Fashion colour forecasting based on BP neural network

Xiaohong Kong[1*], Lixia Chang[2], Junpeng Xu[1]

[1]*School of Mechanical and Electrical Engineering, Henna Institute of Science and Technology, China, 453003, Henan, China*

[2]*School of Fashion, Henna Institute of Science and Technology, China, 453003*

Received 12 July 2014, www.cmnt.lv

Abstract

Fashion colour forecasting is a hot issue in fashion industry and also a hard problem because of much uncertain information. Utilizing strong mapping capability of BP Neural Network (BPNN) for nonlinear function, this paper investigated the forecasting model of fashion colour. Based on colour data of recent several years, the forecasting model for future colour trend is discussed and built. The historical data are input to train the Neural Network weights and the different parameters of BPNN were investigated to find how to affect the forecasting performance. The results demonstrate the algorithm is very efficient in colour forecasting and can approximate nonlinear relationship of fashion colour very close.

Keywords: Fashion colour, BP neural network, Trend forecasting, Network parameters

1 Introduction

Fashion colour is the colour or colours that the most people prefer in a given period of time. It is very important to forecast the future colour trend that will be popular in 24 months later during the fashion selling seasons and offer the trend palettes for clothing enterprises [1-2]. The obtained palettes will offer the industries useful information and guidance in marketing and manufacturing [3]. And it has long been considered as the most powerful driving forces. Therefore, accurate forecast of fashion colour is not only a marketing strategy of garment industry, but also a hot issue in international fashion market. If the market products do not meet pop elements, it would be unsalable or have to be withdrawn from the market [3-4].

Currently, many authoritative associations on colours release fashion colour palettes every year such as INTER-COLOUR, JAFCA CFCA, PANTONE®, and so on, but little detailed information is known about the forecasting methodologies dealing with this complex and uncertain process [5]. What's more, the lag and confidence of palettes during propagating restrict its commercial value, which stimulates the development of research on fashion colour forecasting. So far, several theories and methods have been applied into this field by using statistical analysis [6-7], grey theory [5, 8], artificial neural network [9-12] and hybrid models [13-14]. Although some achievements obtained, fashion colour prediction is still in exploring stage on the whole and few convincing forecasting systems had been established. Deficiencies and controversies are still existed and the validity of prediction has yet been a pressing problem of the day in the textile and clothing industries [15].

Fashion colour forecasting involves various factors such as sales psychology, geographical, social psychology, political situation, climate, season, skin colour, education level [2-3]. These factors are all interrelated to each other and the fashion colour palettes usually have non-linear features, so forecast of fashion colour becomes a very difficult task. Therefore, an effective forecasting system is required to be established. Utilizing strong function nonlinear mapping capability of Back Propagation neural network (called BPNN)[16], this paper carried out simulation and forecast of fashion colour and built the forecasting model. It is well-known that a single hidden layer BP neural network can approximate arbitrary nonlinear mapping relationship [9, 15-16]. The successive 6 years' fashion colour palettes, released by INTER-COLOUR, from 2007 to 2012 are taken as the objects in this study. The colours are pre-treated using PANTONE colour system and used as input parameters. In some degree, colour data of recent years reflect the trend and greatly reduce the uncertain character. Finally, the different parameters of BPNN were investigated to find how to affect the forecast performance.

2 Algorithm model

2.1 BP NEURAL NETWORK (BPNN)

The artificial Neural Network algorithm mimics biological neural networks, and it is composed of many neurons and adjustable weights connection [16]. Artificial neural network system is made with a large-scale parallel processing, distributed information storage, good self-organizing and self-learning ability and other characteristics. It is widely used in the information processing, pattern

* ***Corresponding author*** e-mail: nancykong@hist.edu.cn

recognition, intelligent control and system modelling and other areas to adapt systems to the environment, find the law of things and recognize or control object [9-12, 17-19]. In particular, the error back propagation algorithm can approximate nonlinear function with strong mapping ability and the colour forecasting is the case. In this network, data stream propagate forward and error signal back-propagation. The network can be represented by a directed acyclic graph $G=(N,W)$, where N is a set of neurons and W is a set of weights, corresponding to the data dependence among neurons. The weights are associated with the amount of data units to be transferred from one neuron to another neuron.

