Chapter 8
The Color Characteristic Model Based on Optimized BP Neural Network

Hongtao Miao and Lizheng Zhang

Abstract Due to BP neural network's own limitations, it requires a large number of samples for complex prediction, and its generalization ability is weak. This paper puts forward an optimized BP neural network algorithm. The methods focus on data normalization in order to improve maximum speed limit, the inertia constant, and fitness function to eventually optimize BP neural network weight value and threshold value to reduce its distribution range before using BP neural network for color prediction. Reducing the possibility of BP neural network prediction model getting into local optimization has good nonlinear fitting capability and higher prediction accuracy of color space conversion. If using in the ICC profile, it will ensure the accuracy of the color conversion.

Keywords PSO · BP neural network · Color management · ICC profile

8.1 Introduction

At present, the color management system CMS uses ICC standard to manage the color space of different equipment in workflow process. The color management system is composed of Profile Connection Space (PCS), the ICC profile, and color management module (CMM). Among them, the color management module (CMM) is the core. Color space conversion requires a large amount of accurate data. These data are

H. Miao (✉)
Packaging and Printing Department, Henan University of Animal Husbandry and Economy,
Zhengzhou, China
e-mail: 774609787@qq.com

L. Zhang
School of Printing and Packaging Engineering, Beijing Institute of Graphic Communication,
Beijing, China
e-mail: zhanglizheng@bigc.edu.cn

© Springer Science+Business Media Singapore 2016

Y. Ouyang et al. (eds.), *Advanced Graphic Communications, Packaging Technology and Materials*, Lecture Notes in Electrical Engineering 369,
DOI 10.1007/978-981-10-0072-0_8

collected in a lot of color space model experiments, which use common printing materials and process under different production conditions.

But at present, color-matching accuracy of main color management software cannot meet the requirements of industrial production. So in the color-matching algorithm, a new matching algorithm is needed to make up for the ICC profile. The PSO-BP neural network algorithm can combine the advantages of two algorithms so that color management can meet the practical application requirements completely (see [1, 2]).

8.2 BP Neural Network Algorithms

The neural network algorithm is inspired by the function of human brain and neural system. If color space is considered as a nonlinear control system, it can be used to simulate the system.

As show in Fig. 8.1, this is a layer network system with three inputs and three outputs. Each circle represents a neuron, a vector arrow represents a neural key, and its added value is $W_{ij,}$ where i is the number of nerves which receive the input, and j is the number of the output nerve. So the output of a_i is as follows:

$$a_i = f\left(\sum w_{ij} a_j\right) \tag{8.1}$$

Summation range is all output of a_i. f is nonlinear function.

In order to train this network, some training data are required (x_k, y_k), which means that if the input is x_k, then output of the network should be y_k, and if the output of the network is z_k, then the error is as follows:

$$E = \sum \|y_k - z_k\|^2 \tag{8.2}$$

The sum of its summation is all training data. At this time, the error can determine the direction and intensity of the value adjustment:

$$\Delta w_{ij} = g\left(E, \Delta w_{ij}\right) \tag{8.3}$$

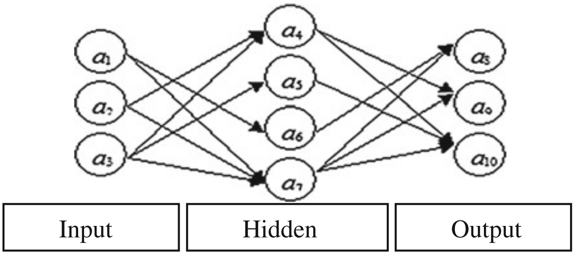

Fig. 8.1 Three layer networks

The accuracy of the neural network method depends on the structure, training method, and training data of the network. Compared with other methods, the advantages of neural network are as follow:

(1) A valuable memory space is saved, if it is used in hardware environment and the speed is not a problem.
(2) If the network structure and training method are reasonable and efficient, its precision can also meet the application requirements [3].

