The New Algorithm of BP Artificial Neural Network

Mao Luo1, a , Shaoyun Song 2,b
1 School of information Technology and Engineering,Yuxi Normal University, Yuxi,Yunnan,China
aluomao@yxnu.net, bssy@yxnu.net

Abstract

Propose a new algorithm based on BP neural network algorithm on existing, which basic principle is choosing a freedom weight, by solving the linear equations to achieve hidden layer, combination freedom weight , then obtain weight is necessary weight. This algorithm hasn’t the traditional method such as the local minimum and the slower rate of convergence in BP neural networks algorithm.

Keywords: Neural network, MATALB simulation, new BP Algorithm

Introduction

BP (Back Propagation) network is the 1986 team of scientists headed by Rumelhart and McCelland proposed an error back propagation algorithm by former training multilayer feed-forward network, is one of the most widely used neural network model. BP network can learn and store large amounts of input - output mode mappings, without prior reveal mathematical equations that describe the mapping relationships. It's learning rule is to use the steepest descent method, back-propagation through the network to continuously adjust the weights and thresholds, so that the smallest error sum of squares of the network. BP neural network topology includes an input layer (input), hidden layer (hide layer) and output layer (output layer).

BP artificial neural network is to modify the weights between each layer of neurons by gradient descent method, the error declining to achieve the desired accuracy. Essentially, this calculation process is an iterative process, iterative algorithms are generally closely associated with the initial choice, a good choice of the initial value, the convergence speed is fast, if the initial choice is not good, then the slow convergence or simply do not converge. When using gradient descent algorithm to adjust the weights, there is inevitably fall into local minima problems. Although a lot of papers have reported a variety of improved methods to speed up the convergence, Increase the inertia term (also called momentum) and other methods. However, since these methods are iterative and optimization of the basic idea, it is impossible to resolve dependencies and local minima of the initial value problem fundamentally.

BP network structure and parameter assumptions

Consider the three-tier structure of BP network, as shown in Figure 1. Network parameters assumptions are as follows:

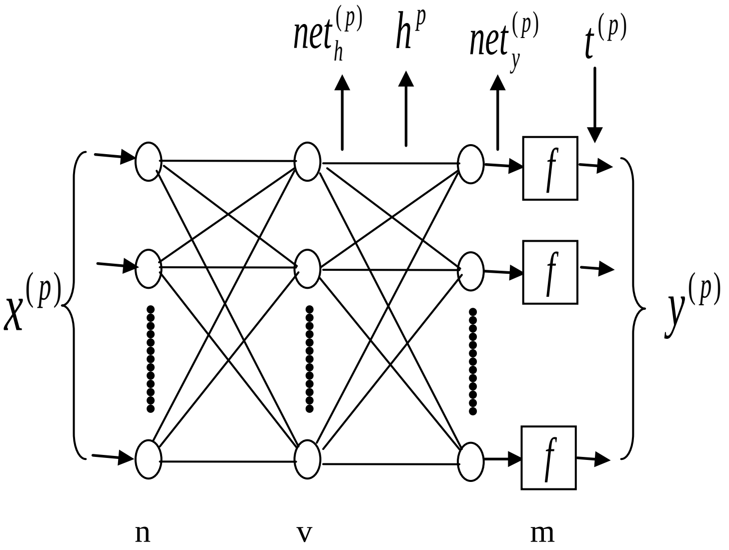

Figure 1 three BP network structure

(1) the number of samples: *k*
(2) the number of input Node: *n*
(3) the number of hidden layer neurons: *v*
(4) the number of output layer neurons: *m*
(5) the input vector: x^p
(6) Weighted hidden layers and vectors: net_h^p
(7) hidden layer output vector: h^p
(8) the output layer weights and vectors: net_y^p
(9) the output layer output vector: y^p
(10) The teacher vectors: t^p

Where $p = 1,2,3, ..., k$. Input layer and hidden layer, hidden layer and output weighting matrix layer are:

$$W_f = \begin{bmatrix} w_{11}, w_{12}, \ldots, w_{1n} \\ w_{21,} w_{22}, \ldots, w_{2n} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ w_{v1}, w_{v2}, \ldots, w_{vn} \end{bmatrix} \qquad \omega = \begin{bmatrix} \omega_{11}, \omega_{12}, \ldots, \omega_{1v} \\ \omega_{21,} \omega_{22}, \ldots, \omega_{2v} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \omega_{m1}, \omega_{m2}, \ldots, \omega_{mv} \end{bmatrix}$$

Weight matrix between the input and hidden layers are *v* × *n* dimensional matrix，it can be free , the value of taking a random number *(-1,1)* within the matrix between the hidden layer and output layer is *m* × *v*-dimensional ,it is the unknown matrix.

