Journal of Autonomous Intelligence (2021) Volume 4 Issue 1.
doi: 10.32629/jai.v4i1.483

Original Research Article

Principle of Machine Learning and Its Potential Application in Climate Prediction

***Shengping He*[1,2*], *Huijun Wang*[1,3,4], *Hua Li*[1,3], *Jiazhen Zhao*[1]**

[1] Key laboratory of meteorological disasters of the Ministry of Education/Collaborative Innovation Center for meteorological disaster prediction, early warning and assessment, Nanjing University of Information Engineering, Nanjing 210044, China

[2] Institute of Geophysics, University of Bergen, Bergen 5020, Norway

[3] Zhu Kezhen-Nansen International Research Center, Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing 10029, China

[4] Climate change research center, Chinese Academy of Sciences, Beijing 10029, China

ABSTRACT

After two "cold winters of artificial intelligence", machine learning has once again entered the public's vision in recent ten years, and has a momentum of rapid development. It has achieved great success in practical applications such as image recognition and speech recognition system. It is one of the main tasks and objectives of machine learning to summarize key information and main features from known data sets, so as to accurately identify and predict new data. From this perspective, the idea of integrating machine learning into climate prediction is feasible. Firstly, taking the adjustment of linear fitting parameters (i.e. slope and intercept) as an example, this paper introduces the process of machine learning optimizing parameters through gradient descent algorithm and finally obtaining linear fitting function. Secondly, this paper introduces the construction idea of neural network and how to apply neural network to fit nonlinear function. Finally, the framework principle of convolutional neural network for deep learning is described, and the convolutional neural network is applied to the return test of monthly temperature in winter in East Asia, and compared with the return results of climate dynamic model. This paper will help to understand the basic principle of machine learning and provide some reference ideas for the application of machine learning to climate prediction.

Keywords: *Machine Learning; Neural Network; Convolutional Neural Network; Climate Prediction; Winter Temperature in East Asia*

ARTICLE INFO
Received: Feb 17, 2021
Accepted: Apr 7, 2021
Available online: Apr 13, 2021

***CORRESPONDING AUTHOR**
Shengping He
hshp@mail.iap.ac.cn;

CITATION
He S, Wang H, Li H, *et al.* Principle of machine learning and its potential application in climate prediction. Journal of Autonomous Intelligence 2021; 4(1): 13-28. doi: 10.32629/jai.v4i1.483

1. Introduction

In 1956, McCarthy *et al.* (1956) put forward the concept of artificial intelligence. Three years later, Samuel (1959) proposed a way to realize artificial intelligence—machine learning. Subsequently, AI experienced two take-off times: the 1960s to 1970s and the 1980s. Nevertheless, AI has not made satisfactory achievements. It has experienced two "AI winter" in the late 1970s and early 1990s. Despite the ups and downs in the development of artificial intelligence, as a branch of artificial intelligence and a way to realize artificial intelligence, the development of machine learning (especially the updating of algorithms) has never stopped, and has gradually developed into an interdisciplinary subject involving probability theory, statistics, approximation theory and so on. In recent years, with the significant improvement of the performance of computer hardware facilities, the massive growth of research data, the significant reduction of storage cost and the obvious improvement of algorithms, machine learning, especially deep learning, has once again back to public's focus and achieved a series of successes. Some machine models trained with a large amount of data can accurately predict new data, such as automatic driving, image recognition and speech recognition, which are successful applications of machine learning (Huntingford *et al.*, 2019).

Human beings have always been committed to understanding and predicting the changes of the world, and the most successful example is numerical weather forecasting. Now, its prediction skills for 3–5 days of 500 hPa geopotential height field in the northern hemisphere have reached more than 90% (Bauer *et al.*, 2015). However, climate prediction on seasonal scale and climate prediction on longer time scale are still great challenges (Hantson *et al.*, 2016). Driven by the in-depth understanding of the climate system change mechanism, the observation data, reanalysis data and numerical simulation data of the earth system have increased rapidly in the past 40 years. In particular, the fifth phase (CMIP5) and the sixth phase of the international coupled model comparison program (CMIP6) provide tens of billions of bytes of data resources for climate change, climate prediction and climate prediction research (Stockhouse & Lautenschlager, 2017). How to fully extract useful information and acquire new knowledge from "big data" poses a new challenge to traditional analysis methods. Machine learning and artificial intelligence bring new opportunities. Machine learning can discover and extract new inmatically identify extreme weather events without any threshold (Liu *et al.*, 2016). In addition, ma-

With the aggravation of climate change and its negative impacts (Pörtner *et al.*, 2019), it is increasingly important and urgent to improve the ability of climate prediction. However, this is still a severe challenge to the current dynamic climate prediction model. Machine learning, supported by high-performance computers, "big data" and advanced algorithms, has improved new ideas and opportunities for improving the skills of climate prediction. This paper will briefly introduce the basic principles of machine learning centred on gradient deterrelated signals from the "big data" of the earth system. For example, the SST information of a key area can improve the climate prediction skills of a certain area on land in the coming months. On this basis, artificial intelligence can provide the society with automatic early warning of extreme weather and climate events (Huntingford *et al.*, 2019).

Nowadays, machine learning is gradually combined with climate prediction and weather prediction, and a large number of innovative research results have emerged in related fields. Ham *et al.* (2019) constructed a machine prediction model for ENSO index by using deep neural network. The results show that the prediction skill of deep learning prediction model for ENSO 7–21 months in advance is higher than that of most current dynamic climate prediction models. The shallow neural network machine model can also better distinguish the central and eastern ENSO events (Toms *et al.*, 2020). In addition, machine learning can also be applied to weather forecasting business (Men *et al.*, 2019). Weyn *et al.* (2019) constructed a machine prediction model of 500 hPa potential height grid field by using convolution neural network (deep learning). Its prediction skill of 3 days in advance is obviously better than that of the dynamic barotropic vorticity model, although its performance is still inferior to the current operational numerical weather prediction system. The convolutional neural network machine model can also predict the frontal system of weather scale (Lagerquist *et al.*, 2019) The deep learning model can also autochine learning can be used to reduce the uncertainty of future climate prediction (Kuang *et al.*, 2020). scent, the construction of neural network and the framework of deep learning. Finally, an example of applying deep learning to winter temperature prediction in East Asia is introduced.

2. Introduction to Artificial Intelligence, Machine Learning and Deep Learning

In the 1950s, John McCarthy and others launched the Dartmouth summer artificial intelligence research program (McCarthy *et al.*, 1956) to explore topics such as automatic computers and neural networks. The concept of "artificial intelligence" was born. "Artificial intelligence" aims to endow computers with the ability of "thinking", which refers to the theory and development to realize that computer systems can perform tasks that usually require human intelligence. Obviously, "artificial intelligence" is a concept or general term covering a wide range. The early "artificial intelligence" was mainly realized through hard coding, that is, based on the existing knowledge system of human beings, the code program was designed manually to complete the tasks that challenge human beings. For example, the computer chess player "Dark Blue" designed by IBM is to fully formalize the rules of chess and then describe them to the computer through hard coding. "Dark blue" defeated world chess champion Gary Kasparov on May 11, 1997. However, with the improvement of practical application requirements and the limitations of human cognitive system, the bottleneck of hard coded "artificial intelligence" began to show: It can not solve more complex problems. In order to make up for the disadvantage that hard coding has high requirements on human cognitive system, scientists put forward a new idea of building "artificial intelligence", that is, to make it the characteristic of computer to automatically generalizing and summarizing information from big data, i.e. machine learning. Although machine learning still needs to be realized through coding, it has a feature that is obviously different from the traditional hard coding method. In the early stage of task execution, the computer does not give specific rules to solve the problem (such as the known chess rules of "dark blue"), but uses a large amount of data and constantly "trains" the computer through some algorithm. At the same time, a loss function is used to measure the learning effect of the computer, and the direction of "training" is adjusted through the optimization algorithm. Through repeated iterative calculation, the computer finally has the optimal scheme or rules to solve the problem (i.e. Parameters, see Section II). In this way, the "trained" machine can be put into practical application, such as face recognition and speech recognition system, which are the results of machine learning. It can be seen that algorithm is the core of machine learning, and neural network is one of the classical algorithms. Deep learning is to realize machine learning by using neural networks with more levels (i.e. the meaning of depth).