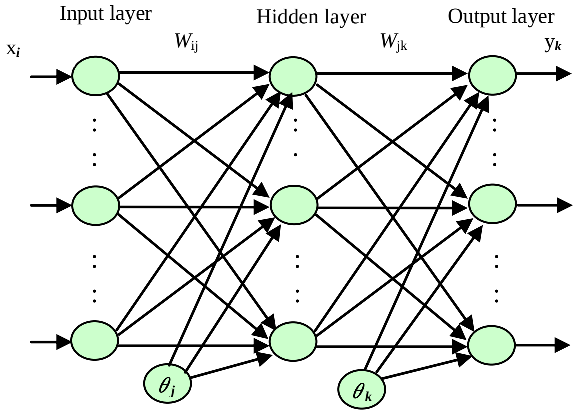


FIGURE 1 The architecture of a BPNN model

BP neural network is suitable to solve the problems of fuzzy and uncertain system and the most commonly used network is a three-layer feed-forward model shown in Figure 1 [16]. It consists of one input layer, one hidden layer and one output layer. The input layer has n inputs and the output layer m outputs, denoted as $x_i(i=1,\cdots,n)$, and $y_k(k=1,\cdots,m)$ respectively. Suppose the hidden layer has h neurons. The neurons of adjacent layers are fully connected with weights w_{ij} and w_{jk} and the data only flow from the input layer to output layer. The neurons of hidden layer and the output have different transfer function named as f_h and f_o. When the $x_i(i=1,\cdots,n)$ is input to network, $\sum_{i=1}^{n} w_{ij}x_i (j=1,\cdots,h)$ are applied to h neurons together with bias θ_j of hidden layer. The transfer function determines the output of every neuron. The output layer has the same process and finally $y_k(k=1,\cdots,m)$ comes out. θ_k represents the bias of the output layer.

BP algorithm is a supervised learning algorithm. In this case, the weights are not known ahead and obtained from sample data. The main idea is: the BP model must be trained by input samples; During the training process, error back-propagation algorithm is used to adjust weights and the biases repeatedly, and it is ensured that the output vector and the desired vector is as close as possible; When the error of output layer is less than a specified error, the training is finished and the network weights and biases are saved for further forecasting. The specific steps are as follows [16]:

(1) Initialize the weights (w_{ij}, w_{jk}) and bias (θ_j, θ_k).

(2) Calculate the output of the hidden layer neuron according to Equation (1).

$$a_j = f_h\left[\sum_{i=1}^{n} w_{ij}x_i - \theta_j\right]. \tag{1}$$

(3) Calculate the output of output neuron according to Equation (2).

$$y_k = f_o\left[\sum_{j=1}^{h} w_{jk}b_j - \theta_k\right], \tag{2}$$

if the bias $\mathbf{\theta_k}$ is considered as $-1\times w_{0j}(w_{0j}=\theta_k)$, Equation (2) can be rewritten as $y_k = f_o\left[\sum_{j=0}^{h} w_{jk}b_j\right]$.

(4) Calculate the error between the desired output and network output according to Equation (3), t_k is the target output.

$$E=\sum_{k=1}^{m}(t_k-y_k)^2 . \tag{3}$$

(5) Error back propagation and the weights are updated according to Equation (4).

$$\begin{aligned}\Delta W(t+1) &= \eta\delta[W(t), X(t), T(t)]X(t) \\ W(t+1) &= \Delta W(t+1)+W(t)\end{aligned}, \tag{4}$$

where W denotes the connection weight matrix, X denotes the input vector and T the target vector, $\boldsymbol{\delta}$ represents the back propagation error based on different learning rule and η denotes the learning rate, η >0.

(6) Repeat step (2) –step (5) until the error satisfy the performance.

2.2 BPNN MODEL FOR THE FASHION COLOUR PREDICTING

In this study, a BP neural network with three-layer is adopted. The neurons of the hidden layer use sigmoid transfer function and the output layer linear function [20]. The nonlinear transfer functions can represent nonlinear and linear relationships between input and output vectors of the network. The linear neurons of the output layer allow the network to produce values outside the range [–1,+1]. BPNN is a negative gradient descent algorithm. There are two ways to realize the gradient descent algorithm, incremental mode and batch mode. In the incremental model, the gradient is calculated and the weights are updated after each input is applied to the network. In the batch mode, only when the whole sample has been trained, the weights and bias are updated. The negative error gradients of training samples determine the change of the weights and bias.

BPNN is powerful to distinguish the complex mode with its strong learning ability and generalization ability. The algorithm transform a set of the input/output samples into a nonlinear optimization problem by iterative lear-ning method, but its convergence speed is slow and easy to fall into local minima. In order to solve the network instability of learning defects, many improved algorithms are raised. In this paper, several considered versions are shown in Equation (5).

$$\Delta W(t+1) = \eta\delta[W(t), X(t), T(t)]X(t) + \alpha\Delta W(t), \tag{5}$$

where α is the momentum factor, $\alpha \in (0,1)$ and $\alpha\Delta W(t)$ reflects the experience accumulated previously.