The new algorithm

On the net_y^p , *i*-th component of the *k*-th of input samples to the output layer output vector

$$\begin{cases} h_1^1\omega_{i1} + h_2^1\omega_{i2} + \cdots + h_v^1\omega_{iv} = net_y^1 i \\ h_1^2\omega_{i1} + h_2^2\omega_{i2} + \cdots + h_v^2\omega_{iv} = net_y^2 i \\ \cdots\cdots\cdots \\ h_1^k\omega_{i1} + h_2^k\omega_{i2} + \cdots + h_v^k\omega_{iv} = net_y^k i \end{cases} \tag{1}$$

Assuming performance function of neural network output layer is $y = \lambda x$, where λ is a constant.in formula (1), Where *x* refers to any of a neuron's net input, *y* represents the net input neuron to its actual output *x* time.

When $x = \frac{1}{\lambda} y$ then $net_y^p i = \beta t_i^p$,where p=1，2，…，k (2)

If the output layer performance function is $f(x) = \frac{1}{1+e^{-x}}$ then $x = \ln(\frac{y}{1-y})$ and so $net_y^p i = \ln(\frac{t_i^{(p)}}{1-t_i^{(p)}})$.

Assume $d_i = [net_y^1 i + net_y^2 i + \cdots + net_y^k i]^T$ then equation (1) can be expressed as follows:

$$[h]_{k\times v}\omega_i = d_i \tag{3}$$

Equation (3) is a known amount of the right side, the left side of ω_i is the amount to be requested.

$$h_i = f(net_h^p i) = f(\sum_{j=1}^{m} x_i w_{ij})$$

(4)

Among them, let hidden layer performance function for sigmoid function.It is following:

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

(5)

According to the basic theorem of linear equations known: formula (5) has solution of necessary and sufficient conditions that is equal rank of $[h]_{k\times v}$ and $[h,b]_{k\times(v+1)}$ augmented matrix . Another aspect, as each row is independent of a linear $[h]_{k\times v}$, then *v* less than time *k*, In general (5) no solution, thus requiring *v* greater or equal *k*. In order to make the structure more simple for neural network ,should be as small as possible so that the *v*, so choose *v*=*k*, time as long as $[h]_{k\times k}$ is non-singular, the formula (5) will have a unique solution $\omega_i=[\omega_{i1},\omega_{i2},\cdots,\omega_{ik}]$, where (*i = 1, 2, ..., m*), then we can obtain a matrix to be seeking power. Equation (5) Total demand solution m of *k*-order equations, solving ignored when initializing a working computer.

Matlab Tools Boxes

matlab neural network toolbox has all the usual basic model of neural networks, such as perception and BP neural network. The various neural network model, matlab integrates a variety of learning algorithms. BP neural network commonly used functions show in the table 1.

Table 1 BP neural network commonly used functions

Function Type	Function Name	Function uses
The create functionof Forward feedback network	newcf	Create a cascade forward the feedback network
	newff	Create a forward the feedback BP network
The transfer function	logsig	S type of logarithmic function
	tansig	S type of tangent function
	purelin	Pure linear function
Learning function	Learngd	Based on gradient descent learning function
	Learngdm	Momentum gradient descent learning function
Performance function	Mse	Mean square error function
	Msereg	MSE normalization function

Display Functions	Plotperf	Draw a grid function
	Plotes	Draw error surface of a single neuron
	Plotep	Draw weights and thresholds position on the error surface
	errsurf	Calculation error surface of single neurons

Algorithm Analysis

The optimization algorithm abandoned the traditional thinking patterns of samples given by selecting the structure of the network, selected a group of freedom weight, and then directly solving a set of linear equations to be seeking to obtain another set of rights. The new algorithm teachers return value only once. This makes the mapping problem between the original sample mode iterative algorithm to optimize the objective function by traditional transformed into problem solving linear equations, thus greatly simplifying the calculation and improve the computing speed. In this paper, out PCA Gaussian elimination method for solving linear equations, it's time complexity is polynomial single.

