International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.9, No.4 (2016), pp.323-330
http://dx.doi.org/10.14257/ijsip.2016.9.4.28

Facial Image Recognition Algorithm Based on BP Neural Network

Peihua Su

Engineering College of Xi'an International University, 710077, Xi'an, Shaanxi, China
doudousu1205@163.com

Abstract

The efficiency, quality and accuracy of facial image recognition are restricted by luminance, posture, image quality, massive data and method of image recognition, etc. In response to this, this thesis proposes a facial image recognition algorithm based on BP neural network. It improves on traditional BP neutral network by constructing neutrons of facial image recognition in the input layer, hidden layer and output layer. And by constructing the network framework structure of facial image recognition, it also constructs design elements of facial image recognition from input code and output code and therefore constructs the facial image recognition algorithm based on BP neural network. This thesis verifies the algorithm through practical cases and proves that the algorithm is effective and operable.

***Keywords**: Facial image recognition; BP neural network; Pattern recognition; Multi-feature; Algorithm*

1. Introduction

Facial image recognition is a hotspot in the study of image processing and recognition as well as an integrated part of artificial intelligence design [1-2]. Facial image recognition is to obtain users' facial image through video collecting tools and analyze the position of facial features, face contour and direction of face through the algorithm so as to compare these features with samples in the database and figure out the identity of the user. Facial image recognition is usually based on single training sample of local characteristic area. However, as the efficiency, quality and accuracy of facial image recognition are restricted by luminance, posture, image quality, massive data and method of image recognition, such technology is still unable to realize high quality, high accuracy and high efficiency.

At present, some scholars and experts have made extensive study on facial image recognition with fruitful results. But there is room for matching images quickly and accurately. Artificial neural network (ANN) is a mathematical model for biomimetic neural network [6-8]. It can effectively deal with tangible and intangible information during mode recognition through sample training, in particular dealing with the multilayer feed-forward network (or BP (Back Propagation) neural network [9-12] through error back propagation training. The multilayer feed-forward network is widely applied in engineering and has provided new solution to facial image recognition featured by massive data, multiple factors and multi-features.

2. Basic Concepts of BP Neural Network

2.1. Neural Model

Artificial neural network (ANN) is a mathematical model for biomimetic neural network. The neural network is connected by a large quantity of nodes (artificial neurons). Each node represents a specified input function, which is called activation function. The

ISSN: 2005-4254 IJSIP

connection of every two nodes represents a weighted value that passes through the connection signal, which is called the weight. Network input results in different output due to way of connection, weight and activation function. Neurons are the fundaments of the artificial neural network (ANN). With a real number vector $A=(a1,a2,\cdots,an)$ as input, and $W=(w1,w2,\cdots,wn)$ as neuron weight vector, b is bias current, and F is transfer function, so the neuron output t is:

$$t=F*(W*A+b) \tag{1}$$

Correspondingly, the neural model is constructed as in Figure 1:

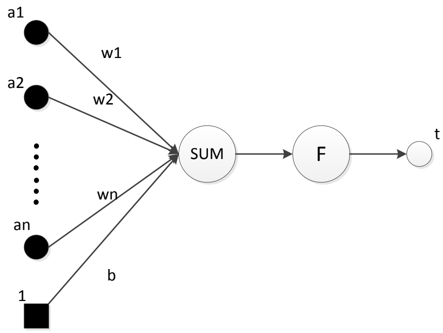


Figure 1. Neural Model

2.2. BP Network Model

BP neural network is so far one of the widest applied neural network models. BP network is able to study and store input-output reflection without the necessity to know about the mathematical equation that describes the reflection. Its learning rule follows gradient descent which adjusts weight and threshold through back propagation to minimize the sum of error squares. The topological structure of BP neural network includes the input layer, hidden layer and output layer, as shown in Figure 2. Neurons in the input layer are responsible for receiving input information and transmitting them to neurons in the hidden layer. The hidden layer is for information processing in which neurons are responsible for information exchange. According to the demand of information change ability, the hidden layer can be designed as single hidden layer or multi-hidden layer. The last hidden layer passes information to neurons in the output layer where the learning is completed through forward-propagation. The output layer outputs the processing results. If the real output differs from the expectation, back propagation of error will be conducted.

