Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 20(5): 9489–9510.
DOI: 10.3934/mbe.2023417
Received: 18 December 2022
Revised: 24 February 2023
Accepted: 02 March 2023
Published: 20 March 2023

Research article

Adopting improved Adam optimizer to train dendritic neuron model for water quality prediction

Jing Cao[1], Dong Zhao[2], Chenlei Tian[1], Ting Jin[1,*] and Fei Song[1,*]

[1] College of Science, Nanjing Forestry University, Nanjing 210037, Jiangsu, China
[2] Wuxi Guotong Environmental Testing Technology, Co., Ltd, 214191, Jiangsu, China

* **Correspondence:** Email: tingjin@njfu.edu.cn, songfei@njfu.edu.cn.

Abstract: As one of continuous concern all over the world, the problem of water quality may cause diseases and poisoning and even endanger people's lives. Therefore, the prediction of water quality is of great significance to the efficient management of water resources. However, existing prediction algorithms not only require more operation time but also have low accuracy. In recent years, neural networks are widely used to predict water quality, and the computational power of individual neurons has attracted more and more attention. The main content of this research is to use a novel dendritic neuron model (DNM) to predict water quality. In DNM, dendrites combine synapses of different states instead of simple linear weighting, which has a better fitting ability compared with traditional neural networks. In addition, a recent optimization algorithm called AMSGrad (Adaptive Gradient Method) has been introduced to improve the performance of the Adam dendritic neuron model (ADNM). The performance of ADNM is compared with that of traditional neural networks, and the simulation results show that ADNM is better than traditional neural networks in mean square error, root mean square error and other indicators. Furthermore, the stability and accuracy of ADNM are better than those of other conventional models. Based on trained neural networks, policymakers and managers can use the model to predict the water quality. Real-time water quality level at the monitoring site can be presented so that measures can be taken to avoid diseases caused by water quality problems.

Keywords: dendritic neuron model; optimization algorithm; water quality prediction; machine learning

1. Introduction

Water is the material basis for the survival of human beings and all living things, and it is also an indispensable natural resource for the development of human civilization. However, with the development of the economy and the growth of population, the water resources available for are increasingly

scarce. Water pollution will further aggravate the shortage of water resources. In order to make effective use of water resources and reduce water pollution, the degree of water pollution needs to be predicted in time to manage water resources efficiently [1]. Water quality prediction can be mainly divided into two types of problems. One is the prediction of related water quality indicators based on time series data [2], and the other is the regression prediction of related water quality indicators based on cross-sectional data.

Many statistical models have been used to solve the problem of water quality prediction, such as multiple regression [3] and the finite element method [4]. However, the prediction accuracy of these traditional models is unsatisfactory, and it is difficult to use them to solve the predicting problems of large data sets and complex relationships between different independent variables. With the advancement of the internet and artificial intelligence, more and more machine learning methods have been proposed to predict water quality [5–11], and a series of prediction tools represented by artificial neural networks (ANNs) has attracted the most attention. For regression problems, a tree-based model has been used to solve the problem of water quality prediction. XGBoost and random forest, as classical tree models, have been successfully applied to predict microbial water quality [12]. At the same time, in [13], the author compared eight common machine learning algorithms in the coastal water quality assessment, and the results show that XGBoost model had the most stable performance and significantly reduced the uncertainty in predicting water quality index (WQI). The history of the neural network can be traced back to the first ANN theory model proposed by psychologist McCulloch and the mathematical logician Pitts in 1943 [14], which ushered in the era of neuroscience theory and ANN research. Later, Rosenblatt [15] invented the perceptron and its learning algorithm, which was the first practical application of ANN and demonstrated the pattern recognition ability of ANNs. However, in 1969, Minsky and Papert stated in Perception that ANNs could not solve nonlinear problems, and then the development of neural networks gradually slowed down. After that, with the further development of computer science and error backpropagation algorithms [16], the training of a multi-layer neural network became possible, and then the deep neural network in the 21st century has pushed ANNs fully forward. Nowadays, various networks have been proposed and used to solve practical problems [17–19]. For example, the VGG-16 network, which was runner-up in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) competition in 2014, has 13 convolutional layers and 3 fully connected layers with a large number of parameters, and the deep residual network won the first place in the ILSVRC competition classification task in 2015. They are both famous models in deep learning [20].

There have been some studies researches on ANNs for water quality prediction. For example, ANNs have been proposed to predict the water quality of urban drinking water [21]. Based on the results of ANNs prediction, managers can make relevant decisions and maintenance measures. ANNs and support vector machine were used as comparison algorithms to predict the water quality of sewage treatment plants. Compared with support vector machine, optimized ANNs can produce stable prediction performance [7]. In addition, there have been numerous studies using neural networks to predict water quality [22–25]. For water quality prediction based on time series, long short-term memory (LSTM) has attracted the most attention, and the excellent prediction ability of LSTM has been reflected in water quality prediction [26–29]. LSTM is good at dealing with time series problems, but it does not perform as well as conventional methods in cross-sectional data. Although ANN model can complete the task of water quality prediction, the traditional ANN model based on gradient de-

scent method has the disadvantages of slow convergence ability and high computational cost [30], and thus it is rarely used to solve practical problems. In addition, some conventional machine learning algorithms have been used in water quality prediction [13], such as random forest [31, 32], decision tree [33, 34], K-Nearest Neighbor (KNN) [35] and ensemble learning [12, 36], all of which have been proved to be feasible.

Conventional ANN models are based on the complex combination of multiple neurons, with little research on the mapping capability of a single neuron. At the same time, using dendrite structure as the basic neuron model's construction is being widely studied [37, 38]. The single neuron has a lot of interconnections on a dendritic unit, and the capacity of dendritic computation can well explain the multiple functions of neurons. Compared to the structure based McCulloch-Pitts neurons, dendritic structures have more possibility on various computational tasks. DNM based on the characteristics of dendritic structures [39–41] was proposed in [42]. A single neuron as a synapse can be nonlinearly mapped by the Sigmoid function, which has been proven to be able to approximate any complex continuous function [43, 44]. The nonlinear effect in DNM is not only the Sigmoid function mapping of synaptic layer but also the Boolean logic of dendritic layer and membrane layer, namely, AND, OR and NOT, which provides stronger fitting ability for the whole DNM without great time loss. These are simple additions and multiplications, not complex functions. DNM was originally used to solve classification problems and achieved good results [45–47]. With the help of time delay and embedding size technology, DNM has also been successfully applied to solve time series problems, and the prediction effect is highly competitive compared with the classical LSTM algorithm [48–51]. There are few studies on regression based on cross-section data in DNM, and it is very common in real life. However, the powerful nonlinear fitting ability of DNM makes it possible to solve such regression problems. Furthermore, the learning algorithm is very important for the efficiency of DNM. A good learning algorithm can make DNM obtain faster and more stable convergence results. Gradient descent method is used in the training of DNM in [45, 46]. The weight is updated by subtracting the product of its gradient and the factor learning rate from the current weight. However, a fixed learning step will influence the learning result. In this research, a recently proposed optimization algorithm, called AMSGrad [52], is a variant of Adam [53] and has been used to train DNM [54]. It is superior to gradient descent on regression tasks. Therefore, AMSGrad algorithm is applied to train DNM, so as to achieve better accuracy in water quality prediction.

In this research, DNM is utilized with enhanced fitting capability to address the regression problem and is applied to a prevalent water quality prediction problem. The primary contribution of this study is that the proposed ADNM is the new attempt to compare water quality prediction with other machine learning techniques. Its excellent performance would catch more researchers' attention in the future. Section 2 introduces the DNM and its learning algorithm. Section 3 describes the whole regression problem and gives the experimental results and analysis. Section 4 is a summary of the whole paper.

2. Prediction model and learning algorithm

This section mainly contains two parts. The first part is the prediction model, and the other is the learning algorithm of DNM.

2.1. Dendritic neuron model

Compared with the traditional ANN model, the biggest characteristic of DNM is that it does not build a model with the help of multiple identical neuronal structures. There are nonlinear interactions between synapses and dendrites in DNM, which include conventional logical operations such as AND, OR and NOT. With the help of a simple logic operation, a more innovative neuron model can be developed. The most important feature of synapses is that they can automatically learn neuron morphology according to the input, including forming a forward input, backward output and constant output. In the process of network learning, the useless dendrites are automatically discarded, while the useful ones are retained and participate in the supervised classification problem. The structure of the ANN and DNM network is shown in Figure 1. The orange circles represent the input. DNM has four dendritic layers, and each layer has four synapses. The blue circles represent the membrane layer, which is used to multiply the signal of each dendritic layer, and the final soma layer is used to Sigmoid map the input.