3. Principles of Machine Learning

Machine learning can be divided into supervised learning, unsupervised learning and reinforcement learning (Dougherty *et al.*, 1995). This paper mainly focuses on supervised learning. The characteristic of supervised learning is that each "training data" has a clear output expectation (i.e. "label data"). In order to explain the "learning" process of the machine simply and clearly, we take the simplest linear regression as an example to show how to continuously "train" the machine and finally obtain the parameters of the linear regression equation (i.e. slope θ_1 and intercept θ_2). Build a linear function: $y = 2.5x + 3.5 + \delta$ ($x = 1, 2, 3, \ldots, 20$), among which δ represents the noise data conforming to the random normal distribution. The mapping relationship between x and y is shown in the scatter diagram of **Figure 1**(A). From the perspective of machine learning, x is called "training data" and y "label data" (**Table 1**).

Input the "training data" x into the computer and randomly give any two initial parameters of the computer, namely slope θ_1 and intercept θ_2. Since the goal of the computer is to "learn" a linear relationship, the "predicted value" $\hat{y}$ of the corresponding output should meet $y = \theta_1^0 \times x + \theta_2^0$. In order to evaluate the "learning" effect of computer, that is, to measure the difference between $\hat{y}$ and y, a cost function, also known as loss function, needs to be introduced. Root mean square error is selected here:

$$\sum_{i=1}^{m}(\hat{y}_i - y_i)^2 \tag{1}$$

Table 1. “Training data” denotes x, the “label data” denotes y, and “random noise data” denotes δ

X	Δ	Y
1	0. 57	6. 57
2	-0. 39	8. 11
3	0. 19	11. 19
4	-0. 78	12. 72
5	2. 5	18. 5
6	1. 48	19. 98
7	0. 27	21. 27
8	-0. 82	22. 68
9	-1. 58	24. 42
10	-0. 86	27. 64
11	-0. 31	30. 69
12	-0. 91	32. 59
13	-1. 64	34. 36
14	0. 47	38. 97
15	0. 63	41. 63
16	0. 17	43. 67
17	-0. 78	45. 22
18	-0. 57	47. 93
19	-0. 25	50. 75

Of which M represents the number of data sets. Since “training data” x and “label data” y are determined data sets, and only parameters θ_1 and intercept θ_2 are uncertain, so the loss function written as $f(\theta_1, \theta_2)$ is actually about θ_1 and θ_2. For the convenience of description, the parameters are expressed in the form of vectors $\Theta(\theta_1, \theta_2)$. In other words, the ultimate goal of “training” the machine is to adjust the parameters Θ and make the value of $f(\Theta)$ reaches the minimum.

According to the principle of derivative function, that is, $f(\Theta)$’ derivative function $\nabla f(\Theta)_{|(\theta_1^0,\theta_2^0)}$ at certain point $\Theta^0(\theta_1^0,\theta_2^0)$ represents the fastest direction that $f(\Theta)$ increases. In order to effectively “learn” towards the minimum value of $f(\Theta)$, the machine can adjust parameters along the opposite direction of the derivative function to obtain new parameters $\Theta^1(\theta_1^1,\theta_2^1)$, namely:

$$\Theta^1 = \Theta^0 - \alpha \times \nabla f(\Theta)_{|\Theta^0} \tag{2}$$

If $\Theta^1$is not the parameter when $f(\Theta)$ reaches its minimum value, then the machine continue to adjust the parameter along the opposite direction of the derivative function to obtain new parameters θ_2, namely:

$$\Theta^2 = \Theta^1 - \alpha \times \nabla f(\Theta)_{|\Theta^1} \tag{3}$$

Among them, α∈(0,1) is called “learning efficiency”. By iterating the above calculation process repeatedly, the computer will continuously reduce the loss function $f(\Theta)$, and lock the parameter until it is less than a critical value Θ. At this time, the parameter $\Theta(\theta_1^n,\theta_2^n)$ (n represents the final number of iterations) “learned” by the computer will make the “predicted value” $\hat{y}$ approach the “label data” y optimally. The above process of adjusting parameters along the opposite direction of derivative function is called “gradient descent” method (Ruder, 2016); The module similar to adjusting parameters is called “optimizer”.

Return to the problem of machine learning to solve the above linear regression. In order to make the description easier, the above “predicted value” $\hat{y}$ “training data” x, parameters (θ_1, θ_2) and “label data” y are expressed in the form of matrix respectively:

$$\hat{Y} = \begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_{20} \end{pmatrix},$$

$$X= \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_{20} & 1 \end{pmatrix},$$

$$\Theta= \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix},$$

$$Y= \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{20} \end{pmatrix}.$$

Therefore, the loss function can be expressed as:

$$f(\Theta) = \frac{1}{2m}(X \cdot \Theta - Y)^{\mathrm{T}}(X \cdot \Theta - Y) \tag{4}$$

Of which: m is the number of each data set, i.e. 20, and the constant 1/2 is to avoid redundant constants in the subsequent derivation function. The derivative function of the loss function is:

$$\nabla f(\Theta) = \frac{1}{m} X^{\mathrm{T}}(X \cdot \Theta - Y) \tag{5}$$

Firstly, the initial parameter of the computer given randomly $\Theta^0(20,-20)$ is substituted into formula (4) and formula (5) together with “training data” x and “label data” y (**Table 1**), and the loss function and its derivative can be obtained:

$f(20.-20) = 17\,934.65,$

$$\nabla f(\Theta)_{|\Theta^0(20,-20)} = (\frac{\partial f}{\partial \theta_1}, \frac{\partial f}{\partial \theta_2})_{|\Theta^0(20,-20)}$$
$$= (2264.65,160.25)$$

At the same time, the “learning” efficiency of the machine α is set to 0.01. According to formula (2), the machine updates the parameters to:

$$(\theta 1, \theta 2) = (20, -20) - 0.01 \times (2264.65,160.25)$$
$$= (-2.646, -21.603)$$

The machine continuously updates the parameter (θ_1, θ_2) through the above “learning” process (**Figure 1**B), and the value of the loss function continues to decrease (as shown by the red line in **Figure 1**(C). After about 3000 iterative calculations, the parameter (θ_1, θ_2) basically tends to be stable (**Figure 1**B), indicating that the loss function has approached its minimum value, and this point (θ_1^n, θ_2^n) is also the position where the derivative of the loss function is the smallest (i.e. The slope is the smallest). Set the critical value $\nabla f(\theta_1^n, \theta_2^n)$ of the derivative function to 10 × 10^{-5}, that is, when the value of the derivative function is less than the critical value, the machine will stop “learning” and lock the parameter (θ_1^n, θ_2^n) at this time, i.e. (2.50,3.53). Therefore, the linear fitting curve finally “learned” by the machine is: $\hat{y} = 2.50x + 3.53$ (**Figure 1**A: red line), which basically conforms to the linear relationship between “training data” x and “label data” *y* (**Figure 1**A: scattered points).

Of course, the above is only the process of “training” the machine. The ultimate goal of machine learning is to use the “trained” machine (i.e. complete the optimization of parameters) to predict the new data (commonly referred to as “test data”) that the machine has never engaged before. It will be described in detail in section III and section IV.

4. The Idea of Applying Neural Network to Climate Prediction

Linear regression is a common method in climate prediction and climate change research. Of course, the linear regression function can be calculated quickly without machine learning. However, in the process of practical research and application, we often face a large number of observation and numerical simulation data. Due to the complexity of the climate system, there may be some nonlinear relationship between the data, and the above machine learning model based on linear relationship will lose its function. At this time, deep learning can give play to its great advantages. Deep learning is based on neural network, which usually includes one input layer, several hidden layers and one output layer. Each neural layer contains several neurons (in fact, it represents the node containing a specific data). The input layer is responsible for receiving “training data” or “test data”, and the output layer is responsible for exporting “prediction data”. The main function of the hidden layer is to connect the input layer and the output layer through a large number of parameters. The loss function can be constructed by using the “prediction data” of the output layer and the known “label data”, and then the loss function can be reduced and the parameters can be adjusted through the optimizer. When the loss function reaches the minimum value, the parameters will be locked, that is, complete the “training” of the machine (see Section 2). The key question here is: how do neurons connect between input layer, hidden layer and output layer? The answer is: matrix multiplication.