However, the disadvantages of applying neural network to complete color space conversion are long training time, which cannot be calculated in minutes or hours, and it will take a few days. The research found that using the PSO algorithm to optimize the neural network to complete the color space conversion can effectively improve the shortcomings of neural network algorithm. As a simple and effective random search algorithm, PSO can also be used to optimize the neural network [4]. Although the research of this aspect is still in the initial stage, the research results show that PSO has great potential in optimizing neural network.

8.3 PSO Algorithms

PSO is a very simple algorithm and can effectively optimize the various functions. To some extent, this algorithm is between genetic algorithm and evolutionary programming. The algorithm is very dependent on the random process, which is also its similarity with evolutionary programming. In this algorithm, global optimal and local optimal adjustment is very similar in crossover operator of genetic algorithm. This algorithm is also used to the concept of fitness, which is the common feature of all evolutionary computation methods [5, 6].

In the PSO algorithm, the m dimension (the dimension of each particle) is searched by a group of N particles; each particle corresponds to a potential solution to an optimization problem. The position and speed of i are expressed as $x_i = (x_{i1}, x_{i2},\ldots,x_{im})$ and $\upsilon_i = (\upsilon_{i1}, \upsilon_{i2},\ldots,\upsilon_{im})$. The target function value of x_i is called the fitness value off_i, and the algorithm can measure the particle's quality; υ_i determines the direction and distance of particle movement. Each particle is updated by tracing two optimal solutions: p_i is the historical optimal solution of particle search, and p_g is the optimal solution for the whole group. Particle velocity and position renewal formula are as follows:

$$\upsilon_{id}^{k+1} = \omega\upsilon_{id}^{k} + c_1\xi\left(p_{id}^{k} - x_{id}^{k}\right) + c_2\eta\left(p_{gd}^{k} - x_{id}^{k}\right) \tag{8.4}$$

$$x_{id}^{k+1} = x_{id}^{k} + \upsilon_{id}^{k+1} \tag{8.5}$$

Among them, c_1 and c_2 are the learning factors, it usually take '2'. ω is the inertia value, ξ and η are the number of random between (0, 1), and each one-dimensional speed is limited within $\upsilon_{\max}$ [7].

8.4 Improvement of PSO Algorithm

8.4.1 Dynamically Adjust the Learning Factor Strategy

In the PSO algorithm, the expression of the learning factors c_1 and c_2 is usually 2. Through the experiment, the learning factor of the time change has a great influence on the performance of the algorithm, and the expression is as follows:

$$c_1 = (c_{1f} - c_{1i}) * \text{iter}/\text{MAXITER} + c_{1i} \tag{8.6}$$

$$c_2 = (c_{2f} - c_{2i}) * \text{iter}/\text{MAXITER} + c_{2i} \tag{8.7}$$

Among them, c_{1f} and c_{1i} are the maximum and minimum values of c_1, c_{2f} and c_{2i} are the maximum and minimum values of c_2. iter is the number of iterations of the algorithm, and MAXITER is the maximum number of iterations that the algorithm allows [8].

8.4.2 Inertia Weight Factor Change

In the PSO, inertia weight factor is the most important adjustable parameter. It was found that dynamic w can obtain the stable and better results of the fixed value w.

Inertia weight expressions are as follows:

$$\omega = (\omega_1 - \omega_2) * (\text{MAXITER} - \text{iter})/\text{MAXITER} + \omega_2 \tag{8.8}$$

Among them, ω_1 and ω_2 are the maximum and the minimum of the inertia weight, iter is the number of the current iteration, and MAXITER is the maximum iteration number of the algorithm. By study, if inertia weight in the interval [0.4, 0.95], with the iteration number of linear decline, the problem has better effect [9].