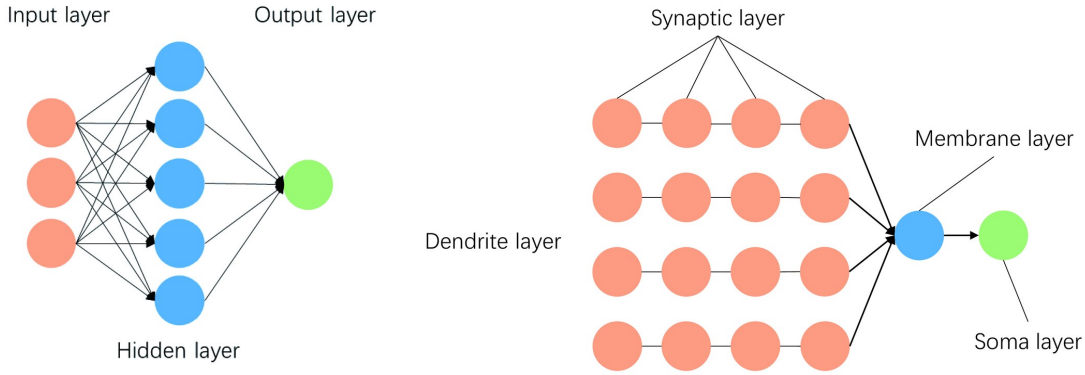


Figure 1. Model morphology of the ANN and DNM.

Synaptic function: In this layer, the Sigmoid function is proposed to simulate the function mapping of the Synaptic layer. Input values are first calculated linearly with parameters and then obtained by the Sigmoid function mapping. The specific expression of this layer is

$$Y_{ij} = \frac{1}{1 + e^{-k(w_{ij}x_i - \theta_{ij})}}. \tag{2.1}$$

k is a natural number, and the input x_i represents the input of the i-th synapse, $(i = 1, 2, ..., N)$. N is the dimension of input, that is, the number of independent variables. w_{ij} and θ_{ij} correspond to the parameter values on the i-th synapse, the j-th dendrite, $(j = 1, 2, ..., M)$. M is the number of dendritic layers, and different parameter values M can be set for different dendritic layers. Synapses can learn to obtain different parameter combinations, resulting in four types of knots with different properties, which are introduced as follows:

- Constant 0 connection: There are two types of constant 0 connections, where the output of the synapse is always approximately equal to 0 when the parameter combination satisfies $w_{ij} < 0 < \theta_{ij}$ and $0 < w_{ij} < \theta_{ij}$, and the input is between 0 and 1.
- Constant 1 connection: There are two types of constant 1 connections, where the output of the synapse is always approximately equal to 1 when the parameter combination satisfies $\theta_{ij} < w_{ij} < 0$ and $\theta_{ij} < 0 < w_{ij}$, and the input is between 0 and 1.

- Excitatory connection: There is a type of Excitatory connection, where the output of the synapse is positively related to the input when the parameter combination satisfies $0 < \theta_{ij} < w_{ij}$, and the input is between 0 and 1.
- Inhibitory connection: There is only one kind of Inhibitory connection, where the output of the synapse will be negatively correlated with the input when the parameter combination satisfies $w_{ij} < \theta_{ij} < 0$, and the input is between 0 and 1.

All of the above cases are shown in Figure 2.

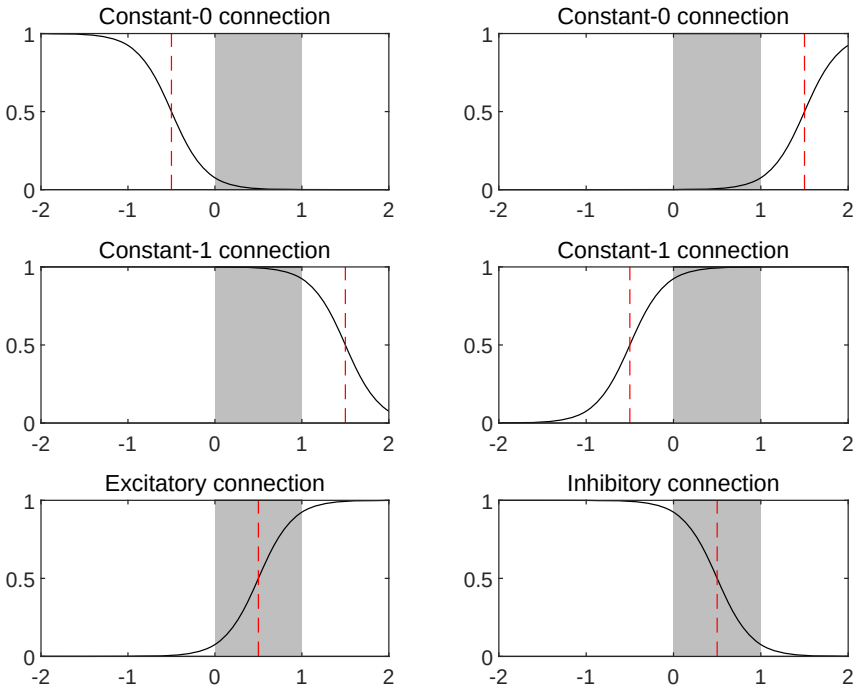


Figure 2. Six parameter combinations in the synaptic layer.

Dendrite layer: The dendritic layer will perform a multiplication function for the output of the synapse. Since the input is divided into four categories at the synaptic layer, synapses with constant 0 output will affect the whole dendrite, and the output of the dendritic layer will become 0 after continuous multiplication. Similarly, synapses with constant 1 output will be ignored by dendrites, that is, constant 1 does not affect any conjunction calculation. The unique processing method of the dendritic layer enables DNM to have the pruning function, that is, the dendrites with the output of constant 0 will be pruned, while the synapses with the output of constant 1 will be ignored. Through these two pruning operations, a more specific network structure can be formed. The concatenation operation of the j-th dendrite layer is described as

$$Z_j = \prod_{i=1}^{n} Y_{ij}. \tag{2.2}$$

Membrane layer: The dendritic layer multiplies the values of all the synapses, and the membrane layer sums the results of all the dendritic layers. The addition function refers to Eq (2.3). b_j denotes

the weight of the j-th dendritic layer. In the classification problem, b_j takes the value 1. The OR and AND logic functions can be realized by the multiplication and addition operations between the dendritic layer and the membrane layer, and the signal from the membrane layer will be sent to the next layer to deactivate the soma body

$$V = \sum_{j=1}^{M} b_j Z_j. \tag{2.3}$$

In this paper, b is regarded as a parameter to be learned.

Soma layer: The soma layer is the last layer of DNM, which combines the results of all the previous multiplications and additions, compares them to a threshold and then uses a sigmoid function for calculation, that is

$$O = \frac{1}{1 + e^{-k_s(V - Q_s)}}. \tag{2.4}$$

The value of Q_s is between 0 and 1. If DNM is used to solve the binary classification problem, it is best to set the value to 0.5. In the regression problem of this research, we will delete the soma layer and take the output of the membrane layer as the result of the last layer. In addition, the parameters to be learned are also changed from the original two to three, that is, the coefficient b_j of the membrane layer is newly added.