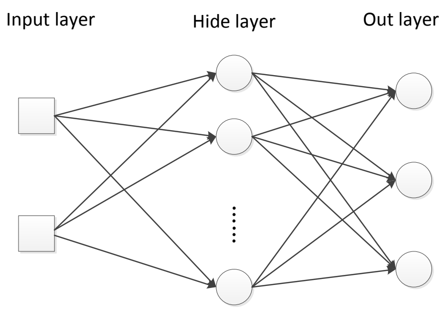


Figure 2. BP Neural Network Model

3. The Facial Image Recognition Algorithm Based on BP Neural Network

The learning task of facial image recognition is to categorize images of different postures, such as judging whether a man is happy, sad, angry or cold, judge in which direction he faces to, or judge whether he wears glasses. These features will become the objective function of learning whose input is a facial image and output is the direction he faces.

3.1. Input Code of Facial Image Recognition

The input of artificial neural network (ANN) is described by images. This thesis decodes images to 30×32 pixel luminance. Each pixel corresponds to a network input. Luminance in the range of 0-255 is reduced to 0-1 by scale so that the value of the input unit and the output unit fall in the same range. In fact, the 30×32 pixel image is exactly the 128×120 pixel image, only with lower resolution. Each image of low resolution is obtained according to the average of several pixel of luminance of high resolution. With image of low resolution, the number of input and weight is easier to deal with, which reduces the burden on calculation but remains sufficient solution to categorize images. Figure 3 is an input image of 30×32 resolution.

Figure 3. Image Input Under 30×32 Resolution

3.2. Output Code of Facial Image Recognition

In this thesis, the output code is 1-of-n code. As human face has four directions, four different output units are set up each representing one direction. 1-of-n output code is adopted because of two reasons: one is that the network provides greater flexibility to the objective function (namely in the output layer there are n times of weight that is available), the other is that for 1-of-n code, differences between the maximum output and the second maximum output can serve as forecast confidence of network, considering that unclear categorization may lead to close calculation results. This thesis uses 0.9 and 0.1, rather than 1 and 0 to represent the objective value of each output unit, in part Sigmoid unit cannot produce 0 and 1 given limited weight. For any attempt to the match objective value 0 and 1 through training network, the gradient descent will result in infinite growth of weight.

3.3. Network Structure of Facial Image Recognition

It is very common to use one-layer or two-layer sigmoid unit in BP neural network, while sometimes three-layer sigmoid unit may also be applied. The three-layer sigmoid unit can represent large objective function, but it is seldom used because it prolongs the training time. As a result, this thesis adopts two-layer sigmoid unit, namely a hidden layer plus an output layer.

3.4. Facial Image Recognition Algorithm Based on BP and Operation Steps

Facial image recognition is a hotspot in computer science. It belongs to biological feature recognition technology which distinguishes individuals from biological features.

The BP neural network model includes the input-output model, effort function model, error calculation model and self-study model.

If f is non-linear function, q is threshold of neural unit, the intangible node output model is expressed as:

$$O_j = f\left(\sum w_{ij} * x_i - q_j\right) \tag{2}$$

The node output model is:

$$Y_k = f\left(\sum t_{jk} * O_j - q_k\right) \tag{3}$$

The stimulus pulse strength of upper layer node to lower layer input is usually a sigmoid function with consecutive values between 0 and 1.

$$f(x) = 1/\left(1+e^{-x}\right) \tag{4}$$

Therefore, if tpi is the expected output value of node i, Opi is the calculated value, the error between the expected value and the calculated value can be modeled as:

$$Ep = \frac{1}{2} * \sum (tpi - Opi)^2 \tag{5}$$

Correspondingly, the self-learning model is:

$$\Delta w_{ij}(n+1) = h * \Phi_i * O_j + \alpha * \Delta w_{ij}(n) \tag{6}$$

Where h is learning factor, Φ_i is calculation error of output node i, O_j is calculation input of output node j and α is dynamic factor.

