ben Nasr, Mohamed Chtourou, A self-organizing map-based initialization for hybrid training of feedforward neural networks,Applied Soft Computing 11 (2011) 4458-4464.

[20] M. Ben Nasr, M. Chtourou, A fuzzy neighborhood-based training algorithm for feedforward neural networks, Neural Computing and Application 18 (2) (2009)127–133.

[21] T. Kohonen, The self-organizing map, Proceedings of the IEEE 78 (9) (1990)1464–1480

[22] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation Parallel Distributed Processing: Explorations in the Microstructures of Cognition, vol. 1, MIT Press, 1986, pp. 318–362.

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

[24] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation Parallel Distributed Processing: Explorations in the Microstructures of Cognition, vol. 1, MIT Press, 1986, pp. 318–362.