2.2. Learning algorithm

As we introduced above, the traditional DNM mainly has two parameters, namely, the independent variable coefficient w and the bias coefficient θ of the synapse layer. The most intelligent part of the neural network is that it can automatically learn the parameters of the whole system according to the given data. For any set of inputs, a set of output values can be obtained by linear operation of parameter w and θ and then nonlinear mapping of synaptic layer. In this research, the biggest difference with the conventional DNM is that the output of the soma layer and the membrane layer should be optimized, so that one more parameter in the membrane layer, namely, the above parameter b, should be learned in the membrane layer. The essence of the error back-propagation algorithm is to find the chain derivative of the function and constantly modify the connection parameters by calculating the gradient value, so as to continuously improve the prediction effect of DNM. The AMSGrad algorithm is used to train the DNM model. The error between the network output and the real output can be expressed as

$$E = \frac{1}{2}(T - V)^2. \tag{2.5}$$

The output V of the input value is compared with the real data T, and the w, θ and b are continuously corrected by the AMSGrad algorithm to make the error E as small as possible. We define the partial derivative of E about the parameters w_{ij}, θ_{ij} and b_j to be

$$\delta w_{ij} = \frac{\partial E}{\partial w_{ij}}, \tag{2.6}$$

$$\delta \theta_{ij} = \frac{\partial E}{\partial \theta_{ij}}, \tag{2.7}$$

$$\delta b_j = \frac{\partial E}{\partial b_j}. \tag{2.8}$$

With the help of the chain derivative rule, we can give the calculation methods of $\frac{\partial E}{\partial w_{ij}}$, $\frac{\partial E}{\partial b_j}$ and $\frac{\partial E}{\partial \theta_{ij}}$,

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial V} \cdot \frac{\partial V}{\partial Z_j} \cdot \frac{\partial Z_j}{\partial Y_{ij}} \cdot \frac{\partial Y_{ij}}{\partial w_{ij}}, \tag{2.9}$$

$$\frac{\partial E}{\partial \theta_{ij}} = \frac{\partial E}{\partial V} \cdot \frac{\partial V}{\partial Z_j} \cdot \frac{\partial Z_j}{\partial Y_{ij}} \cdot \frac{\partial Y_{ij}}{\partial \theta_{ij}}, \tag{2.10}$$

$$\frac{\partial E}{\partial b_j} = \frac{\partial E}{\partial V} \cdot \frac{\partial V}{\partial b_j}. \tag{2.11}$$

The specific solution formulas in the above are

$$\frac{\partial E}{\partial V} = V - T, \tag{2.12}$$

$$\frac{\partial V}{\partial Z_j} = b_j, \tag{2.13}$$

$$\frac{\partial Z_j}{\partial Y_{ij}} = \prod_{L=1 \text{ and } L \neq i}^{n} Y_{Lj}, \tag{2.14}$$

$$\frac{\partial Y_{ij}}{\partial w_{ij}} = \frac{k x_i e^{-k(x_i w_{ij} - \theta_{ij})}}{\left(1 + e^{-k(x_i w_{ij} - \theta_{ij})}\right)^2}, \tag{2.15}$$

$$\frac{\partial Y_{ij}}{\partial \theta_{ij}} = \frac{-k e^{-k(x_i w_{ij} - \theta_{ij})}}{\left(1 + e^{-k(x_i w_{ij} - \theta_{ij})}\right)^2}, \tag{2.16}$$

$$\frac{\partial V}{\partial b_j} = Z_j. \tag{2.17}$$

According to AMSGrad, we define $m_1(t)$, $m_2(t)$, $m_3(t)$ as the exponential moving average of the partial derivative for w_{ij}, θ_{ij}, b_j in the t-th iteration. Let $v_1(t)$, $v_2(t)$, $v_3(t)$ be the exponential moving average of squared partial derivatives for w_{ij}, θ_{ij}, b_j in the t-th iteration, given by the formulas

$$m_1(t) = \beta_1 \cdot m_1(t-1) + (1-\beta_1) \cdot \delta w_{ij}, \tag{2.18}$$

$$m_2(t) = \beta_1 \cdot m_2(t-1) + (1-\beta_1) \cdot \delta \theta_{ij}, \tag{2.19}$$

$$m_3(t) = \beta_1 \cdot m_3(t-1) + (1-\beta_1) \cdot \delta b_j, \tag{2.20}$$

$$v_1(t) = \beta_2 \cdot v_1(t-1) + (1-\beta_2) \cdot (\delta w_{ij})^2, \tag{2.21}$$

$$v_2(t) = \beta_2 \cdot v_2(t-1) + (1-\beta_2) \cdot (\delta \theta_{ij})^2, \tag{2.22}$$

$$v_3(t) = \beta_2 \cdot v_3(t-1) + (1-\beta_2) \cdot (\delta b_j)^2, \tag{2.23}$$

where β_1 and β_2 are constant. Their values are 0.9 and 0.999 [52]. Then, the update equations of these parameters are

$$\hat{v}_1(t) = \max(\hat{v}_1(t-1), v_1(t)), \tag{2.24}$$

$$\hat{v}_2(t) = \max(\hat{v}_2(t-1), v_2(t)), \tag{2.25}$$

$$\hat{v}_3(t) = \max(\hat{v}_3(t-1), v_3(t)), \tag{2.26}$$

$$w_{ij}(t+1) = w_{ij}(t) - \frac{\eta}{\sqrt{\hat{v}_1(t)+\varepsilon}} \cdot m_1(t), \tag{2.27}$$

$$\theta_{ij}(t+1) = \theta_{ij}(t) - \frac{\eta}{\sqrt{\hat{v}_2(t)+\varepsilon}} \cdot m_2(t), \tag{2.28}$$

$$b_j(t+1) = b_j(t) - \frac{\eta}{\sqrt{\hat{v}_3(t)+\varepsilon}} \cdot m_3(t), \tag{2.29}$$

where η is learning rate. ε is a constant, and $\varepsilon = 1.0 \times 10^{-5}$ in our experiment, which prevents the divisor of the learning rate from being 0. The conventional Back Propagation (BP) learning algorithm directly updates the parameters according to the following equations:

$$w_{ij}(t+1) = w_{ij}(t) - \eta \cdot \delta w_{ij}, \tag{2.30}$$

$$\theta_{ij}(t+1) = \theta_{ij}(t) - \eta \cdot \delta \theta_{ij}, \tag{2.31}$$

$$b_j(t+1) = b_j(t) - \eta \cdot \delta b_j. \tag{2.32}$$

3. Experiment and discussion

3.1. Data description

The data were obtained from the six township street of Huishan District, Wuxi City, Jiangsu Province, which is managed by Wuxi Guotong Environmental Testing Technology Co., Ltd. These villages and towns are Wuxi Huishan Economic and Technological Development Zone, Yanqiao, Qianzhou, Yuqi, Luoshe, Qianqiao and Yangshan. Measurer regularly tested the water quality of rivers in these towns and obtained water quality reports over a period of time. The original cross-sectional data only includes three periods, and they are from January 2020 to December 2020, from January 2021 to December 2021 and from January 2022 to May 2022. All the lost data caused by epidemic prevention, construction, remediation and other reasons will be deleted. The preprocessed data and feature information are presented in Table 1. The number of samples in the three periods are 1700, 1980 and 825, respectively. For the convenience of expression, we abbreviate these three sections of data as data sets in 2020–2022. The label for the regression problem is a numerical value known as the "Current Composite Pollution Index," which measures the overall quality of water. This index enables the manager to assess the level of pollution in the water area and make informed decisions to address it.

According to China's Environmental quality standards for surface water, water quality can be divided into three types: class III, IV and V, based on the value of the attributes of the current in the river. However, these attributes can be inconsistent in determining water quality standards. To overcome this issue, a "Current Composite Pollution Index" is used, which is a numerical value that provides a comprehensive assessment of water quality. The higher the value is, the worse the water quality. Machine learning technology can be used to develop a prediction model based on existing detection data, enabling the accurate prediction of the Current Composite Pollution Index. This allows the manager to make informed decisions regarding water quality and take appropriate measures to address any pollution issues.

Table 1. Feature information about the dataset.

Attributes	Value range (mg/L)	Mean	Variance	Feature description
Dissolved Oxygen (DO)	2.170–14.830	7.286	3.328	The total amount of molecular oxygen in air dissolved in water.
	2.300–11.900	6.956	1.853	
	4.200–16.000	7.461	2.183	
Ammonia Nitrogen	0.035–10.200	0.724	0.802	The total amount of nitrogen in the form of free ammonia (NH_3) and ammonium ions (NH_4^+).
	0.005–7.200	0.593	0.511	
	0.026–7.480	0.585	0.649	
Total phosphorus (TP)	0.010–1.180	0.149	0.015	The total content of phosphorus in water.
	0.010–3.740	0.154	0.033	
	0.010–0.770	0.122	0.009	
Permanganate index	0.900–11.600	3.805	2.172	The degree of water polluted by reducing agents.
	1.100–13.200	3.914	2.461	
	0.800–10.200	4.135	2.566	

3.2. Data normalization

The synaptic layer is a nonlinear mapping that requires input values to be in the range of 0 and 1, However, the value of the independent variable is a number greater than 1. To resolve this, we need to normalize the input. The specific formula is

$$X_i^{normalized} = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}}, \tag{3.1}$$

where X_{min} and X_{max} represent the minimum and maximum values of the corresponding data, respectively. Since the last layer of the DNM structure in this paper is deleted, the output value does not need data normalization, nor does it need data inverse normalization.