The learning rate is a multiple of the negative gradient to determine the changes of the weights and biases. An adaptive algorithm will be adopted and learning rate η will be adjusted during the training period. The weights are updated after all of the inputs are applied to the network. A flow chart of the BP is shown in Figure 2.

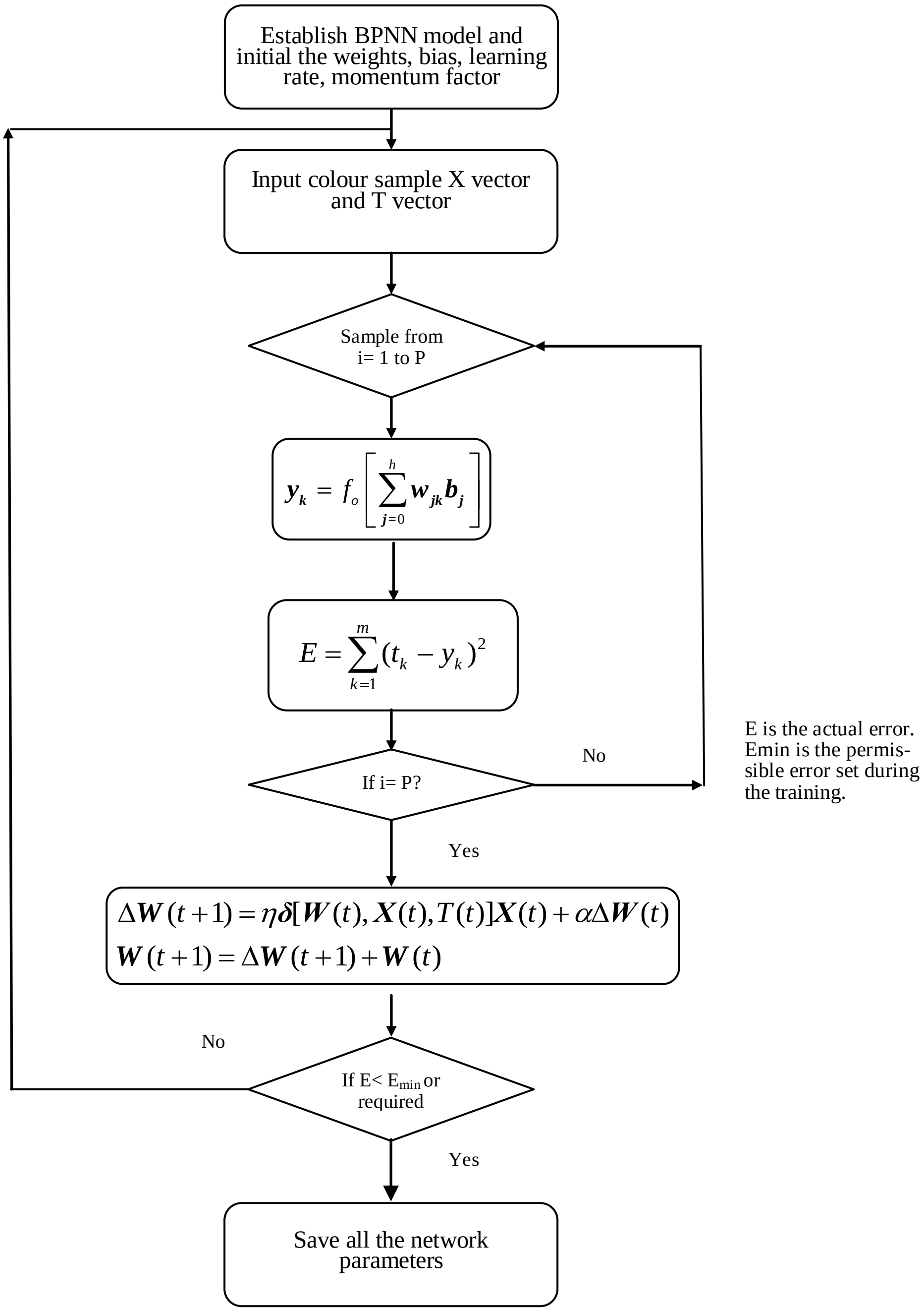


FIGURE 2 The algorithm flow diagram

3 Experiments based on BPNN algorithm

3.1 TEST BENCHMARK

Fashion colour conveys consumers a tendency of tones that will be widely accepted while not specific one or several colours. Generally, more than 40 colours can appear in every palette, and there is no exact corresponding colours in every year, therefore, the colours must be classified and quantified according to the colour theory on hue, lightness and chroma, respectively. Fashion Colour palettes for women's Spring/summer, released by INTER-COLOUR are taken as the test benchmark in this study. These successive fashion colour palettes from 2007 to 2012 were collected. In order to exploit the historical data in BP algorithm, the colours are quantified to numeric data by applying Pantone colour system by PANTONE INC, which is influential in colour specification. The detailed quantification theory and method can be seen in our previous research results [6, 11].

The study only discussed the hue feature. The colours are firstly reflected as their corresponding domains in the Pantone colour space, shown in Table 1[6]. And the colours are transformed further according to Equation (6) and here P_i represents the proportion in the palette.

$$P_i = \frac{n_i}{N}, \tag{6}$$

where n_i depicts the numbers of the colours reflected into the corresponding domains. i denotes the different colour domains; and N indicates the whole numbers of colours in every palette. The quantified results were listed in Table 2.

TABLE 1 Colour domain in the Pantone colour space

Domain	Colour	Domain	Colour	Domain	Colour	Domain	Colour	Domain	Colour
[6-10]	Yellow	[11-14]	Yellow/red	[15-16]	Red	[17-24]	Red/Purple	[25-33]	Purple
[34-38]	Blue /Purple	[39-47]	Blue	[48-52]	Blue /Green	[53-59]	Green	[60-64] [0-5]	Yellow /Green

TABLE 2 Colour ratio in the palettes from 2007 to 2012

Year	Yellow	Yellow/Red	Red	Red/Purple	Purple	Purple/Blue	Blue	Blue/Green	Green	Green/ Yellow
2007	0.242	0.069	0.104	0.034	0.103	0.034	0.242	0.034	0.034	0.104
2008	0.279	0.047	0.07	0.023	0.023	0.093	0.233	0.023	0.047	0.162
