Journal of Electronic Research and Application

Research Article

Data Processing and Artificial Neural Network in the Background of Big Data

Yang You

Chemical Engineering and Technology, Zhongshan University, China

Abstract: The existing significance of big data technology lies not only in collecting massive information, but also in professional processing and analysis. It transforms information into data and extracts valuable knowledge from data. The advent of the era of big data has brought us a new development model, but also produced many emerging industries, such as cloud computing, artificial intelligence and so on. Based on this, this paper studies the artificial neural network and back propagation algorithm in this context, so that computer technology can better serve human beings, which is of great significance to promote the further development of artificial intelligence technology.

Keywords: Big data; Artificial intelligence; Neural network

Publication date: March, 2020

Publication online: 31 March, 2020

****Corresponding author:*** Yang You, daimulin@xyzrgroup.com

1 Overview of neural networks in the background of big data

1.1 Concept of neural network

What is artificial neural network? Although we know that human beings can't fully understand how the brain works for the time being, there are a kind of algorithms that imitate its operation form. This kind of algorithm is a function connecting the self-random variable and the cause random variable. We take today's temperature and tomorrow's temperature as the example. Although the temperature of tomorrow cannot be accurately judged by today's temperature, it is absolutely undeniable that tomorrow's temperature is related to today's temperature. And this relationship is the bridge between self-random variable and cause random variable. However, the random variables are full of uncertainty, and the function is a certain relationship, so the artificial neural network lacks theoretical basis so far. Because it can't be proved at present, it is still active in various fields. In short, neural network is a kind of algorithm that imitates human brain neurons and their connections.

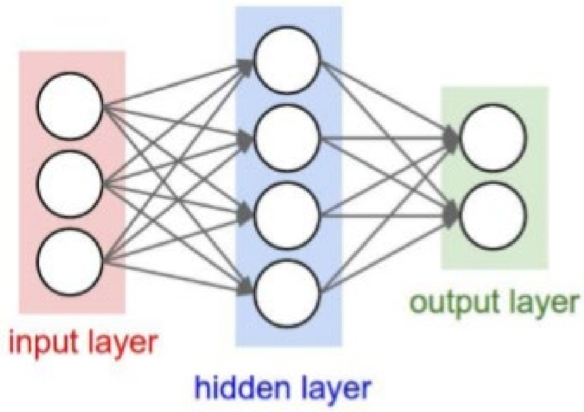

Figure 1. Structure diagram of full connected neural network

The neural network is a neuron connected by the algorithm. Figure 1 is a full connected (FC) neural network, that is, all neurons at the upper level are connected with all neurons at the lower level. Some basic rules of neural network can be obtained from Figure 1.

(1) Neurons are distributed according to layers. “Input layer” is the input layer, which is responsible for receiving input data; “output layer” is the output layer, which is responsible for outputting data. “Hidden layer” is the hidden layer, i.e. a general term of input layer and output layer, because they are invisible to the outside.

(2) There is no connection between the neurons in

each layer.

(3) Layer N neurons and layer N-1 neurons are connected. It is also the meaning of full connection explained above. The output of layer N-1 is the input of layer N.

(4) Every connection has a weight.

1.2 Network neuron and its "nucleus" - activation function

The activation function is similar to the nucleus of a neuron. It determines whether the neuron is activated. The following describes a common activation function namely sigmoid function.

Suppose the input of neuron is vector $\vec{x}$ and the weight vector is $\vec{\omega}$ (the bias term is ω_g), then its output value is $\vec{y}$:

$$\vec{y} = sigmoid(\vec{w}^T \cdot \vec{x})$$

The sigmoid function is defined as follows:

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

Its function image is shown in Figure 2:

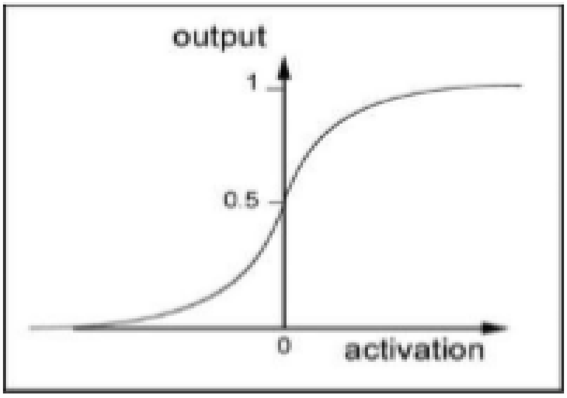

Figure 2. Function diagram

According to this, the general property of activation function is that the codomain of value is between (0,1), and the monotonicity can be transferred and reflected. When the sigmoid function is derivative, we can get: $y^{'} = y(1 - y)$, that is to say, the derivative function of sigmoid function can still be represented by sigmoid function. This is also an advantage as an activation function.