3.3. Evaluation metrics

To evaluate the prediction performance of different models, five typical regression indicators have been selected. These indicators are Mean Square Error (MSE), Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Correlation Coefficient (R). The calculation formulas for these indicators are

- MSE

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(V_i - T_i)^2, \tag{3.2}$$

- MAPE

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{T_i - V_i}{T_i}\right|, \tag{3.3}$$

- MAE

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|T_i - V_i|, \tag{3.4}$$

- RMSE

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(T_i - V_i)^2}, \tag{3.5}$$

- R

$$R = \frac{\sum_{i=1}^{n}\left(T_i - \overline{T_i}\right)\left(V_i - \overline{V_i}\right)}{\sqrt{\sum_{i=1}^{n}\left(T_i - \overline{T_i}\right)^2 \sum_{i=1}^{n}\left(V_i - \overline{V_i}\right)^2}}. \tag{3.6}$$

In the above formula, $\overline{T}$ is the average value of real data, and $\overline{V}$ is the average value of predicted data. T_i and V_i denote the $i-th$ desired value and predicted value. n means the number of the output set.

3.4. Parameters setting

In the DNM introduced in this paper, in addition to the hyperparameters M and k that are determined by the model, the hyperparameters of its learning algorithm also need to be taken into account. The traditional Gradient Descent Method has only one hyperparameter, which is the learning rate. AMSGrad introduces two additional hyperparameters, i.e., β and ε, but these hyperparameters have already been specified. The learning rate (η) of AMSGrad can be changed adaptively, so its initial value is set to the η of the gradient descent method. In summary, there are three parameters to be determined: the number of dendritic layer M, the constant of synaptic layer k and the learning rate η.

Since their values are in fact within an interval, it is necessary to traverse each interval value to obtain the best parameter combination, which will cause too much time loss. Therefore, we take the discrete value of equal length as the parameter to be selected. The number of dendritic layer M can be $\{1, 5, 10, 15\}$, the learning rate η can be {0.01, 0.001, 0.0001, 0.00001}, and the constant k can be {1, 5, 10, 15}. This results in 64 possible combinations, which would take too much time to evaluate. Considering that the optimal parameter combination can be obtained with the minimum number of experiments, an orthogonal experiment is a great choice. According to the orthogonality, we select some representative points from the comprehensive experiments [55]. Therefore, the orthogonal experiment in this paper will include 16 experiments, which are recorded as $L_{16}(4^3)$, and they are presented in the table. The optimal combination of hyperparameters is listed in Table 2. In order to find the optimal combination from the 16 parameter groups, we set the DNM as the parameters of the corresponding combination and conducted experiments on the three datasets. Their MSE results are shown in Table 3. As shown in the table, the best parameter combination for the 2020-dataset is group 12, for 2021-dataset is group 8 and for 2022-dataset is group 12.

Table 2. Orthogonal experimental group of parameters.

group	η	M	k	group	η	M	k
1	0.01	1	1	9	0.0001	1	10
2	0.01	5	5	10	0.0001	5	15
3	0.01	10	10	11	0.0001	10	1
4	0.01	15	15	12	0.0001	15	5
5	0.001	1	5	13	0.00001	1	15
6	0.001	5	10	14	0.00001	5	1
7	0.001	10	15	15	0.00001	10	5
8	0.001	15	1	16	0.00001	15	10

Table 3. The orthogonal experiment results of 16 parameter groups.

Group	2020-dataset	2021-dataset	2022-dataset
1	7.73E-00	7.23E-00	1.09E-01
2	7.18E-00	7.45E-00	5.73E-00
3	8.05E-00	6.26E-00	4.49E-00
4	8.15E-00	7.03E-00	5.27E-00
5	2.00E-00	2.19E-00	7.33E-01
6	1.56E-00	8.51E-00	4.18E-01
7	9.28E+01	4.26E-00	5.11E-01
8	7.90E-02	**1.21E-01**	9.92E-02
9	1.51E-00	1.97E-00	2.40E-01
10	1.95E-00	6.50E-01	3.13E-01
11	5.54E-01	1.00E-00	7.73E-01
12	**1.91E-02**	2.02E-01	**5.97E-02**
13	8.91E-01	1.68E-00	1.01E-00
14	1.57E-00	2.73E-00	1.09E-00
15	3.77E-01	1.60E-01	5.96E-01
16	1.97E-01	4.54E-01	5.08E-01

3.5. Model comparison

Table 4. Details of some compared models.

Basic model	Description & Parameters	Marked as
SVR	Sigmoid function kernel	SVR-s
SVR	Polynomial function kernel	SVR-p
SVR	RBF function kernel	SVR-r
DT	Improved bagging strategy TreeNumber = 50	RF
MLP	Iteration = 3000 Hidden neurons = 5	MLP
DT	Least-squares boosting Number of learning cycles = 100	GBDT
ELM	Hidden neurons = 5	ELM

In this section, seven regression models are selected for comparative experiments with ADNM (DNM-AMSGrad). They are the SVM with RBF kernel (R-SVM), the SVM with a polynomial kernel (P-SVM), the SVM with a sigmoid kernel (S-SVM), the MLP neural network [56], the random forests (RF), the gradient boosting decision tree (GBDT) [57], the extreme gradient boosting (XGBoost) [58], the extreme learning machine (ELM) [59] and the original DNM. GBDT uses the least square method to improve the model. The parameter settings of these models are shown in Table 4. The training set and test set of the three data sets are divided into 7:3. Meanwhile, in order to avoid extreme situations, the eight algorithms are run 30 times to take the average value, and the MSE, MAPE, MAE, RMSE

and R are counted. The iteration number of the learning algorithm is 3000, and the running number of MLP is set to be the same as that of DNM. Part of the experiments are conducted on a PC with a 2.30 GHz Intel (R) Core (TM) i7-10875H CPU using MATLAB R2021a.

Table 5. Evaluation metrics result of compared models on each data set.

Models	2020-dataset				
	MSE	MAPE	MAE	RMSE	R
ADNM	1.71E-03	6.52E-03	1.37E-02	3.48E-02	9.94E-01
DNM	4.50E-02	6.52E-02	1.25E-01	2.05E-01	9.90E-01
SVR-s	3.11E-01	1.55E-01	3.56E-01	5.55E-01	9.49E-01
SVR-r	6.39E-01	6.98E-02	2.28E-01	7.85E-01	8.65E-01
SVR-p	1.09E-01	1.10E-01	2.19E-01	2.59E-01	9.88E-01
RF	5.68E-02	4.19E-02	1.02E-01	2.32E-01	9.88E-01
MLP	1.65E-01	1.03E-01	2.13E-01	3.34E-01	9.53E-01
GBDT	1.02E-01	4.64E-02	1.30E-01	3.14E-01	9.77E-01
XGBoost	6.34E-02	4.11E-02	1.14E-01	2.48E-01	9.88E-01
ELM	1.68E-01	1.09E-01	2.23E-01	3.47E-01	9.60E-01
Models	2021-dataset				
	MSE	MAPE	MAE	RMSE	R
ADNM	3.76E-03	7.50E-03	1.54E-02	4.65E-02	9.97E-01
DNM	1.61E-01	1.55E-01	2.57E-01	3.91E-01	9.66E-01
SVR-s	5.78E-01	1.55E-01	3.55E-01	7.44E-01	8.83E-01
SVR-r	7.72E-01	6.10E-02	1.92E-01	8.56E-01	8.07E-01
SVR-p	4.50E-02	7.76E-02	1.43E-01	1.95E-01	9.88E-01
RF	1.78E-01	4.34E-02	1.09E-01	3.75E-01	9.63E-01
MLP	9.69E+01	3.08E+00	5.42E+00	6.73E+00	2.55E-01
GBDT	2.49E-01	5.46E-02	1.52E-01	4.82E-01	9.43E-01
XGBoost	1.42E-01	4.18E-02	1.19E-01	3.63E-01	9.76E-01
ELM	2.13E-01	1.08E-01	2.21E-01	4.12E-01	9.57E-01
Models	2022-dataset				
	MSE	MAPE	MAE	RMSE	R
ADNM	2.10E-02	6.08E-02	1.03E-01	1.44E-01	9.92E-01
DNM	7.19E-02	1.18E-01	1.88E-01	2.63E-01	9.75E-01
SVR-s	6.24E-01	2.38E-01	4.58E-01	7.78E-01	9.19E-01
SVR-r	7.62E-01	1.40E-01	3.18E-01	8.42E-01	7.54E-01
SVR-p	5.10E-01	2.83E-01	4.25E-01	6.32E-01	8.80E-01
RF	9.55E-02	8.21E-02	1.56E-01	2.96E-01	9.74E-01
MLP	1.06E-01	1.28E-01	2.19E-01	3.14E-01	9.64E-01
GBDT	1.08E-01	9.28E-02	1.82E-01	3.25E-01	9.64E-01
XGBoost	6.07E-02	8.10E-02	1.57E-01	2.42E-01	9.82E-01
ELM	1.03E-01	1.09E-01	1.94E-01	3.06E-01	9.74E-01