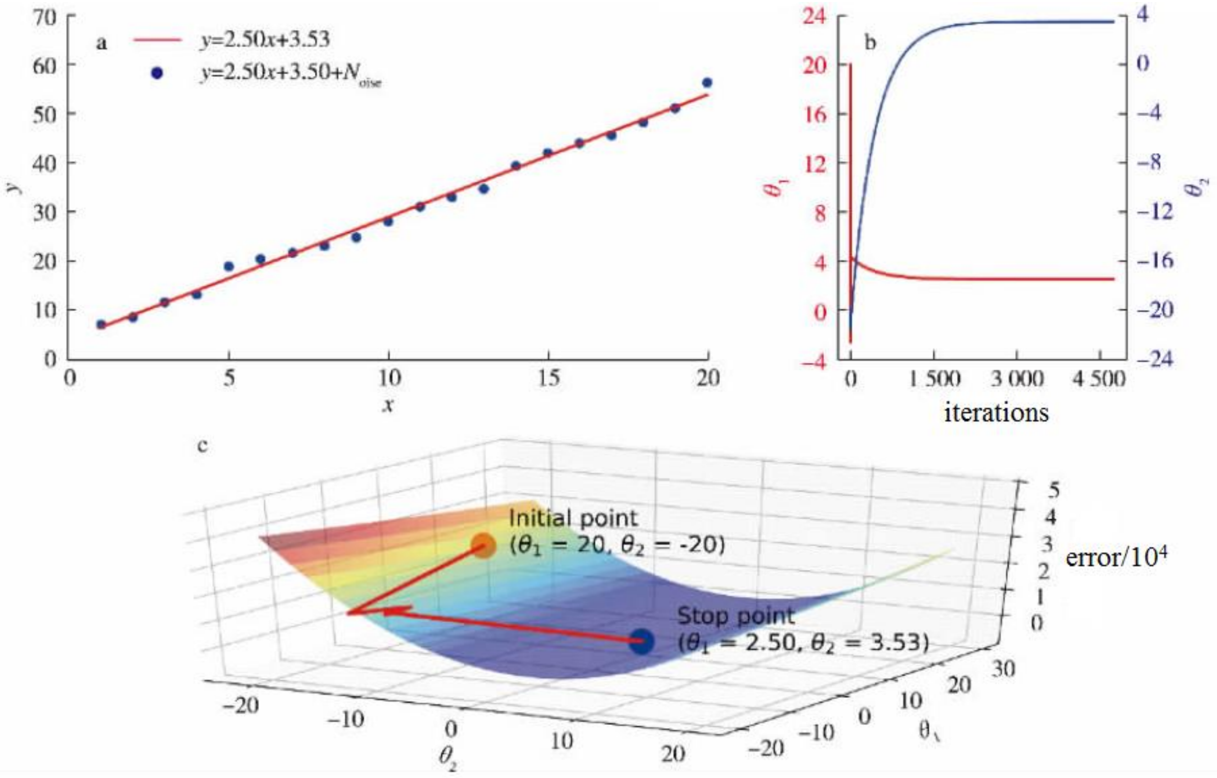

Figure 1. (A) Scatters indicate the mapping relationship between the 'train data' x and 'labeled data' y, the red line indices the linear fitting by machine linear; (B) the updating of weights along the iteration; (C) the gradient descent of machine learning.

For simplicity, first build a shallow neural network: an input layer containing a neuron node; two hidden layers, including 4 and 5 neuron nodes respectively; an output layer contains a neuron node (**Figure 2**A). We will gradually analyze the connection mode between neurons from the perspective of matrix multiplication.

1). Input layer to first hidden layer

Because the input layer has only one neuron, i.e. only one data, it can be regarded as a matrix $\boldsymbol{X} = [x_1^1]$ with one row and one column. The subscript of x_i^j here represents the j-th characteristic data of the i-th sample. $x_1^1$and x_1^2 for example, can represent the temperature and precipitation of an observation station at the first observation time, and so on (the same below). The hidden layer contains 4 neurons, i.e. 4 data, which can be expressed as a matrix with one row and four columns $\boldsymbol{Y} = [y_1^1, y_1^2, y_1^3, y_1^4]$. In order to realize the mapping between matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$, a parameter matrix with one row and four columns $\boldsymbol{\omega} = [\omega_{11}, \omega_{12}, \omega_{13}, \omega_{14}]$ can be constructed. Therefore, the $\boldsymbol{X} \cdot \boldsymbol{\omega} = \boldsymbol{Y}$ connection between the input layer and the neurons of the first hidden layer is realized (**Figure 2**B).

2). First hidden layer to second hidden layer

There are five neurons in the second hidden layer, which can be expressed as a matrix with one row and five columns $\boldsymbol{Z} = [z_1^1, z_1^2, z_1^3, z_1^4, z_1^5]$. According to the above ideas, a new parameter matrix with five rows and five columns **θ** needs to be constructed. So $\boldsymbol{Y} \cdot \boldsymbol{\theta} = \boldsymbol{Z}$ realizes the connection between the neurons of the first hidden layer and the second hidden layer (**Figure 2**B).

3). Second hidden layer to output layer

There is only one predicted value P_1 in output layer, so it is only necessary to build a new parameter matrix with five rows and one column **μ** and the connection between hidden layer 2 and output layer neurons can be realized (**Figure 2**B).

After the above neural network is constructed, a batch of "training data" sets (assuming n samples) can be input into the neural network to obtain n numbers of "prediction data" sets. Combined with the corresponding n number of "tag data", we can get the information about $\boldsymbol{\omega}$,$\boldsymbol{\theta}$ and $\boldsymbol{\mu}$ of the loss function $f(\boldsymbol{\omega}, \boldsymbol{\theta}, \boldsymbol{\mu})$. Then, the optimizer continuously reduces the loss function and updates the parameters $\boldsymbol{\omega}$,$\boldsymbol{\theta}$ and $\boldsymbol{\mu}$ at the same time (**Figure 2**B); see Section IV). It is worth noting that in order to explore the nonlinear relationship between the hidden layer and its front and rear layers, the neural network will introduce a nonlinear "excitation function" into the hidden layer. This is also the reason why the neural network algorithm is superior to the linear model (Specht, 1991).

In order to intuitively show the learning effect of neural network, a nonlinear function $y' = \sin(3.5 \times \cos(2.5\theta))$ is constructed. θ is 300 equally spaced data between –1 and 1, i.e. the "training data" set; the function curve between y' and θ is shown in the blue line of **Figure 3**(A). A certain random noise is superimposed on the nonlinear function to obtain 300 "tag data" *y*. The mapping result between θ and y is shown in **Figure 3**(A). Building a neural network includes one input layer, two hidden layers (including 16 neurons respectively) and one output layer. Input the "training data" θ into the neural network and use the hyperbolic tangent function (TANH) excitation function in the hidden layer (**Figure 3**B). When the machine goes through 6000 iterations, the output value learned by the machine is shown in the yellow line in **Figure 3**(A). At this time, the value of the loss function is 0.01. It can be seen that neural network machine learning has a good performance in solving nonlinear problems. When the rectified linear unit (RELU) "excitation function" (**Figure 3**C) is adopted, the output value of the machine after 6000 iterations is as the red line in **Figure 3**(A), and the value of the loss function is 0.052.

The neural network constructed above aims at the input layer with only one eigenvalue $[x_1^1]$ (**Figure 2**A). If the input layer needs to process multiple eigenvalues, such as the East Asian winter temperature index predicted with the autumn Arctic sea ice index and the autumn Eurasian snow index (i.e. the two eigenvalues of the input layer correspond to one output value), how to build a neural network? At this time, it is only necessary to increase the number of neurons in the input layer to two (i.e. the input layer matrix is two columns $[x_1^1, x_1^2]$) and the parameter matrix multiplied by it to two rows (**Figure 2**B), so as to complete the construction of neural network with two eigenvalues in the input layer, and the process goes on. Due to the complexity and diversity of climate systems and the nonlinear interaction between climate systems (Hasselmann, 1999), machine learning will be used to build climate prediction models in the future to further improve the skills of climate prediction.