In order to compare the traditional BP algorithm and the new algorithm, in times of traditional BP algorithm to make an introduction. The step of BP algorithm is as follows:

(A) randomly selected a set of initial weights [W].

(B) calculate the total error $e=\frac{1}{2}\sum_{p=1}^{k}\sum_{i=1}^{m}(y_i^{(p)}-t_i^{(p)})^2$.

(C) If $e<\varepsilon$ (ε is given resolution), the computer stops the output weights.

(D) If $e>\varepsilon$ then corrected gradient descent weights, then return (2).

Gradient descent method is as follows:

$\omega(s+1)=\omega(s)+\eta(-\frac{\partial E}{\partial\omega})$, Where η is the learning rate.

$$\frac{\partial e}{\partial\omega j}=\sum_{p=1}^{k}y_i^{(p)}(1-y_i^{(p)})(y_i^{(p)}-t_i^{(p)})h_j^{(p)}$$

$$\frac{\partial e}{\partial w_{ij}}=\sum_{p=1}^{k}[h_i^{(p)}(1-h_i^{(p)})x_j^{(p)}\sum_{q=1}^{m}(y_q^{(k)}-t_q^{(p)})(1-y_q^{(p)})w_{qi}]$$

Comparison of Algorithms

The new algorithm can solve the image encoding, XOR, and many other issues, there is only one more XOR problem with Matlab computer simulation comparison (*n = 2, v = 4, m = 1*), in order to converge the conventional method, take $\eta = 1.8, \varepsilon = 10^{-6}$, the initialization power is a set of values and a new algorithm ,show in Table 2.

Table 2 compares the results of simulation

New algorithms	Input Signal	x^1=(1,1)，x^2=(0,1) ，x^3=(1,0) ，x^4=(1,1)
	Teacher Signal	t^1=0.001，t^2=0.999，t^3=0.999，t^4=0.001
	The Initial Random Value	w_{11}=3.98，w_{12}=-5.6，w_{21}=-2.08，w_{22}=6.38 w_{31}=-3.10，w_{32}=8.68，w_{41}=--8.8，w_{42}=9.6
	Solve Wait Weight	ω_{11}=1.1181，ω_{12}=-15.5034，ω_{13}=-14.4755，ω_{14}=1.0000
	Time Consuming	Less than 1 Seconds
BP algorithms	Initialization Weight	w_{11}=4.0，w_{12}=-6.0，w_{21}=-2.0，w_{22}=6.0 w_{31}=-3.0，w_{32}=8.0，w_{41}=--9.0，w_{42}=10.0
		ω_{11}=-1.1181，ω_{12}=-15.5034，ω_{13}=-14.4755，ω_{14}=1.0000
	Convergence value	w_{11}=4.01，w_{12}=-5.99，w_{21}=-2.06，w_{22}=5.97 w_{31}=-2.98，w_{32}=8.01，w_{41}=--9.02，w_{42}=9.98
		ω_{11}=0.7836，ω_{12}=-15.0039，ω_{13}=14.7439，ω_{14}=0.9920
	Iterative Number	1560
	Time Consuming	58 Seconds

Conclusion

In this paper, the abandoned gradient descent optimization idea in traditional BP algorithm, a novel to solve the right through the preliminary liberty thought, the nonlinear mapping complex (Lenovo) problem is transformed into solving a set of linear equations of the topic, thus greatly speed up and simplify the calculation process, its time complexity is polynomial. Fundamentally overcome the convergence speed of the conventional BP algorithm is slow, easy to fall into local minimum problem, the computer simulation results show that the algorithm is very effective.

References

[1] Zhang Liming. Artificial neural network [M]. model and its application to Fudan University press. 1992

[2] Wang Hongyuan, Shi Guodong. Artificial neural network technology and its application [M]. Beijing: Sinopec press, 2002

[3] Wang Xu, Wang Hong, Wang Wenhui. Artificial neural network principle and application [M]. Shenyang: Northeastern University press, 2000

[4] Li Guangqiong, Jiang and volt. Research on improvement of BP neural network algorithm [J]. Journal of Changde Teachers College (NATURAL SCIENCE EDITION), 2003,15 (2): 31-34.

[5] Zhang Daiyuan, an artificial neural network algorithm [J]. new computing technology and automation, 2000,19 (2): 35-38.

[6] An improved [J]. Journal of Qinghai University [6] the king of Qinghai.BP neural network algorithm (NATURAL SCIENCE EDITION), 2004,22 (3): 82-84.