8.4.3 Maximum Speed Setting

PSO algorithm has the characteristics of losing the particle diversity in the late stage of flight. It is best to maintain the particle diversity without increasing the particle number and particle dimension. Because of changing the particle diversity by increasing the number of particles or increasing the particle dimension, the number of arithmetic operations of the algorithm will increase exponentially, which will seriously affect the effectiveness of the algorithm. In order to fully and effectively maintain diversity of particle, the particle swarm can be divided into two parts. Without an increase in particle number and particle dimension, two PSO are each used to global search and local optimization. For global search, when it is iterated, the particle velocity will be reinitialized once; with the increase of the number of iterations, it will search space and search the global optimal value, while the probability of the neighborhood value will be significantly increased. For local search, it will search the optimal value rapidly and accurately nearby the historical optimal value, which is searched by global search [10]. In this way, the different particle swarm has to do its job, and the global search and the local optimization are carried out simultaneously. In particular, the position of global search particle swarm is updated as formula (8.5), and the speed update formula is as follows:

$$v_{id}^{k+1} = \text{rand}(0, v_{\max}) \tag{8.9}$$

Global search particles emphasize the particle diversity, in order to avoid the particle search space; the particle speed of the update is the random number limited between (0, $v_{\max}$). The location update is related to the last position of the particle and the current speed [11].

The formula of the position of local optimization particle swarm's single-step update is shown in formula (8.5), and the formula of the speed of local optimization particle swarm's single-step update is as follows:

$$v_{id}^{k+1} = \omega v_{id}^{k} + c_1 \xi (p_{id}^{k} - x_{id}^{k}) + c_2 \eta (p_{gd}^{k} - x_{id}^{k}) + c_3 \zeta (p_{sd}^{k} - x_{id}^{k}) \tag{8.10}$$

Among them, c_3 is the learning factor, ζ is the random number between (0, 1), and p_s is the global search PSO solution of the global history.

The particle swarm of global search transfers the historical optimal individual information to the particle swarm of local optimization particle swarm; the particle swarm will optimize by the formulas (8.5) and (8.10). The learning factor c_3 determines the intensity of the evolutionary information obtained from the global search PSO.

In type (8.8), the choice of ω, c_1, $c_{2,}$ and c_3 has a greater impact on the performance of the algorithm. According to previous test experience, inertia weight is set in the range of [0.4, 0.95], c_1 is set in the range of [0.5, 2.5], c_2 is set in the range of [0.75, 2.5], and the optimum ranges of c_3 are in the range of [0.75, 2.5].

8.4.4 Improved Fitness Function

In order to overcome the premature phenomenon of PSO, the algorithm of PSO is improved in order to improve the search performance of the PSO. The specific transformation process is described below. Firstly, the error formula of the network is determined, "$E_k = T_k - T_{1k}$ (T_k is the measurement value of the network, T_{1k} is the predictive value of the network)." And then, the average color-difference formula of all samples is determined.

$$E_1 = \frac{1}{K}\sum_{K=1}^{N}\sqrt{E_{(1,K)}^2 + E_{(2,K)}^2 + E_{(3,K)}^2} \tag{8.11}$$

(N is the number of samples). Finally, in the programming software determined the fitness function is ‘fitness = exp (1/E1)’.

8.5 Particle Swarm Optimization BP Neural Network

8.5.1 Optimization Process

See Fig. 8.2

8.5.2 Data Normalization

Before training the network, the statistical distribution of the uniform sample is needed. Why use the normalized? First, you must know a concept called singular