3.6. Results and discussion

Five evaluation metrics as the first result table are presented in Table 5. The lower the values of MSE, MAPE, MAE and RMSE are, the better the prediction performance of the model. The closer the R coefficient is to 1, the better the fitting effect of the model. From the evaluation index results alone, ADNM is optimal in all indexes of the three datasets, with the smallest MSE, MAPE, MAE, RMSE and the best R. ADNM even outperforms conventional algorithms by two orders of magnitude on some of the results. In addition, MLP fits poorly on the 2021-dataset, which may be a case of overfitting, but DNM does not have this problem.

Table 6. Friedman-test on all models.

Model	Rankings of MSE	Model	Rankings of MAPE	Model	Rankings of MAE	Model	Rankings of RMSE	Model	Rankings of R
ADNM	1	ADNM	1	ADNM	1	ADNM	1	ADNM	1
DNM	2	XGBoost	2	RF	2	XGBoost	2	XGBoost	2
XGBoost	3	RF	3	XGBoost	3	DNM	3	DNM	3
RF	4	GBDT	4	GBDT	4	RF	4	RF	4
SVR-p	5	DNM	5	DNM	5	SVR-p	5	SVR-p	5
GBDT	6	SVR-r	6	SVR-p	6	ELM	6	ELM	6
ELM	7	ELM	7	ELM	7	GBDT	7	GBDT	7
MLP	8	MLP	8	SVR-r	8	MLP	8	MLP	8
SVR-s	9	SVR-p	9	MLP	9	SVR-s	9	SVR-s	9
SVR-r	10	SVR-s	10	SVR-s	10	SVR-r	10	SVR-r	10

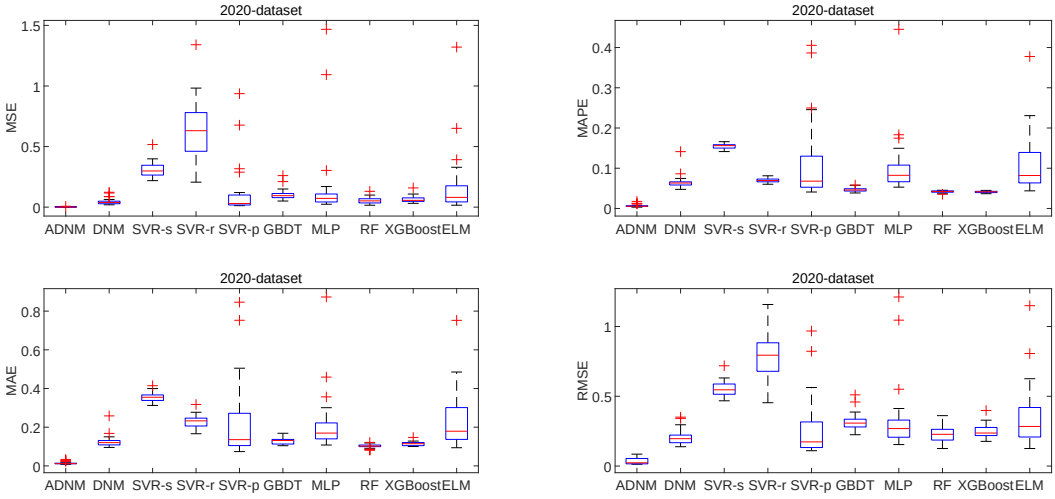


Figure 3. Box-and-whisker of evaluation metrics on 2020-dataset.

The Friedman-test [60] in Table 6 shows that ADNM is the first in all five evaluation criteria. Second, it can be seen that XGBoost and GBDT, two boost algorithms based on decision trees, also have good performance in processing cross section data. Compared to the multi-layer perceptron MLP

and the extreme learning machine ELM, the DNM has better performance, as evidenced by the test results.

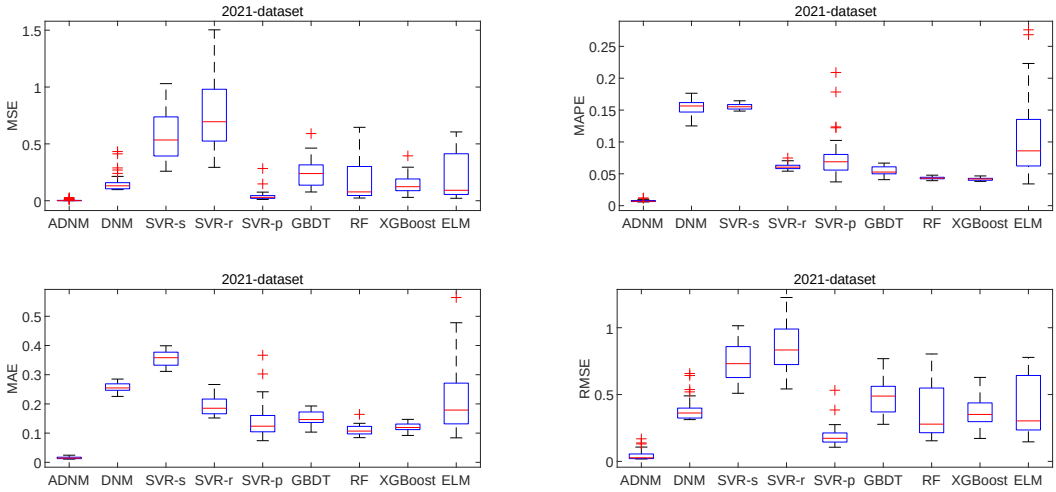


Figure 4. Box-and-whisker of evaluation metrics on 2021-dataset.