2009	0.130	0.218	0.043	0.065	0.022	0.043	0.217	0.001	0.108	0.153
2010	0.244	0.245	0.019	0.113	0.001	0.001	0.188	0.038	0.057	0.094
2011	0.190	0.119	0.072	0.095	0.001	0.001	0.214	0.095	0.025	0.190
2012	0.166	0.194	0.001	0.083	0.028	0.001	0.222	0.056	0.027	0.222

3.2 MODEL PARAMETERS SELECTION

The numbers of the hide layers, the number of processing units and network learning coefficient and other parameters have great flexibility and must be set depending on the circumstances. In many applications, these parameters play an important role. There is no theoretical guidance for selecting the hide layer of the network and the number of units. If BP neural network parameters are not fit, the network cannot guarantee convergence to the global optimal point.

The algorithm is implemented and simulated with neural network toolbox in Matlab 7 [20]. In order to investigate network performance, different parameters were considered as the following:

1) the weights (W_{ij}, W_{jk}) were randomly initialized between (-0.5, +0.5).
2) the learning rate η is adaptive with the training process.
3) the momentum factor α uses the default value.
4) Mean squared error performance function is chosen as target function and target error is set as 0.001.
5) maximum iterations =1000
6) The size of hidden layer neurons, denoted as $Nsize$. $Nsize$ is determined by Equation (7).

$$Nsize = \sqrt{n+m} + (1 \sim 8). \tag{7}$$

Here, n and m represent the number of input layer neurons and output layer neurons.

Sigmoid and linear functions were selected as the transfer function of hidden layer and output layer. In Matlab environment, there are several variations of gradient descent algorithms Different training function and learning function were selected from the following functions [20]:

traingd – Gradient descent backpropagation.

Traingdm – Gradient descent with momentum backpropagation.

Traingda – Gradient descent with adaptive lr backpropagation.

Traingdx – Gradient descent w/momentum & adaptive lr back propagation.

trainlm – Levenberg-Marquardt back propagation.

Learngd – Gradient descent weight/bias learning function.

Learngdm – Gradient descent w/momentum weight/ bias learning function.

3.3 TRAINING AND SIMULATING

When the BPNN was trained with the colour samples, the weights were modified continuously. The training stops if the number of iterations exceeds epochs or if the error $E < E\min$.