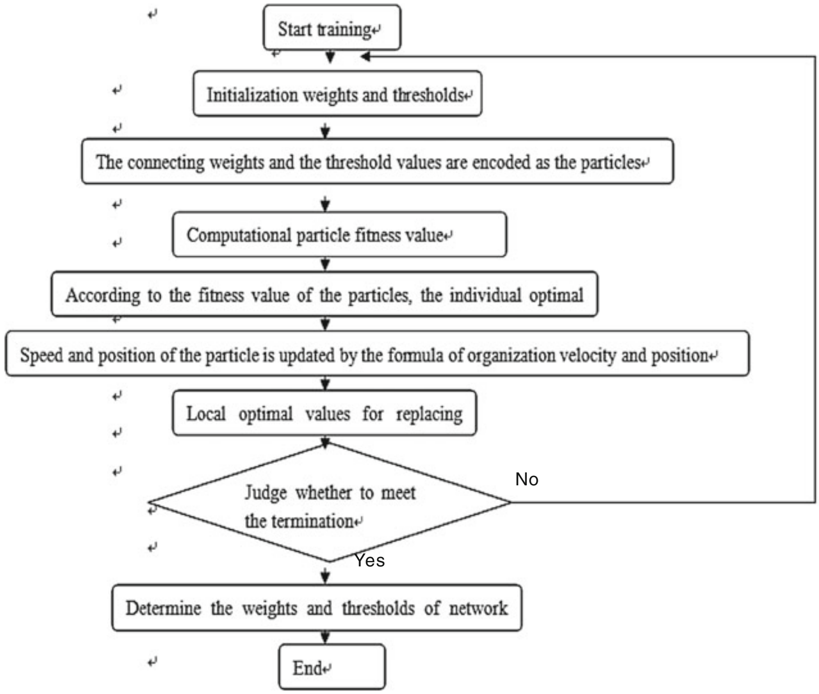
Start training
Initialization weights and thresholds
The connecting weights and the threshold values are encoded as the particles
Computational particle fitness value
According to the fitness value of the particles, the individual optimal
Speed and position of the particle is updated by the formula of organization velocity and position
Local optimal values for replacing
Judge whether to meet the termination
No
Yes
Determine the weights and thresholds of network
End

Fig. 8.2 Optimization process

sample data. The singular sample data are particularly large or small sample relative to other samples.

The following example:

$$m = [0.11 \quad 0.16 \quad 0.32 \quad 0.35 \quad 40; \quad 0.24 \quad 0.27 \quad 0.25 \quad 45];$$

One of the fifth columns of data relative to the other 4 columns of data can be called singular sample data. Because of the increase of network training time caused by singular sample data, it may cause the network to be unable to converge, so the best advanced shape normalized, and if there is no singular sample data, it does not need to be normalized. [12] Premnmx function is used to normalize the input data (RGB) and output data (Lab) of the network, and the normalized data will be distributed within the range of [−1, 1].

The correlation experiment proves that the average color difference fluctuation range obtained by using data normalization is small and stable, so that the error of selection is small.

8.6 Particle Swarm Update Iterative Step Method

Through the particle dimension and neural network connection weights of the mapping relationship and the particle dimension of the search space settings, BP neural network is established, to calculate the fitness value, to update speed, and to calculate the optimal particle (i.e., optimal weights and thresholds) assigned to the neural network.

8.6.1 Simulation Experiment Principle

In order to verify the accuracy of this algorithm, the first step is to establish a standard color chart, print out the standard color chart, and measure the Lab values of every color. The Lab value of the measurement is given as the training data to optimize the BP neural network system after the PSO, so as to carry out the corresponding calculation.

Then, set up a test chart, print out the test chart, and measure the Lab values of each color, as a standard value. After the PSO, the BP neural network system is calculated, and the Lab value of the test color chart is calculated as the reference value. Then, calculate the color difference between the two.

8.6.2 Experimental Instruments

Printer: EPSON9800; color measuring instrument: X-Rite Eye-One Pro divided into light photometer; correction and characteristics of software for i1Profiler; operation platform for the Windows XP system.

8.6.3 Experiment Specific Method

Standard color chart is identified: the Lab from 0 to 100 at intervals of 20–6 level segmentation. Lab was taken from one of the values of a color. Thus, there are 216 colors. After printing, measure the color standard using the software ‘Eye-One’, and record ‘Lab’ value of the 216 colors and store it in a text file. These values as training data will impart BP neural network system, so as to train network.