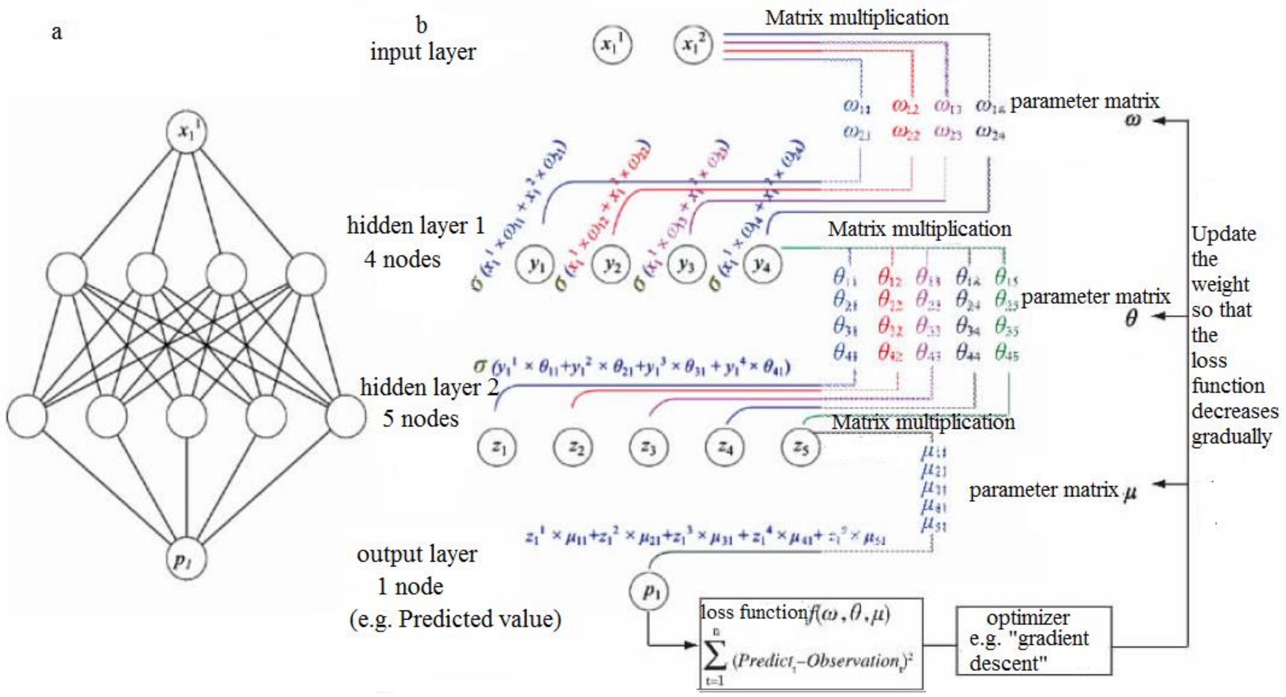
a
x_1^1
p_1
b
input layer
x_1^1
x_1^2
Matrix multiplication
ω_{11} ω_{12} ω_{13} ω_{14}
ω_{21} ω_{22} ω_{23} ω_{24}
parameter matrix ω
hidden layer 1
4 nodes
$\sigma(x_1^1 \times \omega_{11} + x_1^2 \times \omega_{21})$
$\sigma(x_1^1 \times \omega_{12} + x_1^2 \times \omega_{22})$
$\sigma(x_1^1 \times \omega_{13} + x_1^2 \times \omega_{23})$
$\sigma(x_1^1 \times \omega_{14} + x_1^2 \times \omega_{24})$
y_1 y_2 y_3 y_4
Matrix multiplication
θ_{11} θ_{12} θ_{13} θ_{14} θ_{15}
θ_{21} θ_{22} θ_{23} θ_{24} θ_{25}
θ_{31} θ_{32} θ_{33} θ_{34} θ_{35}
θ_{41} θ_{42} θ_{43} θ_{44} θ_{45}
parameter matrix θ
$\sigma(y_1^1 \times \theta_{11} + y_1^2 \times \theta_{21} + y_1^3 \times \theta_{31} + y_1^4 \times \theta_{41})$
hidden layer 2
5 nodes
z_1 z_2 z_3 z_4 z_5
Matrix multiplication
μ_{11} μ_{21} μ_{31} μ_{41} μ_{51}
parameter matrix μ
$z_1^1 \times \mu_{11} + z_1^2 \times \mu_{21} + z_1^3 \times \mu_{31} + z_1^4 \times \mu_{41} + z_1^5 \times \mu_{51}$
output layer
1 node
(e.g. Predicted value)
p_1
loss function $f(\omega, \theta, \mu)$
$\sum_{t=1}^{n} (Predict_t - Observation_t)^2$
optimizer
e.g. "gradient descent"
Update the weight so that the loss function decreases gradually

Figure 2. (A) A shallow neural network; (B) illustrating the architecture of neural network.

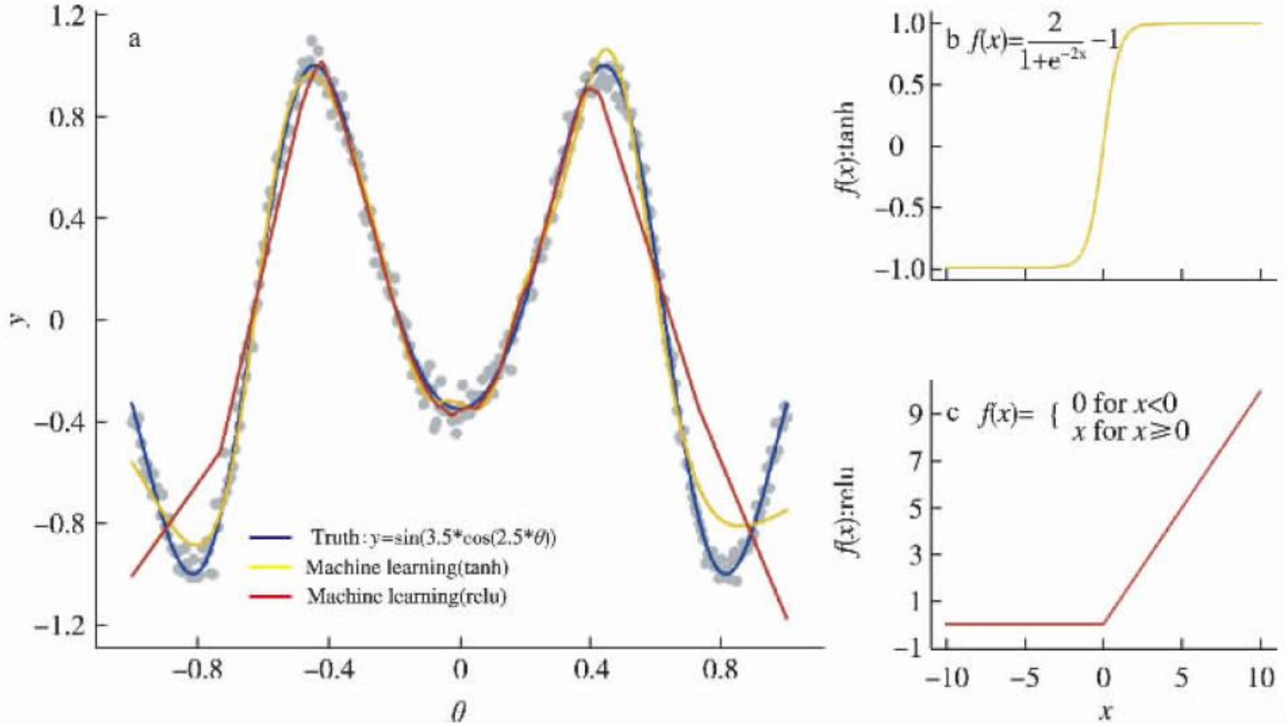

Figure 3. (A) Non-linear fitting by neural network with different activation functions; the blue curve indicates the 'true' curve of non-linear function; scatters indicate the non-linear function $f(\theta) = \sin(3.5 \cos(2.5 \theta))$ overlapped by random noise; the yellow and red curves are results of machine learning with activation function of tanh and relu, respectively; (B) and (C) illustrate the activation function of tanh and relu, respectively.