3.3.1 Single factor and single colour predicting

When the BPNN is trained, the three successive years' fashion colour data is input and the fourth year's data is the output. That is to say, the recent three years of data roll forecast the fourth year trends. In this mode, only one colour of the past three years influences the same colour of the fourth year. For example, the data of 2007, 2008, and 2009 is the input data and the data of 2010 is the output. And so on, the training samples of the yellow are listed in Table 3.

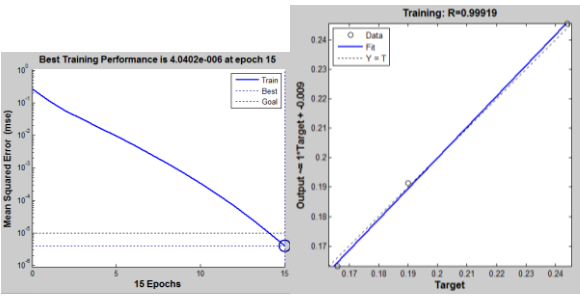
After training the network, the original input vectors were applied the trained network and the simulating results are listed as Table 4.

TABLE 3 Training samples of the hue yellow

Colour	Inputs			Output
Yellow	0.242	0.279	0.130	0.244
	0.279	0.130	0.244	0.190
	0.130	0.244	0.190	0.167

TABLE 4 Single factor forecast of the hue yellow

Year / Value	2010	2011	2012
Actual value	0.2440	0.190	0.1660
Forecast value	0.2454	0.1914	0.1632
Error	-0.0014	-0.0014	0.0028



(a) (b)

FIGURE 3 (a) Performance plot; (b) Regression plot

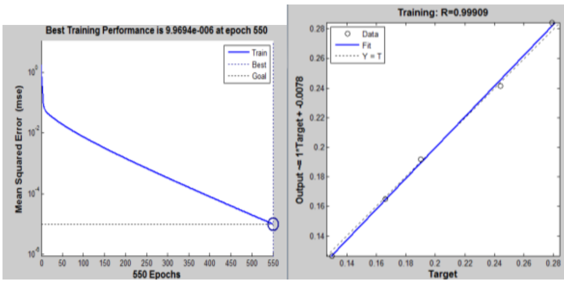
There are two plots in Figure 3 that describe the performance plot and the regression plot respectively. The performance plot shows the mean squared error of the network began with a large value and gradually decreased to a smaller value. It also shows the learning process. The regression plot illustrates regression analysis between the network output and the expecting targets. A dashed line indicates the best linear fit and the solid line shows the perfect fit (output equal to targets). In this example, the best linear fit line is very close to the perfect fit line and the algorithm is very good.

3.3.2 Multiple factors adjacent predicting

Colour forecast is a different task and is affected by many factors. It is not enough to use one colour data of the past three years. In this mode, the next year colour is determined by the ten colours of the former year. In this mode, initial η=0.1, $Nsize$=8, training function= 'traingd', learning function='learngdm' in first experiment. The simulation results were listed in Table 5 and Figure 4.

TABLE 5 Multiple factor forecast of the hue yellow

Year / Value	2008	2009	2010	2011	2012
Actual value	0.2790	0.1300	0.2440	0.1900	0.1660
Forecast value	0.2841	0.1265	0.2416	0.1921	0.1649
Error	-0.0051	0.0035	0.0024	-0.0021	0.0011



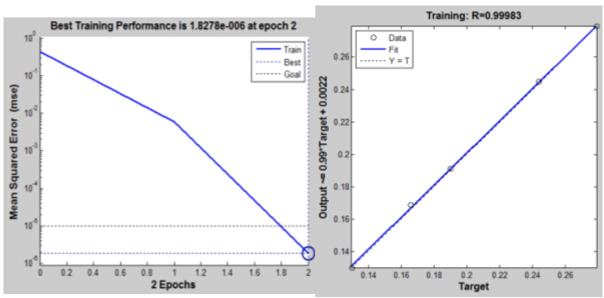
(a) (b)

FIGURE 4 (a) Performance plot; (b) Regression plot

In the second experiment, the training function is modified as 'trainlm' and the other parameters remain unchanged. The experimental results are shown in Table 6 and Figure 5.

TABLE 6 Multiple factors forecast of the hue yellow

Year / Value	2008	2009	2010	2011	2012
Actual value	0.2790	0.1300	0.2440	0.1900	0.1660
Forecast value	0.2841	0.1265	0.2416	0.1921	0.1649
Error	-0.0051	0.0035	0.0024	-0.0021	0.0011



(a) (b)

FIGURE 5 (a) Performance plot (b) Regression plot

From the above results, the multiple factors forecast was not good as single factor forecast because the former only use one year data and the learning information was less.