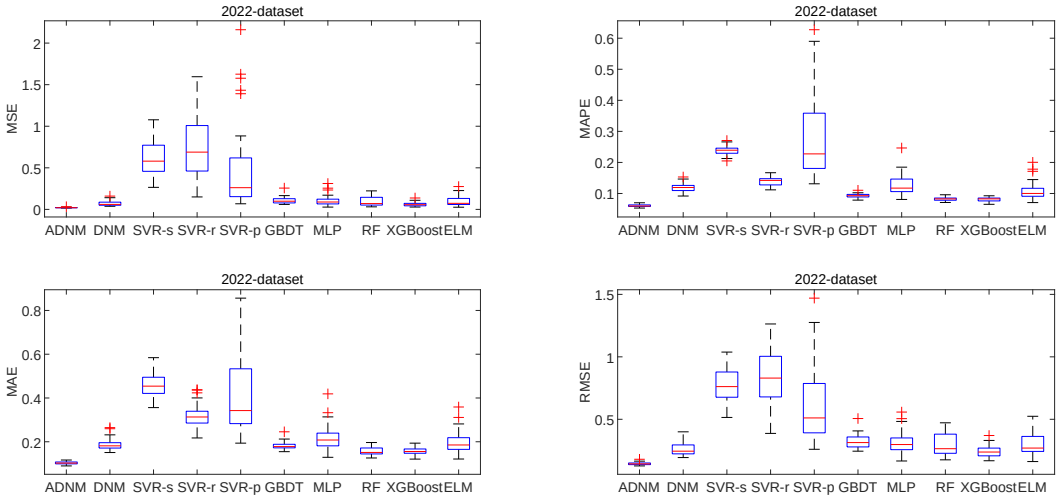
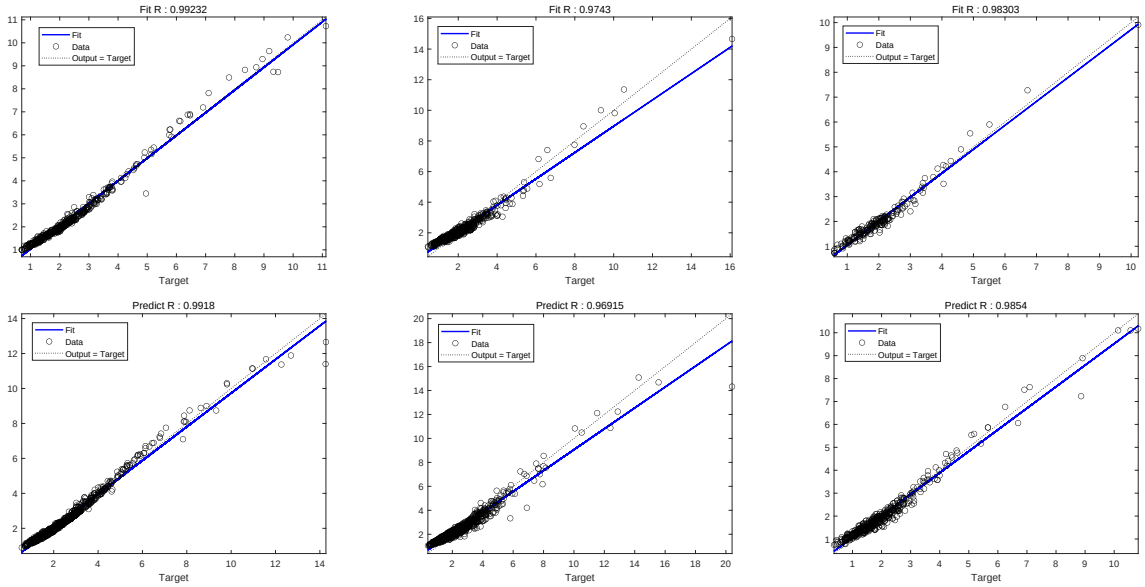


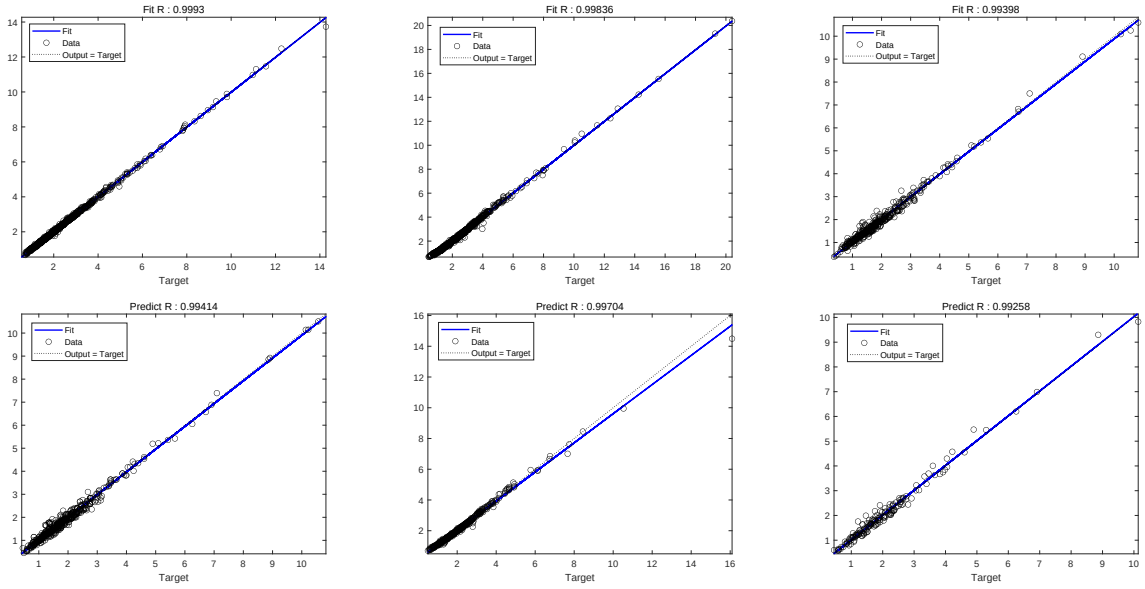
Figure 5. Box-and-whisker of evaluation metrics on 2022-dataset.

Figures 3–5 display box-and-whisker plots of each evaluation index for the three datasets, showing the maximum, minimum, first quarter, median, third quarter and extreme values. The training performance of MLP on the 2021-dataset is extremely poor, so it is not included in the comparison. From the figures, we can observe the distribution of the running results and determine the stability of the model's performance. The ADNM consistently achieves stable results in all training processes.



(a) 2020-dataset **(b)** 2021-dataset **(c)** 2022-dataset

Figure 6. Correlation coefficient graphs of DNM on 2020-dataset, 2021-dataset and 2022-dataset.



(a) 2020-dataset **(b)** 2021-dataset **(c)** 2022-dataset

Figure 7. Correlation coefficient graphs of DNM-AMSGrad on 2020-dataset, 2021-dataset and 2022-dataset.

In order to see the ability of ADNM to fit data more intuitively, we plotted the regression coefficients of ADNM and DNM on the three data sets, as are shown in Figures 6 and 7. The figure depicts the distribution of the predicted results of ADNM and DNM compared to the actual data. Moreover, AMS-

Grad algorithm can solve the issue of slow convergence rate of traditional gradient descent method, thereby enhancing prediction performance. The convergence curves of ADNM and DNM on 2020-dataset, 2021-dataset and 2022-dataset are shown in Figures 8 and 9. Although the maximum number of iterations is set to 3000, AMSGrad nearly converges in 200 generations, while the traditional algorithm needs 500 generations or more. In addition, the convergence accuracy of AMSGrad is better than that of the traditional algorithm.

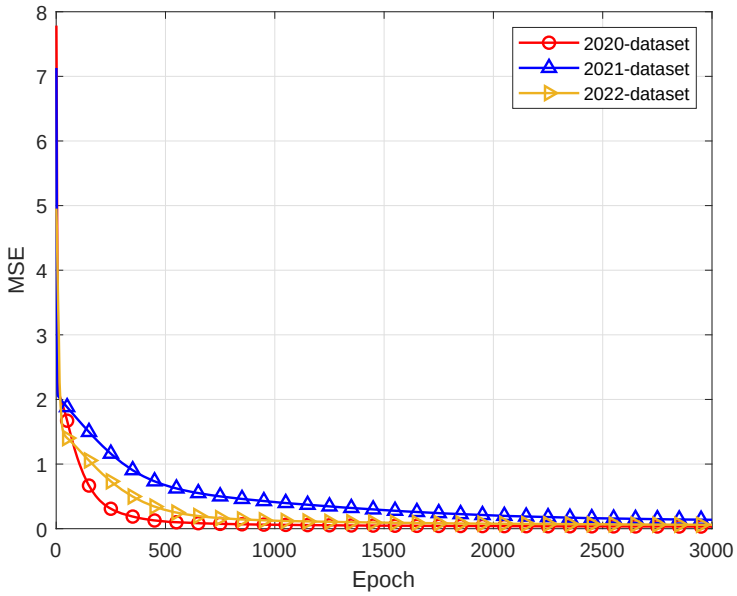


Figure 8. Convergence curves of DNM.

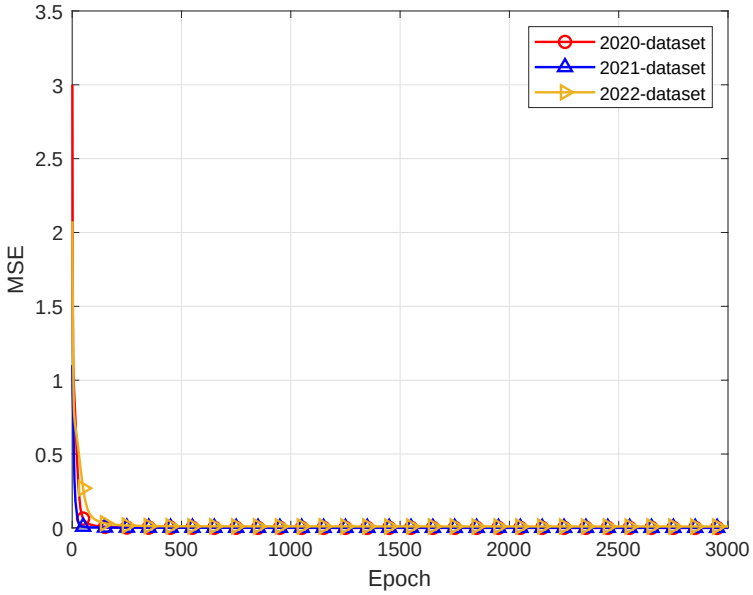


Figure 9. Convergence curves of DNM-AMSGrad.