5. Deep Learning: Convolutional Neural Network and Its Application to Winter Temperature in East Asia

Convolutional neural network (CNN) adds one or more convolutional layers and pooling layers on the basis of ordinary neural network, including maximum pooling and average pooling (Goodfellow *et al.*, 2016). The process of convolution is as follows. Firstly, a convolutional kernel, i.e. a weight matrix, is randomly given, whose dimension is the same as that of the convoluted data, but the horizontal resolution is smaller. The convolution kernel extracts data subsets from the convoluted data in a fixed step according to its own resolution, multiplies them correspondingly, and then sums them, until the retrieval of all data is completed. In order to consider the nonlinearity of the data, the convoluted data will go through an "excitation function", and the final output result will enter the pooling layer (see "step 1" in **Figure 4**).

Pooling (taking the maximum pooling as an example) is based on the specified horizontal resolution (such as 2 × 2) retrieve the output data of the convolution layer according to the specified step size, and output the maximum value within the range of the grid point every time until the retrieval of all data is completed (see "step 2" in **Figure 4**). Two points need to be pointed out.

1). Only one convolution kernel is used in **Figure 4**, so the data after convolution is still two-dimensional. In fact, multiple different convolution check data can be used for convolution, and each convolution kernel convolutes the data according to the above process. Therefore, when all convolution cores complete the convolution process, the horizontal resolution of the output data will be significantly reduced and one dimension will be added (equal to the number of convolution cores).

2). In **Figure 4**, there is only one convolution layer and one pool layer. In practical application, the above convolution and pooling process can be repeated for many times, i.e. the pooled data will undergo convolution and pooling. Convolution kernel, convolution layer and the number of pool layers need to be adjusted according to specific problems and test results. Convert the pooled data into one-dimensional data and input it into the input layer of ordinary neural network, then the construction of convolutional neural network can be completed (see “step 3” in **Figure 4**). Convolutional neural network has achieved great success in computer vision (such as image classification and recognition) and natural language processing (Goodfellow *et al.*, 2016; Huntingford *et al.*, 2019). Two dimensional or three-dimensional data are often used in climate prediction research and application. Therefore, in theory, convolutional neural network can be applied to the field of climate prediction. In addition, the abundance of climate system observation data and numerical simulation data provides sufficient training data for machine learning. In order to try to apply convolutional neural network to climate prediction, this paper uses convolutional neural network method and uses the historical simulation data of the fifth stage coupled model comparison program (CMIP5) to construct a machine prediction model for the monthly temperature index in winter in East Asia. Then input the trained machine prediction model with the historical observation data to carry out the return test on the historical observation time series of monthly temperature in winter in East Asia. The research data, modeling methods and return results are as follows.

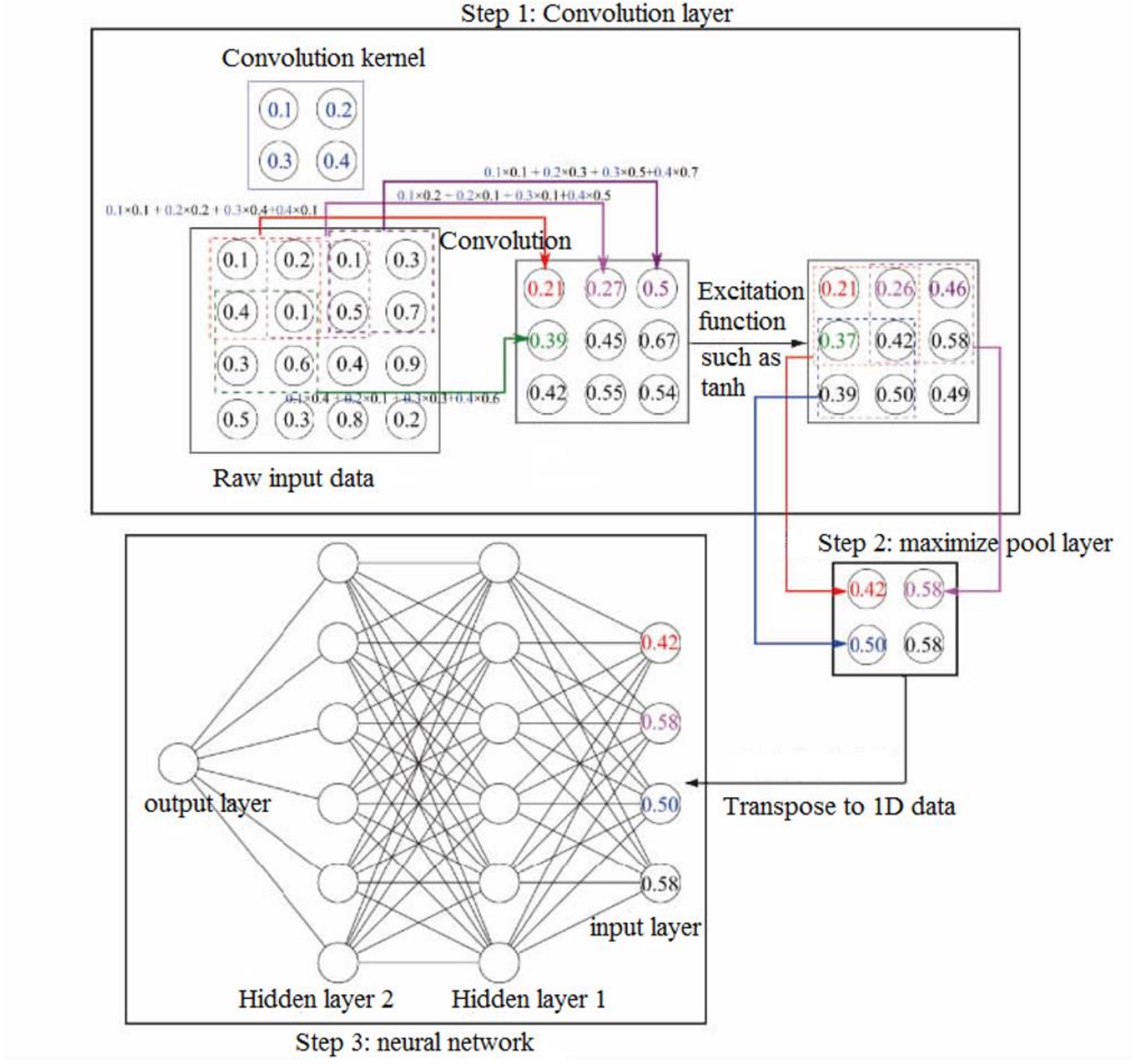

Figure 4. The architecture of revolutionary neural network.

5.1 Research data

The historical simulation data of 21 climate models are taken from CMIP5 from 1861 to 2005, and variables include surface temperature T2m, sea surface temperature SST and 0–300m average ocean temperature T300 (https://esgf-node.llnl.gov/projects/cmip5/). The historical return test results of dynamic climate model CanCM 4i are taken from https://iridl.ldeo.columbia.edu/SOURCES/.Models/NMME/. Observation data include the following. 1). SST data from the global ocean data assimilation system (GODAS) of the National Center for Environmental Prediction (NCEP) (https://psl.noaa.gov/data/gridded/data.godas.html; Behringer & Xue, 2004) from 1982 to 2018. 2). The surface temperature T2m in the reanalysis data (ERA5) of the European Center for medium and long term forecasting is from 1982 to 2018 (C3S, 2017). In order to save the training time of the machine, the SST and T300 of CMIP5 and GODAS are interpolated to 5° × 5° horizontal resolution, ranging from 60° s to 60° n, 0°–360°, i.e. the grid resolution is 24 (Zonal) × 72 (meridional direction).

5.2 Modeling method

1). Build machine “training data” Train_data, “label data” Labeled_data and “test data” Test_data (**Table 2**).

Firstly, T2m, SST and T300 in all data are transformed into anomaly fields (minus the climate state in the corresponding data period), and the anomaly fields in CMIP5, GODAS and EAR5 are recorded as CMIP_SSTA, CMIP_T300A, CMIP_T2mA, GODAS _SSTA, GODAS_T300A, ERA5_T2Ma respectively. It is planned to forecast East Asia winter month by month (i.e. December, January and February) with one month in advance. The prediction factors are SST and T300 anomaly field for three consecutive months in the early stage.