3.3.3 Multiple factors and single colour predicting

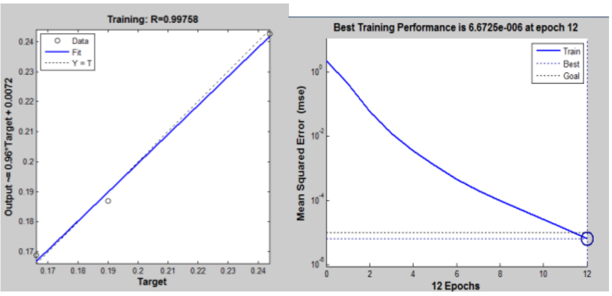
In order to improve the forecast performance, the above two situations were combined. So in this mode, ten colours of the past three years were considered to predict the fourth colour trend.

(1) Different learning rate

In this case, $Nsize$ =8, train function='traingdx', learn function='learngdm'.

TABLE 7 Multiple factors forecast of the hue yellow (η =0.05)

Year / Value	2010	2011	2012
Actual value	0.2440	0.190	0.1660
Forecast value	0.2427	0.1868	0.1688
Error	0.0013	0.0032	-0.0028

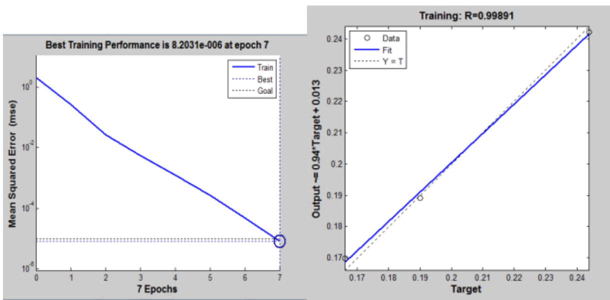


(a) (b)

FIGURE 6 (a) Performance plot; (b) Regression plot

TABLE 8 Multiple factors forecast of the hue yellow (η =0.1)

Year / Value	2010	2011	2012
Actual value	0.2440	0.1900	0.1660
Forecast value	0.2435	0.1886	0.1623
Error	0.0005	0.0014	0.0037



(a) (b)

FIGURE 7 (a) Performance plot; (b) Regression plot

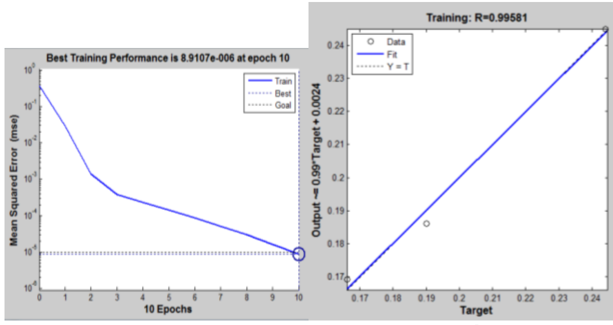
The experimental results of different learning rates are shown in Table 7, Table 8 and Figure 6, Figure 7. Bigger learning rates may make the network weight change too much each time, or even lead to weights jump without convergence during the training process; on the other hand, smaller learning rates lead to learning time too long and can't guarantee convergence to a minimum. In this example, η =0.1 has better results.

TABLE 9 Multiple factors forecast of the hue yellow (train function='traingdx', learning function='learngd')

Year / Value	2010	2011	2012
Actual value	0.2440	0.1900	0.1660
Forecast value	0.2448	0.1861	0.1693
Error	-0.0008	0.0039	-0.0033

(2) Different training function and learning function

In this experiment, η =0.1, $Nsize$ =8.

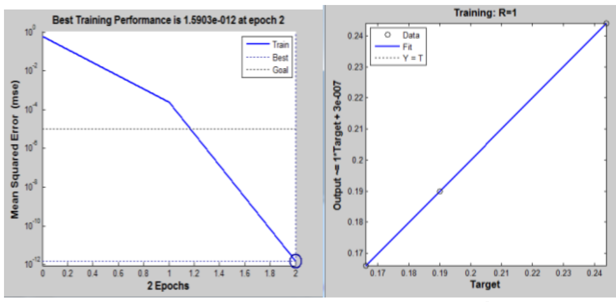


(a) (b)

FIGURE 8 (a) Performance plot; (b) Regression plot

TABLE 10 Multiple factors forecast of the hue yellow (training function='trainlm', learning function='learngdm')