1.3 "axon" of network neuron —— output of neural network

Neural network is actually a function from input vector $\vec{x}$ to output vector $\vec{y}$, that is:

$$\vec{y} = f_{network}(\vec{x})$$

Starting from the input layer, we assign the output value of the upper layer to the input value of the next layer according to the processing way layer by layer, and then the neuron performs the calculation operation again, initially transfers the output value until the final output layer outputs the value.

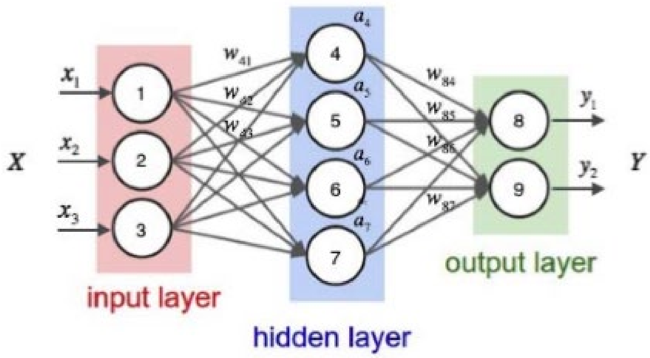

Figure 3. Input and output structure of neural network

As shown in Figure 3, the input layer has three nodes, the hidden layer has four nodes, and the output layer has two nodes. The neural network is a fully connected network, so every few points are connected with each node of its upper layer, and each connection has a weight value.

The output value of the fourth node can be obtained as follows:

$$a_4 = sigmoid(\vec{w}^T \cdot \vec{x})$$
$$= sigmoid(w_{41}x_1 + w_{42}x_2 + w_{43}x_3 + w_{4b})$$

In the above the last term is the bias term.

Similarly, each value of the hidden layer can be obtained, and each value of the output layer can also be obtained. In addition, it can also be expressed by matrix. Here we use the first two layers for example:

$$\vec{a} = f(W \cdot \vec{x})$$

These three proofs are all the input values of the hidden layer, all the weights from the input layer to the hidden layer, and all the output values of the input layer.

2 Analysis of back propagation algorithm

In order to obtain the model, we need to know the super parameters and parameters. Super parameters refer to some fixed and artificially-set values, such as the number of layers and the number of nodes in each layer. The acquisition of parameters requires supervised training. When the trained model is obtained, it can be used to test the test value. Then we can get some error terms, that is, the error term is the error of the output value, and deduce it in reverse. We use the error of the last layer, that is, the output layer, to push forward. For the simple full connected neural network mentioned above, the output layer error can be used to get the hidden layer error, which is the back propagation algorithm.

Each training sample is assumed to be $(\vec{x},\vec{t})$, where vector $\vec{x}$ is the characteristic of the training sample and $\vec{t}$ is the target value of the sample. According to the algorithm described above, the error term of each node can be calculated according to the following formula.

For input layer node i:

$$\delta_i = y_i(1-y_i)(t_i-y_i)$$

In the formula, δ_i is the error term of the node i, y_i is the output value of the node i, and t_i is the target value of the node i corresponding to the sample.

For hidden-layer nodes:

$$\delta_i = a_i(1-a_i)\sum_{k\in outputs} w_{ki}\delta_k$$

Where a_i is the output value of the node i, w_{ki} is the connection weight from the node i to its next-layer node k, and δ_k is the error term of next layer k of the node i.

Finally, we need to update the full value on each connection:

$$w_{ji} \Leftarrow \omega_{ji} + \eta\delta_j x_{ji}$$

Where w_{ji} is the weight from node i to node j, η is the Changshu that becomes the learning rate, δ_j is the error term of the node, and x_{ji} is the input that node i passes to node j.

Therefore, it is necessary to know the error term of the node in the later layer when calculating the error term of the node in some layer. When all the node error terms are completely calculated, the weights can be updated. Then we can get the final model of the whole neural network through forward training and back propagation.

3 Convolution neural network analyses

In the process of image processing, the fully connected neural network has many disadvantages. For example, the number of parameters is too large to use the location information among pixels and the limit of network layers. So convolution neural network can be used to better deal with the task of image recognition.