The test target of identified: the lab from 0 to 100 at intervals of 10–11 level segmentation, lab were taken one of the values of a color, so a total of 1331 colors. On the one hand the color standard value will input training network where the calculated values should appear; on the other hand, after printing use ‘Eye-One’ measure the color standard, record test color values, and store it in a text file.

In order to facilitate comparison, in this paper program algorithm is prepared under the MATLAB environment to construct two kinds of forecasting model, BP neural network prediction model (BP model) and the particle swarm algorithm-optimized BP neural network prediction model (PSO-BP model). In the experiment, the minimum average color difference E was used as the evaluation criterion, and the color difference of the color was calculated by comparing the color difference of the color to the uniform color space of CIE1976L*a*b*:

$$\Delta E_{ab}^{*} = \sqrt{\Delta L^2 + \Delta a^2 + \Delta b^2} \tag{8.12}$$

Analysis in Table 8.1 shows, under the same network structure, transfer function, training function and expected color error, which are the conditions, improved PSO-BP network model accuracy (average value of 1.37) was higher than that of BP neural network (ΔE averaged 3.02), indicating that PSO-BP network model of color space conversion nonlinear quasi-ability is stronger than the BP neural network, and PSO algorithm can indeed affect optimization of initial weights of BP neural network and the threshold.

Table 8.1 Forecast effect data of BP and PSO-BP network

Model types	Maximum ΔE	Minimum ΔE	Averaged ΔE
BP	14.1	1.84	3.02
PSO-BP	3.85	1.11	1.37

8.7 Conclusions

It is important to find the network weights and threshold in BP neural network algorithm, mainly through improved maximum speed limit, setting dynamic inertia constant and fitness function to optimize the weights and thresholds of BP neural network, narrow the distribution range, and then use the BP neural network algorithm for color prediction. Experiments show that the proposed method greatly reduces the possibility of local minimum of BP neural network and improves the convergence rate of the model. And the accuracy of data prediction is improved obviously, which is of great help to color-matching in color feature file, and further guarantees the accuracy of color transfer.

References

1. Zhang, Y., Wu, Q., & Ma, Z. (2010). A survey of particle swarm optimization and its application. *Micro Computer Information, 11*(3), 72–77.
2. Zhou, J., Chen, J., & Liu, G. (2013). An overview of inertia weight in particle swarm optimization. *Guangdong Power, 32*(3), 55–56.
3. Grosan, C., & Abraham, A. (2007). Hybrid evolutionary algorithms, methodologies, architectures, and reviews. *Studies in Computational Intelligence (SCI), 75*, 1–17.
4. Garnier, S., Gautrais, J., & Theraulaz, G. (2007). The biological principles of swarm intelligence. *Swarm Intelligence, 1*, 3–31.
5. Eberhart, R., & Shi, Y. (2001). Particle swarm optimization: Developments, applications and resources. In *Proceedings of IEEE Congress Evolutionary Computation* (Vol. 1, pp. 81–86)
6. Parsopoulos, K., & Vrahatis, M. (2002). Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing, 1*, 235–306.
7. Banks, A., Vincent, J., & Anyakoha, C. (2007). A review of particle swarm optimization. Part I: background and development. *Natural Computing, 6*, 467–484.
8. Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization: An overview. *Swarm Intelligence, 1*(1), 33–57.
9. van Ast, J., Babuška, R., De Schutter, B. (2008). Particle swarms in optimization and control. In *Proceedings of the 17th World Congress The International Federation of Automatic Control* (pp. 5131–5136). Seoul, Korea, July 6–11
10. Li, R-j. (2003). Study on color space conversion model of from RGB to CIEXYZ. *Packaging Engineering, 30*(3), 79–81.
11. Hong, L., & Chu, G-l. (2013). RGB to CIEXYZ color space conversion. *Packaging Engineering, 34*(21), 85–89.
12. del Valle, Y., Venayagamoorthy, G., Mohagheghi, S., et al. (2008). Particle swarm optimization: Basic concepts, variants and applications in power systems. *IEEE Transactions on Evolutionary Computation, 12*(2), 171–195.