It can be seen from the experimental results that, compared with other methods, ADNM can perfectly obtain the connection between different inputs without consuming too much computation with the help of the simplest nonlinear mapping operated by multiplication. Moreover, AMSGrad, as a prominent learning algorithm in recent years, is a great improvement to DNM. ADNM will be a stable and accurate tool for long-term water quality prediction.

4. Conclusions

In this research, we presented a new method for environmental prediction of water quality based on a single dendritic neuron model. This method has four layers of network structure, which are synaptic layer, dendritic layer, membrane layer and soma layer. Compared with the conventional ANN algorithm, DNM has additional nonlinear mapping of dendritic layer and membrane layer, so the fitting ability of DNM is definitely better than that of ANN, which can be proved by the experimental results in this paper. In addition, we use AMSGrad, an excellent Adam variant, to train DNM (ADNM). In order to determine the parameters of ADNM, orthogonal experimental design was used to determine the optimal parameters of ADNM on the three datasets. In the experiment, trained ADNM is used to forecast water quality, and its performance is compared with seven other methods. Five evaluation metrics are used to compare the learning effects of different models. ADNM has very good performance on the five metrics, and the Friedman-test can illustrate this point. We also plotted box-and-whisker plots and fitting curves of seven models on five metrics, and ADNM showed good results. Therefore, it is believed that ADNM is effective for water quality regression prediction. However, it is notable that this research is about the prediction problem of low dimension, small data. Compared with the deep learning techniques, for complex problems, ADNM cannot provide great results, and it is also restricted to a particular network structure. When the data dimension is too large, the gradient of DNM tends to disappear in the learning process and bring too much time loss. In addition, the study did not consider other necessary factors affecting water quality, which is also related to the data set provided. We also need to study the performance of ADNM in solving high dimensional problems. In the future work, we will try to replace the gradient algorithm with some specific excellent learning algorithms to overcome this weakness.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China Grants (12201303, 12201304 and 12071219) and the Natural Science Foundation of Jiangsu Province (no. BK20190745 and BK20210605). This work is also supported by the Academic Program Development of Jiangsu Higher Education Institutions (PAPD), the General Research Projects of Philosophy and Social Sciences in Colleges and Universities (2022SJYB0140), the Jiangsu Province Student Innovation Training Program (202110298040Z and 202210298050Z).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. T. Ma, N. Zhao, Y. Ni, J. Yi, J. P. Wilson, L. He, et al., China's improving inland surface water quality since 2003, *Sci. Adv.*, **6** (2020), eaau3798. https://doi.org/10.1126/sciadv.aau3798
2. N. Nemerow, *Scientific Stream Pollution Analysis*, Scripta Book Co., 1974.
3. O. Kisi, K. S. Parmar, Application of least square support vector machine and multivariate adaptive regression spline models in long term prediction of river water pollution, *J. Hydrol.*, **534** (2016), 104–112. https://doi.org/10.1016/j.jhydrol.2015.12.014
4. Y. Matsuda, A water pollution prediction system by the finite element method, *Adv. Water Resour.*, **2** (1979), 27–34. https://doi.org/10.1016/0309-1708(79)90004-6
5. G. Tan, J. Yan, C. Gao, S. Yang, Prediction of water quality time series data based on least squares support vector machine, *Procedia Eng.*, **31** (2012), 1194–1199. https://doi.org/10.1016/j.proeng.2012.01.1162
6. H. Chen, L. Xu, W. Ai, B. Lin, Q. Feng, K. Cai, Kernel functions embedded in support vector machine learning models for rapid water pollution assessment via near-infrared spectroscopy, *Sci. Total Environ.*, **714** (2020), 136765. https://doi.org/10.1016/j.scitotenv.2020.136765
7. S. Moni, E. Aziz, A. P. A. Majeed, M. Malek, The prediction of blue water footprint at Semambu water treatment plant by means of Artificial Neural Networks (ANN) and Support Vector Machine (SVM) models, *Phys. Chem. Earth*, **123** (2021), 103052. https://doi.org/10.1016/j.pce.2021.103052
8. Y. Khan, C. S. See, Predicting and analyzing water quality using machine learning: a comprehensive model, in *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, (2021), 1–6. https://doi.org/10.1109/LISAT.2016.7494106
9. M. Azrour, J. Mabrouki, G. Fattah, A. Guezzaz, F. Aziz, Machine learning algorithms for efficient water quality prediction, *Model. Earth Syst. Environ.*, **8** (2022), 2793–2801. https://doi.org/10.2166/wqrj.2022.004
10. N. Noori, L. Kalin, S. Isik, Water quality prediction using SWAT-ANN coupled approach, *J. Hydrol.*, **590** (2020), 125220. https://doi.org/10.1016/j.jhydrol.2020.125220
11. L. Kumar, M. S. Afzal, A. Ahmad, Prediction of water turbidity in a marine environment using machine learning: A case study of Hong Kong, *Reg. Stud. Mar. Sci.*, **52** (2022), 102260. https://doi.org/10.1016/j.rsma.2022.102260
12. L. Li, J. Qiao, G. Yu, L. Wang, H. Y. Li, C. Liao, et al., Interpretable tree-based ensemble model for predicting beach water quality, *Water Res.*, **211** (2022), 118078. https://doi.org/10.1016/j.watres.2022.118078
13. M. G. Uddin, S. Nash, M. T. M. Diganta, A. Rahman, A. I. Olbert, Robust machine learning algorithms for predicting coastal water quality index, *J. Environ. Manage.*, **321** (2022), 115923. https://doi.org/10.1016/j.jenvman.2022.115923
14. W. S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, *Bull. Math. Biol.*, **52** (1990), 99–115. https://doi.org/10.1007/BF02459570

15. F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, *Psychol. Rev.*, **65** (1958), 386. https://doi.org/10.1037/h0042519

16. D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, *Nature*, **323** (1986), 533–536. https://doi.org/10.1038/323533a0

17. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE*, **86** (1998), 2278–2324. https://doi.org/10.1109/5.726791

18. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, et al., Backpropagation applied to handwritten zip code recognition, *Neural Comput.*, **1** (1989), 541–551. https://doi.org/10.1162/neco.1989.1.4.541

19. T. Mikolov, M. Karafiát, L. Burget, J. Cernocky, S. Khudanpur, Recurrent neural network based language model, *Interspeech*, **2** (2010), 1045–1048.

20. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016), 770–778.

21. T. Dawood, E. Elwakil, H. M. Novoa, J. F. G. Delgado, Toward urban sustainability and clean potable water: Prediction of water quality via artificial neural networks, *J. Cleaner Prod.*, **291** (2021), 125266. https://doi.org/10.1016/j.jclepro.2020.125266

22. T. A. Sinshaw, C. Q. Surbeck, H. Yasarer, Y. Najjar, Artificial neural network for prediction of total nitrogen and phosphorus in US lakes, *J. Environ. Eng.*, **145** (2019), 04019032. https://doi.org/10.1061/(ASCE)EE.1943-7870.0001528

23. M. Hameed, S. S. Sharqi, Z. M. Yaseen, H. A. Afan, A. Hussain, A. Elshafie, Application of artificial intelligence (AI) techniques in water quality index prediction: a case study in tropical region, Malaysia, *Neural Comput. Appl.*, **28** (2017), 893–905. https://doi.org/10.1007/s00521-016-2404-7

24. A. Kadam, V. Wagh, A. Muley, B. Umrikar, R. Sankhua, Prediction of water quality index using artificial neural network and multiple linear regression modelling approach in Shivganga River basin, India, *Model. Earth Syst. Environ.*, **5** (2019), 951–962. https://doi.org/10.1007/s40808-019-00581-3

25. Y. Zhang, X. Gao, K. Smith, G. Inial, S. Liu, L. B. Conil, et al., Integrating water quality and operation into prediction of water production in drinking water treatment plants by genetic algorithm enhanced artificial neural network, *Water Res.*, **164** (2019), 114888. https://doi.org/10.1016/j.watres.2019.114888