3.1 Definition of convolution neural network

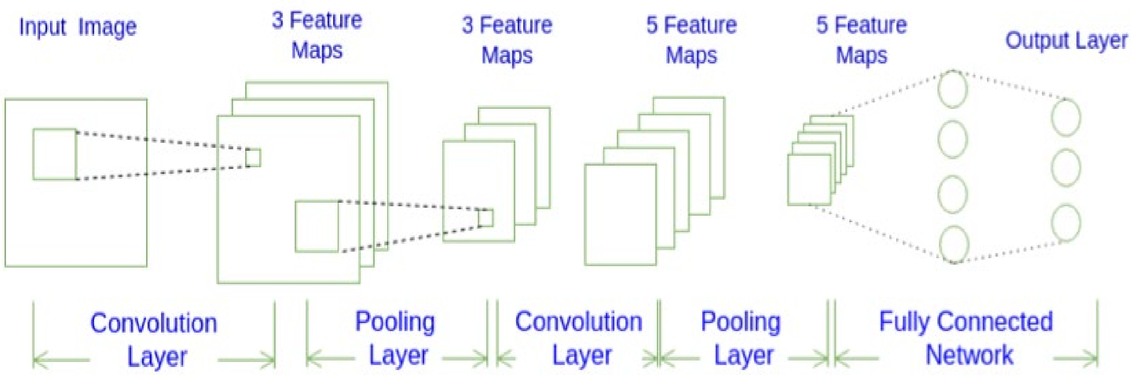

Figure 4. Layer structure of convolutional neural network

It can be seen from Figure 4 that the convolutional neural network consists of multiple convolution layers, pooling layers and full connection layers. Usually the fact is several convolution layers followed by a pooling layer, repeated operations, and finally added a number of fully connected layers. Compared with the fully connected neural network, it has a significant difference, that is, it is multidimensional. In other words, the convolution layer can have multiple filters to get multiple Feature Maps for realizing multidimensional operation.

3.2 Activation function of convolution neural network

Unlike the fully connected neural network, the activation function of convolution network is often relu function. It is defined as:

$$f(x) = max\,(0,x)$$

It has two obvious advantages. First, the calculation is simple and the operation speed can be greatly improved. Second, its derivative is 1, so its gradient will not be reduced, which can greatly increase the number of layers of convolutional neural network.

3.3 Calculation of output value of convolution neural network

Suppose there is a 5 * 5 image, we convolute it with a 3 * 3 filter to get a 3 * 3 Feature Map, as shown in Figure 5.

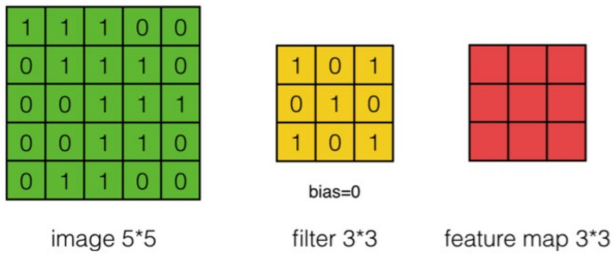

Figure 5. Convolution neural network flow chart

For the convenience of description, each numbered pixel point can be used to represent the elements of row i and column j of the image with $x_{i,j}$; for each weight in the filter, we use $w_{i,j}$ to represent the weight of row i and column j, and use w_b to represent the bias term of the filter; we use $a_{i,j}$ for the elements of row i and column j in the Feature Map; the activation function is represented with f, and then we calculate the convolution with the following formula:

$$a_{i,j} = f\left(\sum_{m=0}^{2}\sum_{n=0}^{2} w_{m,n} x_{i+m,j+n} + w_b\right)$$

If the activation function is a relu function, we can find out:

$$\begin{aligned} a_{0,0} &= relu\big(w_{0,0}x_{0,0} + w_{0,1}x_{0,1} + w_{0,2}x_{0,2} \\ &+ w_{1,0}x_{1,0} + w_{1,1}x_{1,1} + w_{1,2}x_{1,2} \\ &+ w_{2,0}x_{2,0} + w_{2,1}x_{2,1} + w_{2,2}x_{2,2}\big) \\ &= relu(1+0+1+0+1+0+0+0+1+0) = 4 \end{aligned}$$

As shown in Figure 6, all values can be obtained in turn.

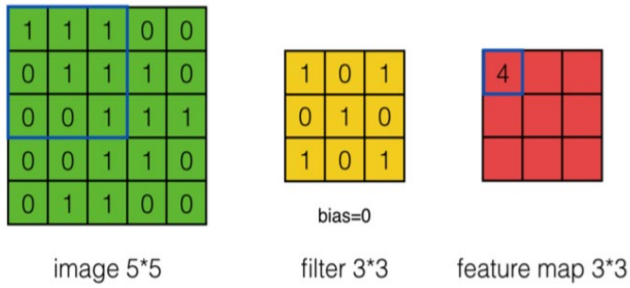

Figure 6. Output value calculation diagram of convolution neural network

References

[1] Mou Q. Discussion on Application and Practice of Computer Information Processing Technology in the Background of Big Data[J]. Network Security Technology and Application, 2019 (11): 65-66.

[2] Hu HP, Bai HQ, Bai YP, et al. Prediction of Flu-Like Based on Improved Genetic Algorithm and Artificial Neural Network[J/OL]. Journal of North China University (Natural Science Edition), 2019 (6): 1-8.

[3] Qian Y, Sun JH, Zhang Y. Research on Intelligent Algorithm in Agricultural Big Data[J]. Northern Horticulture, 2019 (20): 156-161.

[4] Kang CL. Study on the Optimization of BP Neural-Network Nonlinear Function Fitting by Genetic Algorithm[J]. Journal of Xinzhou Normal University, 2019, 35 (5): 22-25.

[5] He R, Huang JH. Algorithm of Data Mining Technology in the Background of Big Data[J]. Electronic Technology and Software Engineering, 2019 (20): 141-142.