Table 2. Machine learning model and its training data (Train_data), labeled data (Labeled_data), testing data (Test_data) and prediction (Prediction)

Machine model	Training data	Lable data	Test data	Estimate
ML1	CMIP_SSTA and CMIP_T300A of September, October and November 1861–2004, recorded as Train_data1	The regional average value of CMIP_T2mA of a region in East Asia (100° ~ 140°E, 10° ~ 30°N) in December 1861–2004 is recorded as CMIP_T2m_Dec	GODAS_SSTA and GODAS_T300A of September, October and November 1982–2017, recorded as Test_data1	The regional average value T2mA in a region of East Asia (100° ~ 140°E, 10° ~ 30°N) in December 1982–2017 is recorded as Pre_T2m_Dec
ML2	CMIP_SSTA and CMIP_T300A of October, November and December 1861–2004, recorded as Train_data2	The regional average value of CMIP_T2mA of a region in East Asia (100° ~ 140°E, 10° ~ 30°N) in January 1862–2005 is recorded as CMIP_T2m_Jan	GODAS_SSTA and GODAS_T300A of October, November and December 1982–2017, recorded as Test _data2	The regional average value T2mA in a region of East Asia (100° ~ 140°E, 10° ~ 30°N) in January 1983–2018 is recorded as Pre_T2m_Jan
ML3	CMIP_SSTA and CMIP_T300A of November and December 1861–2004 and January 1862–2005, recorded as Train_data3	The regional average value of CMIP_T2mA of a region in East Asia (100° ~ 140°E, 10° ~ 30°N) in February 1862–2005 is recorded as CMIP_T2m_Feb	GODAS_SSTA and GODAS_T300A of November and December 1982–2017 and January 1983–2018, recorded as Test_data3	The regional average value T2mA in a region of East Asia (100° ~ 140°E, 10° ~ 30°N) in February 1983–2018 is recorded as Pre_T2m_Feb

In order to test the prediction effect of the machine prediction model, the regional average value of ERA5_T2mA in East Asia (100° ~ 140°E, 10° ~ 30°N) from 1982 to December 2017, 1983 to January 2018 and 1983 to February 2018 were further calculated, and they are recorded as ERA5_T2m_Dec, ERA5_t2m_Jan and ERA5_T2m_Feb respectively. It is worth noting that in order to obtain enough training data samples as much as possible, the full-time data of CMIP5 historical simulation test is used, resulting in a certain overlap between the training data and the test data. However, considering that the correlation coefficient between the climate interannual variability simulated by CMIP5 coupling model and the observation results is very weak, the above overlap will not have a significant impact on the prediction effect of machine learning.

2). Structure of convolutional neural network prediction model.

The convolutional neural network consists of three convolution layers and two maximum pooling layers. The last convolution layer is fully connected with the ordinary neural network. The ordinary neural network contains a hidden layer. The convolution kernel size of the first convolution layer is 8 × 4, and the convolution kernel size of the second and third convolution layers is the grid resolution of 4 × 2. The maximum pool level retrieves the maximum value from the convolution layer with its grid resolution of 2 × 2. In order to obtain a more objective prediction structure, two different numbers (i.e. 30 and 50) of convolution nuclei and hidden layer neurons are tried. For example, C30H50 represents a Convolution Neural Network with 30 convolution nuclei and 50 hidden layer neurons, and so on. At the same time, each convolutional neural network adopts 10 different initial weights for training, and carries out the corresponding return test.

5.3 Return results

Figure 5 shows the temperature index of December, January and February in winter in East Asia which returned one month in advance by the convolutional neural network machine model, which is recorded as Pre _T2m_Dec, Pre_T2m_Jan, Pre_T2m_Feb respectively. The results show that the correlation coefficients between the set average return result of convolutional neural network Pre _T2m_Dec, Pre_T2m_Jan, Pre_T2m_Feb and observation results in December, January and February ERA5_T2m_Dec, ERA5_T2m_Jan, ERA5_T2m_ Feb are 0.77, 0.82 and 0.70 respectively. At the same time, the amplitude of the return index is also close to the observation. It is worth noting that the prediction results of Convolution Neural Network with different numbers of convolution nuclei and hidden layer neurons are not very different. However, the prediction results of different initial fields (shown in the shadow of **Figure 5**) are significantly different. The deepening of neural network can improve the prediction ability of the machine to a certain extent. For example, the return effect of C50H50 in **figure 6**A and **figure 6**B is slightly better than that of C50H30. However, when the neural network structure reaches a certain depth, it becomes particularly important to find the global optimal parameters of the neural network by controlling the initial field. For example, take different initial parameters for the same machine prediction model to train the machine (**Figure 6**: C50H50), the difference between the return result and the observed correlation coefficient can be up to about 0.2 between different sets.

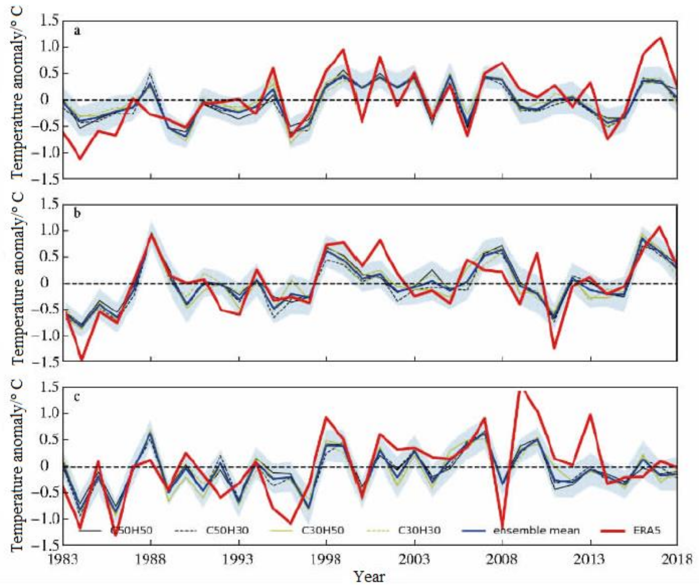

Figure 5. Ensemble-mean time series of (A) December 1982-2017, (B) January 1983-2018, (C) February 1983–2018 aera-averaged (10°-30°N,100°-140°E) T2m anomalies for one-month-lead hindcast using convolutional neural network (CNN) model (blue solid curves) as well as the corresponding observed time series (red curves). Other curves represent results of CNN model with different numbers of convolutional filters and hidden layers; for example, C50H30 indicates the CNN model with 50 convolutional filters and 30 hidden layers, and so on; shading indicates±1 standard deviation of 40 ensemble members.