26. J. Wu, Z. Wang, A hybrid model for water quality prediction based on an artificial neural network, wavelet transform, and long short-term memory, *Water*, **14** (2022), 610. https://doi.org/10.3390/w14040610

27. Y. Wang, J. Zhou, K. Chen, Y. Wang, L. Liu, Water quality prediction method based on LSTM neural network, in *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, (2017), 1–5. https://doi.org/10.1109/ISKE.2017.8258814

28. Q. Ye, X. Yang, C. Chen, J. Wang, River water quality parameters prediction method based on LSTM-RNN model, in *2019 Chinese Control And Decision Conference (CCDC)*, (2019), 3024–3028. https://doi.org/10.1109/CCDC.2019.8832885

29. J. Bi, Y. Lin, Q. Dong, H. Yuan, M. Zhou, Large-scale water quality prediction with integrated deep neural network, *Inf. Sci.*, **571** (2021), 191–205. https://doi.org/10.1016/j.ins.2021.04.057

30. C. Hu, F. Zhao, Improved methods of BP neural network algorithm and its limitation, in *2010 International Forum on Information Technology and Applications*, (2010), 11–14. https://doi.org/10.1109/IFITA.2010.324

31. T. Venkateswarlu, J. Anmala, Application of random forest model in the prediction of river water quality, in *Proceedings of Seventh International Congress on Information and Communication Technology*, (2023), 525–535. https://doi.org/10.1016/j.asej.2021.11.004

32. M. Jeung, S. Baek, J. Beom, K. H. Cho, Y. Her, K. Yoon, Evaluation of random forest and regression tree methods for estimation of mass first flush ratio in urban catchments, *J. Hydrol.*, **575** (2019), 1099–1110. https://doi.org/10.1016/j.jhydrol.2019.05.079

33. H. Lu, X. Ma, Hybrid decision tree-based machine learning models for short-term water quality prediction, *Chemosphere*, **249** (2020), 126169. https://doi.org/10.1016/j.chemosphere.2020.126169

34. S. M. Saghebian, M. T. Sattari, R. Mirabbasi, M. Pal, Ground water quality classification by decision tree method in Ardebil region, Iran, *Arabian J. Geosci.*, **7** (2014), 4767–4777. https://doi.org/10.1007/s12517-013-1042-y

35. Z. Hippe, J. Zamorska, A new approach to application of pattern recognition methods in analytical chemistry. II. Prediction of missing values in water pollution grid using modified KNN-method, *Chem. Anal.*, **44** (1999), 597–602.

36. J. Park, W. H. Lee, K. T. Kim, C. Y. Park, S. Lee, T. Y. Heo, Interpretation of ensemble learning to predict water quality using explainable artificial intelligence, *Sci. Total Environ.*, **832** (2022), 155070. https://doi.org/10.1016/j.scitotenv.2022.155070

37. A. Gidon, T. A. Zolnik, P. Fidzinski, F. Bolduan, A. Papoutsi, P. Poirazi, et al., Dendritic action potentials and computation in human layer 2/3 cortical neurons, *Science*, **367** (2020), 83–87. https://doi.org/10.1126/science.aax6239

38. I. S. Jones, K. P. Kording, Might a single neuron solve interesting machine learning problems through successive computations on its dendritic tree?, *Neural Comput.*, **33** (2021), 1554–1571. https://doi.org/10.1162/neco_a_01390

39. A. Destexhe, E. Marder, Plasticity in single neuron and circuit computations, *Nature*, **431** (2004), 789–795. https://doi.org/10.1038/nature03011

40. C. Koch, Computation and the single neuron, *Nature*, **385** (1997), 207–210. https://doi.org/10.1038/385207a0

41. B. E. Stein, T. R. Stanford, B. A. Rowland, Development of multisensory integration from the perspective of the individual neuron, *Nat. Rev. Neurosci.*, **15** (2014), 520–535. https://doi.org/10.1038/nrn3742

42. Y. Todo, H. Tamura, K. Yamashita, Z. Tang, Unsupervised learnable neuron model with nonlinear interaction on dendrites, *Neural Netw.*, **60** (2014), 96–103. https://doi.org/10.1016/j.neunet.2014.07.011

43. F. Teng, Y. Todo, Dendritic neuron model and its capability of approximation, in *2019 6th International Conference on Systems and Informatics (ICSAI)*, (2019), 542–546. https://doi.org/10.1109/ICSAI48974.2019.9010147

44. J. He, J. Wu, G. Yuan, Y. Todo, Dendritic branches of dnm help to improve approximation accuracy, in *2019 6th International Conference on Systems and Informatics (ICSAI)*, (2019), 533–541. https://doi.org/10.1109/ICSAI48974.2019.9010196

45. Z. Sha, L. Hu, Y. Todo, J. Ji, S. Gao, Z. Tang, A breast cancer classifier using a neuron model with dendritic nonlinearity, *IEICE Trans. Commun.*, **98** (2015), 1365–1376. https://doi.org/10.1587/transinf.2014EDP7418

46. T. Jiang, S. Gao, D. Wang, J. Ji, Y. Todo, Z. Tang, A neuron model with synaptic nonlinearities in a dendritic tree for liver disorder, *IEEJ Trans. Electr. Electron. Eng.*, **12** (2017), 105–115. https://doi.org/10.1002/tee.22350

47. Y. Tang, J. Ji, S. Gao, H. Dai, Y. Yu, Y. Todo, A pruning neural network model in credit classification analysis, *Comput. Intell. Neurosci.*, **15** (2014), 520–535. https://doi.org/10.1155/2018/9390410

48. Z. Song, C. Tang, J. Ji, Y. Todo, Z. Tang, A simple dendritic neural network model-based approach for daily pm2.5 concentration prediction, *Electronics*, **10** (2021), 373. https://doi.org/10.3390/electronics10040373

49. Z. Song, Y. Tang, J. Ji, Y. Todo, Evaluating a dendritic neuron model for wind speed forecasting, *Knowl. Based Syst.*, **201** (2020), 106052. https://doi.org/10.1016/j.knosys.2020.106052

50. T. Zhou, S. Gao, J. Wang, C. Chu, Y. Todo, Z. Tang, Financial time series prediction using a dendritic neuron model, *Knowl. Based Syst.*, **105** (2016), 214–224. https://doi.org/10.1016/j.knosys.2016.05.031

51. W. Chen, J. Sun, S. Gao, J. J. Cheng, J. Wang, Y. Todo, Using a single dendritic neuron to forecast tourist arrivals to japan, *IEICE Trans. Inf. Syst.*, **100** (2017), 190–202. https://doi.org/10.1587/transinf.2016EDP7152

52. S. J. Reddi, S. Kale, S. Kumar, On the convergence of adam and beyond, preprint, arXiv:1904.09237. https://doi.org/10.48550/arXiv.1904.09237

53. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, preprint, arXiv:1412.6980. https://doi.org/10.48550/arXiv.1412.6980

54. J. Ji, M. Dong, Q. Lin, K. C. Tan, Noninvasive cuffless blood pressure estimation with dendritic neural regression, *IEEE Trans. Cybern.*, **2022** (2022). https://doi.org/10.1109/TCYB.2022.3141380

55. J. F. Khaw, B. Lim, L. E. Lim, Optimal design of neural networks using the taguchi method, *Neurocomputing*, **7** (1995), 225–245. https://doi.org/10.1016/0925-2312(94)00013-I

56. D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning internal representations by error propagation, in *California Univ San Diego La Jolla Inst for Cognitive Science*, 1985.

57. J. H. Friedman,, Greedy function approximation: a gradient boosting machine, *Ann. Stat.*, **2001** (2001), 1189–1232. https://doi.org/10.1214/AOS/1013203451

58. T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in *Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining*, (2016), 785–794. https://doi.org/10.1145/2939672.2939785

59. G. B. Huang, Q. Y. Zhu, C. K. Siew, Extreme learning machine: theory and applications, *Neurocomputing*, **70** (2006), 489–501. https://doi.org/10.1016/j.neucom.2005.12.126

60. D. W. Zimmerman, B. D. Zumbo, Relative power of the Wilcoxon test, the Friedman test, and repeated-measures ANOVA on ranks, *J. Exp. Educ.*, **62** (1993), 75–86. https://doi.org/10.1080/00220973.1993.9943832

AIMS Press

© 2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)