Year / Value	2010	2011	2012
Actual value	0.2440	0.1900	0.1660
Forecast value	0.2440	0.1900	0.1660
Error	0	0	0



(a) (b)

FIGURE 9 (a) Performance plot; (b) Regression plot

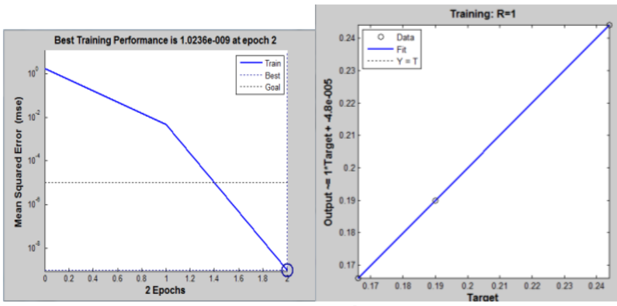
Form the results in Table 9, Table 10 and Figure 8, Figure 9, samples were perfectly fit and the error was zero. TRAINLM training function uses Levenberg – Marquardt algorithm and has the fastest convergence speed, but requires more storage space. Learngdm is a Gradient descent function with momentum back propagation and considers the learning experience.

(3) Different neurons in hidden layer

η =0.1, training function='trainlm', learning function= 'learngdm'. $Nsize$ has different value.

TABLE 11 Multiple factors forecast of the hue yellow ($Nsize$ =7)

Year / Value	2010	2011	2012
Actual value	0.2440	0.1900	0.1660
Forecast value	0.2440	0.1900	0.1660
Error	0	0	0

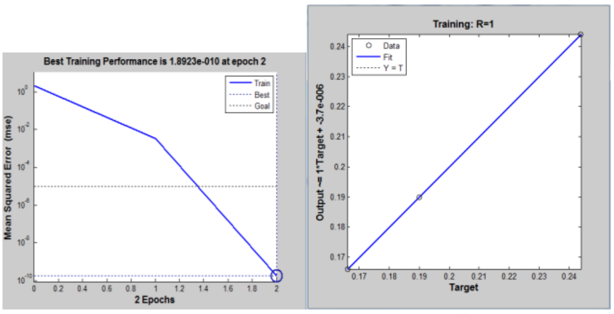


(a) (b)

FIGURE 10 (a) Performance plot; (b) Regression plot (*Nsize* =7)

TABLE 12 Multiple factors forecast of the hue yellow (*Nsize* =6)

Year / Value	2010	2011	2012
Actual value	0.2440	0.1900	0.1660
Forecast value	0.2440	0.1900	0.1660
Error	0	0	0

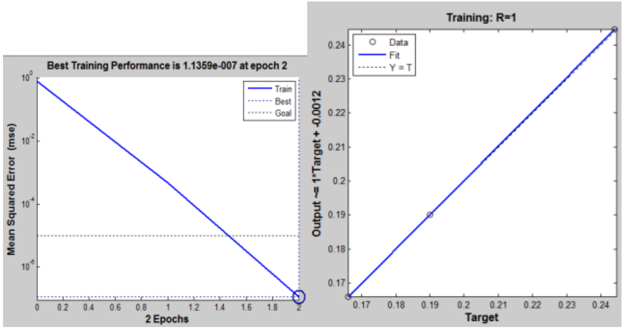


(a) (b)

FIGURE 11 (a) Performance plot; (b) Regression plot (*Nsize* =6)

TABLE 13 Multiple factors forecast of the hue yellow (*Nsize* =2)

Year / Value	2010	2011	2012
Actual value	0.2440	0.1900	0.1660
Forecast value	0.2440	0.1900	0.1660
Error	0	0	0



(a) (b)

FIGURE 12 (a) Performance plot; (b) Regression plot (*Nsize* =2)

The above results under different hidden layers are given in Table 11, Table 12, Table 13 and Figure 10, Figure 11, Figure 12. The results told that the number of hidden layer neurons has less effect to forecast performance. The main reason is that less size of input and output of samples in this example and less neuron can give better results.

4 Conclusions

The above results have demonstrated that the BPNN algorithm can forecast the fashion colour very close. When selecting a different training and learning functions, and other parameters, it can adjust the forecasting performance. In this example, the appropriate samples and network parameters, can achieve zero error prediction and it confirmed the superiority of the neural network in fashion colour forecasting. In the same time, some parameters have little influence on forecasting performance. Especially, multi factors model of the colour data of past three years has considered much information and done better work.