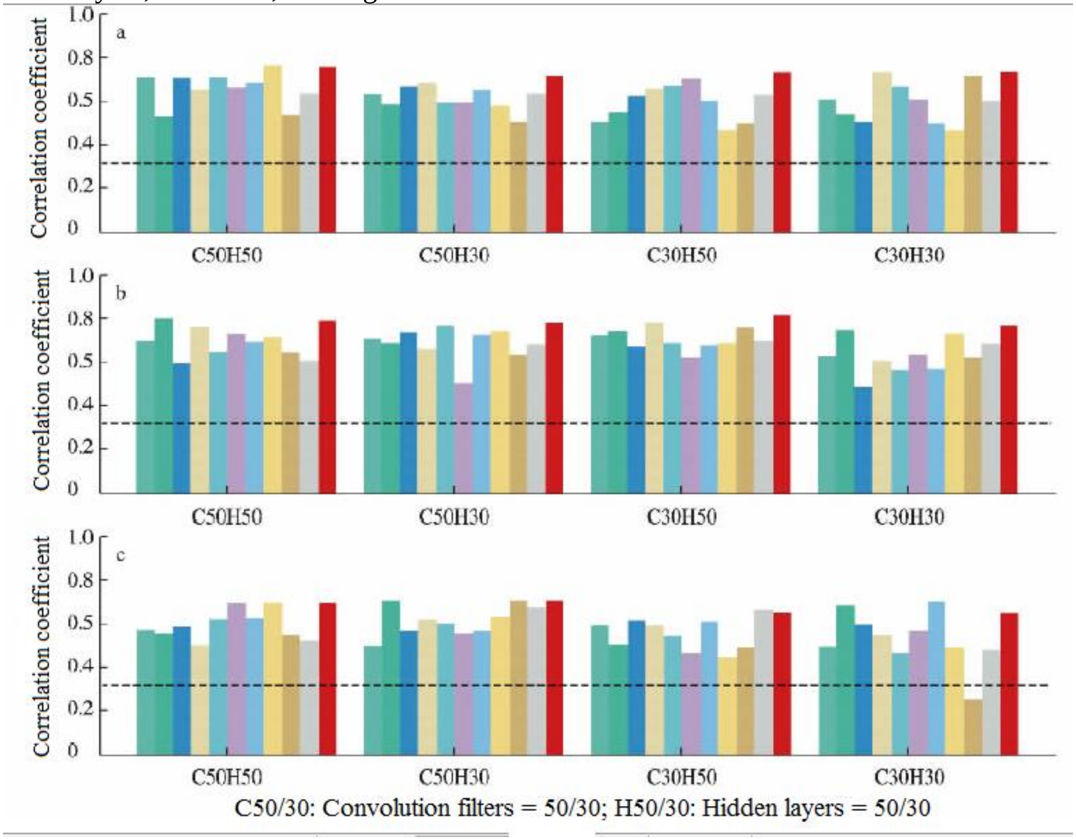

Figure 6. Dataset are the same as Figure 5, but for the correlation coefficients of each ensemble members with the observation in (A) December, (B) January, and (C) February; the red bar indicates the results of ensemble mean in each CNN model; the horizontal dashed line indicates the value at 95% confidence level.

It should be emphasized that the return effect of the in-depth learning model is better than that of the climate dynamic model. As shown in **Figure 7**, the correlation coefficient between the 40 collective return tests and the observation results of the January temperature in a region of East Asia (100° ~ 140°E, 10° ~ 30°N) returned one month in advance by the deep learning model is 0.5–0.8. All passed the 95% reliability test. At the same time, it is also higher than the correlation coefficient (0.42) between the dynamic model CanCM 4i return results and observations. In addition, the temperature skill of a region in East Asia (100° ~ 140°E, 10° ~ 30°N) returned by the deep learning model 2–3 months in advance is generally higher than the return effect of the dynamic model.

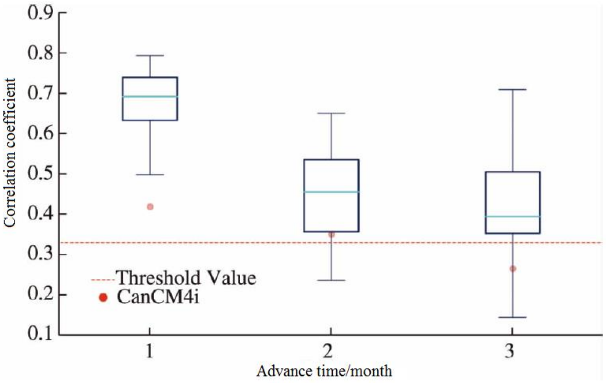

Figure 7. Boxplot for correlation coefficients of observation with each ensemble member's hindcast with one month,two months, and three months in advance; red dot indicate the correlation between the observation and the hindcast by cancm4i; the horizontal dashed line indicates the value at 95% confidence level.

It is worth noting that although the above examples show that machine learning can be applied to short-term climate prediction, however, it does not mean that give any "big data" to the machine, a climate prediction model with good performance can be established. In order to establish a machine learning climate prediction model with high prediction skills, it is necessary to fully understand the climate dynamics behind "big data". In other words, the establishment of machine learning model guided by climate dynamics is very important to give full play to the potential of machine learning in climate prediction. For example, using the same machine learning idea as **Figure 5**B, the return model is established for the average temperature in January at low latitude (100° ~ 140°E, 0° ~ 20°N) and middle latitude (100° ~ 140°E, 30° ~ 50°N). The correlation coefficient between the return result of collective average and the observation result is 0.89 and 0.33 respectively (**Figure 8**). The main reason may be that compared with the climate in middle and high latitudes, the climate in low latitudes is more obviously affected by tropical and subtropical SST (**Figure 9**). The prediction factors in the machine learning prediction model in this paper are mainly the sea surface temperature anomaly of 60°S ~ 60° N and the ocean heat content anomaly of 0 ~ 300m. From the perspective of climate dynamics, the machine learning prediction model in this paper is more suitable for climate prediction in middle and low latitudes. If the machine learning prediction model of mid and high latitude climate is to be established, the impact of mid and high latitude climate system needs to be considered more, such as Eurasian snow, Arctic sea ice, polar vortex, etc. (He *et al.*, 2016; He *et al.*, 2020). It should be emphasized that although the linear regression analysis shows that there is a significant statistical relationship between the temperature anomaly in the low latitude of East Asia and the sea surface temperature in some parts of the world, the return result of the linear regression model based on sea surface temperature is far less than that of machine learning (**Figure omitted**). It further shows the obvious advantages of machine learning in exploring nonlinear processes.

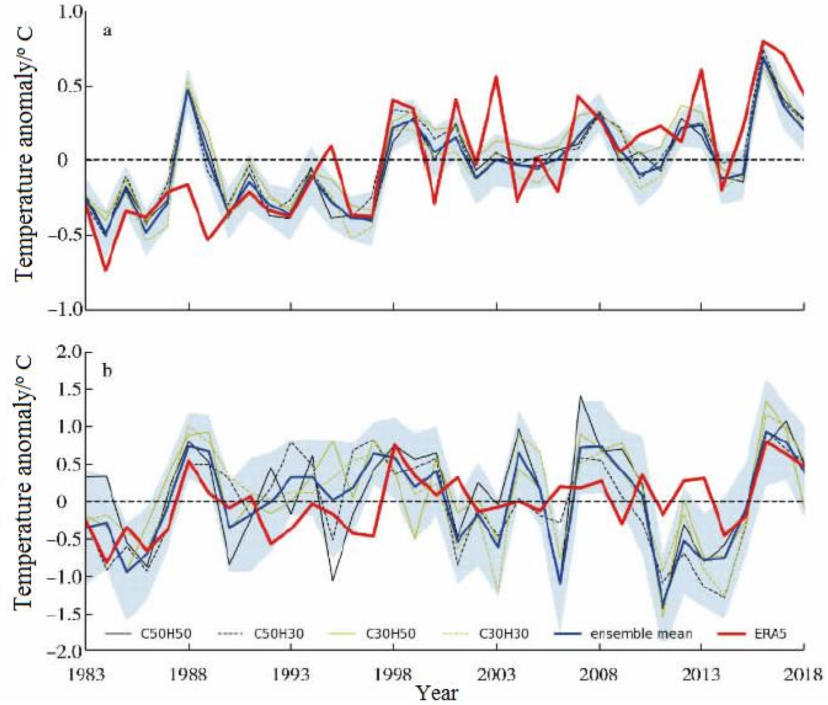

Figure 8. Same as **Figure 5** (B), but for the hindcast of area-averaged T2m anomalies in January 1983—2018 over (A) 0°—20°N,100°—140°E and (B) 30°—50°N,100°—140°E.

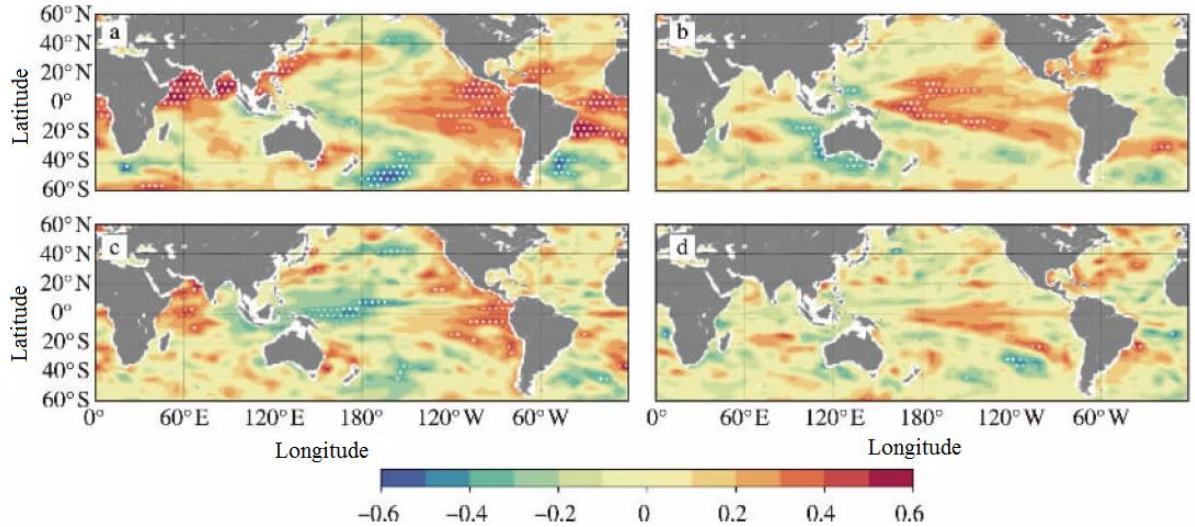

Figure 9. Correlation coefficients (SHADING) of area-averaged T2m anomalies over (0°—20°N,100°—140°E) in January 1983—2018 with the preceding three months'mean (October,November,December) (A) sea surface temperature anomalies and (B) oceanic heat content anomalies from surface to 300m; regions with stippling indicate the values significant at 95% confidence level; (C) and (D) are the same as (A) and (B),respectively,but for the area-averaged T2m anomalies over (30°—50°N,100°—140°E) in January 1983–2018.

6. Summary and Discussion

With the vigorous development of machine learning, this paper focuses on the basic principle of supervised learning of machine learning and analyzes the potential application of machine learning in climate prediction through machine learning examples of linear, nonlinear and deep learning.

Firstly, by introducing a simple example of machine learning to obtain the parameters of linear fitting function, this paper analyzes the significance of "training data", "label data" and "loss function" in machine learning, and shows how machine learning reduces the loss function, updates and optimizes the parameters through "gradient descent" algorithm, and finally obtains a reasonable linear fitting line (**Figure 1**).

Secondly, from the perspective of matrix multiplication, the construction idea from the input layer to the hidden layer and then to the output layer of neural network is analyzed (**Figure 2**). Taking the nonlinear data set as an example, the example of fitting the nonlinear function curve with the neural network machine model is shown, and the learning effects of neural networks with different "excitation functions" are also compared (**Figure 3**).

Then, the basic framework of convolution neural network for deep learning is analyzed, including the function of convolution kernel, the working process of convolution layer and pooling layer, and how the pooling layer is connected to ordinary neural network (**Figure 4**). Finally, it introduces how to build the prediction model of winter monthly temperature in East Asia through CMIP5 "big data" and convolutional neural network, and carry out the return test using the observed data (**Figure 5**, **Figure 6** and **Figure 7**). At the same time, the importance of building machine learning prediction model guided by climate dynamics knowledge is discussed (**Figure 8**, **Figure 9**).

It should be pointed out that machine learning is already a comprehensive discipline, including many algorithms, such as batch gradient descent method, random gradient descent method, small batch gradient descent method, linear regression, logical regression, decision tree, naive Bayes, k-proximity, learning vectorization, support vector machine, random forest, etc. Deep learning is only an important branch of machine learning. Its algorithms include convolutional neural network, cyclic neural network, generative countermeasure network and deep reinforcement learning. This paper only briefly introduces the batch gradient descent algorithm, shallow neural network and Convolution

Neural Network in machine learning, so as to preliminarily understand the principle and function of machine learning and provide some basic knowledge for further understanding of machine learning.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Bauer P, Thorpe A, Brunet G. The quiet revolution of numerical weather prediction. Nature 2015; 525(7567) : 47–55. doi: 10. 1038/nature14956.
2. Behringer D, Xue Y. Evaluation of the global ocean data assimilation system at NCEP: The Pacific Ocean. Proc. Eighth Symp. on Integrated Observing and Assimilation Systems for Atmosphere, Oceans, and Land Surface, Seattle, WA, AMS. 2014 Jan 1. Washington. Washington: Washington State Convention and Trade Center. 2014.
3. Copernicus Climate Change Service (C3S). ERA5: Fifth generation of ECMWF atmospheric reanalyses of the global climate. 2017. Available from: https://cds.climate.copernicus.eu/cdsapp#!/home.
4. Dougherty J, Kohavi R, Sahami M. Supervised and unsupervised discretization of continuous features. Machine Learning Proceedings Amsterdam: Elsevier. 1995. p. 194–202. doi: 10. 1016/b978-1-55860-377-6. 50032-3.
5. Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge: MIT Press. 2016.
6. Ham YG, Kim JH, Luo JJ. Deep learning for multi-year ENSO forecasts. Nature 2019; 573(7775) : 568–572. doi: 10. 1038/s41586-019-1559-7.
7. Hantson S, Arneth A, Harrison SP, *et al.* The status and challenge of global fire odeling. Biogeosciences 2016; 13(11): 3359–3375. doi: 10. 5194 /bg-13-3359-2016.
8. Hasselmann K. Linear and nonlinear signatures. Nature 1999; 398(6730): 755–756. doi: 10. 1038/19635.
9. He S, Wang H, Xu X, *et al.* Impact of Arctic warming and the super El Nino in winter 2015/2016 on the East Asian climate anomaly (in Chinese). Transactions of Atmospheric Sciences 2016; 39(6): 735–743. doi: 10. 13878/j.cnki.dqkxxb.20161008002.
10. He S, Xu X, Furevik T, *et al.* Eurasian cooling linked to the vertical distribution of arctic warming. Geophysical Research Letters 2020; 47(10). doi: 10. 1029/2020gl087212.
11. Huntingford C, Jeffers ES, Bonsall MB, *et al.* Machine learning and artificial intelligence to aid climate change research and preparedness. Environmental Research Letters 2019; 14(12): 124007. doi: 10. 1088/1748-9326/ab4e55.
12. Kuang Z, Song Z, Dong C. Global sea surface temperature over 21st century using a biases correction model based on machine learning (in Chinese). Climate Chang Research Letters 2020; 9(4): 270–284.
13. Lagerquist R, McGovern A, Gagne DJ. Deep learning for spatially explicit prediction of synoptic-scale fronts. Weather Forecast 2019; 34(4): 1137–1160. doi: 10. 1175/waf-d-18-0183. 1.
14. Liu Y J, Racah E, Prabhat, *et al.* Application of deep convolutional neural networks for detecting extreme weather in climate datasets. 2016. Available from: https://arxiv.org/abs/1605. 01156.
15. McCarthy J, Minsky ML, Rochester N, *et al.* A proposal for the Dartmouth summer research project on artificial intelligence. 1956. Available from: http://www.formal.stanford.edu/jmc/history/dartmouth/dartmouth.html.
16. Men X, Jiao R, Wang D, *et al.* A temperature correction method for multi-model ensemble forecast in North China based on machine learning (in Chinese). Climatic and Environmental Research 2019; 24(1): 116–124. doi: 10. 3878/j.issn.1006-9585. 2018. 18049.
17. Pörtner HO, Roberts DC, Masson-Delmotte V. IPCC special report on the ocean and cryosphere in a changing climate. 2019. In press.
18. Ruder S. An overview of gradient descent optimization algorithms. 2016. Available from: https://arxiv.org/abs/1609. 04747.
19. Samuel AL. Some studies in machine learning using the game of checkers. IBM Journal of Research and Development 1959; 3(3): 210–229. doi: 10. 1147/rd.33. 0210.
20. Specht DF. A general regression neural network.

IEEE Transactions on Neural Networks and Learning Systems 1991; 2(6): 568–576. doi: 10. 1109/72. 97934.

21. Stockhause M, Lautenschlager M. CMIP6 data citation of evolving data. Data Science Journal 2017; 16. doi: 10. 5334/dsj-2017-030.

22. Toms BA, Barnes EA, Ebert-Uphoff I. Physically interpretable neural networks for the geosciences: Applications to earth system variability. Journal of Advances in Modeling Earth Systems 2020; 12(9). doi: 10. 1029/2019ms002002.

23. Weyn JA, Durran DR, Caruana R. Can machines learn to predict weather? Using deep learning to predict gridded 500-hPa geopotential height from historical weather data.Journal of Advances in Modeling Earth Systems 2019; 11(8): 2680–2693. doi: 10. 1029/2019ms001705.