Acknowledgments

The work is supported by Scientific and Technological research projects of Henan Province (102102210190) and Foundation of Henan Province Educational Committee (2011A120005).

References

[1] Diane T, Cassidy T 2005 *Colour Forecasting* Oxford: Blackwell Science 6-23

[2] Diane T C 2007 Personal Colour Analysis, Consumer Colour Preferences and Colour Forecasting for the Fashion and Textile Industries *J Colour Design & Creativity* **1**(1) 1-14

[3] Nenni M E, Giustiniano L, Pirolo L 2013 Demand forecasting in the fashion industry: a review *J Engineering Business Management* ahead of print December 2013 DOI: 10.5772/56840

[4] Na Liu, Shuyun Ren, Tsan-Ming Choi, Chi-Leung Hui, Sau-Fun Ng 2013 Sales Forecasting for Fashion Retailing Service Industry: A Review *J Mathematical Problems in Engineering* ahead of print DOI:10.1155/2013/738675

[5] Lin J J, Sun P T, Chen J J, et al. 2010 Applying Gray Model to Predicting Trend of Textile Fashion Colours *J the Textile Institute* **101**(4) 360-8

[6] Chang L X, Gao W D, Lu Y Z, et al. 2010 Analysis on Spring/summer Fashion Colours *J Textile* **31**(3) 98-103

[7] Stanstield J, Whitfield T W 2005 Can Future Colour Trends be Predicted on the Basis of Past Colour Trends?: An Empirical Investigation *J Colour Research & Application* **30**(3) 235-42

[8] Chang L X, Gao W D, Pan R R 2011 Applying grey model for international fashion colour trend forecasting *J Grey System* **14**(4) 159-64

[9] Kenji Suzuki 2011 *Artificial Neural Networks - Industrial and Control Engineering Applications* InTech

[10] Yu Y, Hui C L, Choi T M 2012 An Empirical Study of Intelligent Expert Systems on Forecasting of Fashion Colour Trend *J Expert Systems with Applications* **39** 4383-9

[11] Chang L X, Pan R R, Gao W D 2013 Fashion colour forecasting by applying an improved BP neural network *J Donghua University* **30**(1) 58-62

[12] Bhambure S D, Dhavale A J, Kadole P V, et al. 2013 Artificial Neural Network & its Applications in Textiles *Journal of the textile association* 31-7

[13] Wu Y Z, Zhai Y C, Sun L 2011 Fashion Colours Forecast Based on Gray Back-propagation Neural Network Model *J Donghua University: Natural science* **37**(4) 199-204
[14] Hui C, Ng S, Yu Y 2012 Colour Trend Forecasting of Fashionable Products with Very Few Historical Data *J IEEE Transactions on Systems, Man and Cybernetics, Part C: Application and Review* **42**(6) 1003-10
[15] Adya M, Collopy F 1998 How effective are neural networks at forecasting and prediction? A review and evaluation *J Forecasting* **17** 481–95
[16] Rojas R 1996 *Neutral Networks: A Systematic Introduction* Springer
[17] Xiaoshu S, Hong Z, Xuemin H 2012 Fabric Defect Detection Based on Regional Growing PCNN *J Multimedia* **7**(5) 372-9
[18] Habib M, Rokonuzzaman M 2013 An Empirical Approach to Optimize Design of Back propagation Neural Network Classifier for Textile Defect Inspection *J Mathematics & Computer Science* **3**(4) 617-34
[19] Wu Y Z, Sun L, Le J J 2010 G-LMBPNN: A New Fashion Colours Prediction Model In: *2010 International Conference on Computational Aspects of Social Networks*, Taiyuan, China, 501-4
[20] Häusler S 2010 *Neural Network Toolbox*

Authors

Xiaohong Kong, born on December 14, 1972, China

Current position, grades: researcher at Henna Institute of Science and Technology, China.
University studies: Ph.D. degree in Information Technology and Engineering from Jiangnan University, China in 2007.
Scientific interests: Automatic Control and Intelligent Algorithm.

Lixia Chang, born on January 22, 1972, China

Current position, grades: researcher at Henna Institute of Science and Technology, China.
University studies: Ph.D. degree in Textile engineering from Jiangnan University, China in 2014.
Scientific interests: apparel technology and culture.

Junpeng Xu, born on March 19, 1979, China

Current position, grades: Teacher at Henan Institute of Science and Technology, China.
University studies: master's degree in Fundamental Mathematics from Henan University, China in 2011.
Scientific interests: information technology and Automatic Control.