Based on the above discussion, the BP model is:

$$BACKPROPAGATION\left(training\_examples, \eta, n_{in}, n_{out}, n_{hidden}\right) \tag{7}$$

Where the training sample in $training\_examples$ is an ordered pair in the form of $(\vec{x}, \vec{t})$, $\vec{x}$ is network input vector, $\vec{t}$ is objective output value, η is speed of learning, n_{in} is number of input units, n_{out} is number of output units and n_{hidden} is number of hidden units.

If the input from unit i to unit j is expressed as x_{ji}, the weight from unit i to unit j is expressed as w_{ji}, key steps to realize the algorithm are described as the followings:

Step 1 Create a network with n_{in} input units, n_{hidden} hidden units and n_{out} output units.

Step 2 Initialize all random values which have the minimum network weight;

Step 3 For each sample in training_examples that transmits forward in the network, input case $\vec{x}$ in the network and calculate the output o_u of u in each unit to realize error back propagation.

Step 4 Calculate the error item δ_k for each output unit k in the network:

$$\delta_k = \leftarrow o_k\left(1-o_k\right)\left(t_k - o_k\right) \tag{8}$$

Step 5 Calculate the error item δ_h for each hidden unit h in the network:

$$\delta_h = \leftarrow o_h (1 - o_h) \sum_{k \in outputs} w_{hk} * \delta_k \tag{9}$$

Step 6 Update each network weight w_{ji}:

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij} \tag{10}$$

$$\Delta w_{ij} = \eta * \delta_j * x_{ij} \tag{11}$$

Step 7 Check if the result meets the condition of ending the algorithm, if yes, terminate the steps; if no, repeat Step 2-6.

4. Algorithm Validation and Analysis

This thesis explains the algorithm through the input images in Figure 3 which are processed by Pgmimage.h and pgmimage.c. To be specific, images are read and coded to generate two data structures, namely IMAGE and IMAGELIST. Backprop.h and backprop.c are adopted to support three-layer full connection feed-forward neural network, which realizes the back propagation of gradient descent for the purpose of adjusting the weight. Imagenet.c is used to serve as the 接口 linking documents that carry images and the input unit and it sets up code of objective vector for training. Facetrain.c is used to control the top-layer program of modules. Hidtopgm.c is to shift weight of unit in the hidden layer to image coding for demonstration. Outtopgm.c is to shift weight to image coding for demonstration. At the same time, to visualize the weight learned by the algorithm, the weight of the input unit is initialized as 0, or to say, it follows gradient descent rather than random decrease, in that weight visualization is easier to understand and it has no significant effect on precision. Table 1 and 2 show the experiment settings and parameters respectively.

Table 1. Experiment Settings

CPU:Intel(R) Core(TM)2 Duo CPU 2.93GHz	Storage :2G
OS: Linux 2.6.32-279.22.1.el6.i686	Decoding tool: gcc Red Hat 4.4.6-4

Table 2. Experiment Parameters

Number of input units: 960	Number of hidden units: 3	Number of output units: 4
Speed of learning: 0.3	Impulse: 0.3	Iteration number: 100
Training set: 277	Validation set: 139	Test set: 208

For an artificial neural network (ANN) with 3 hidden units, the precision of test set is 90.3846%, as shown in Figure 3. The error is described in Figure 4.

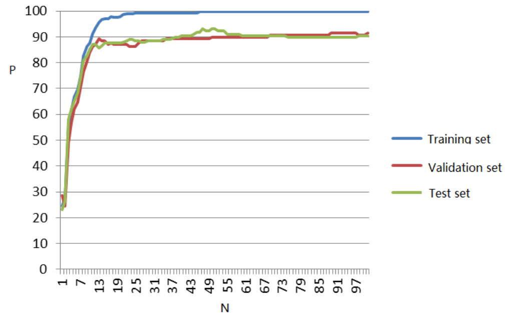


Figure 3. Number of Update for Relative Weight of Precision

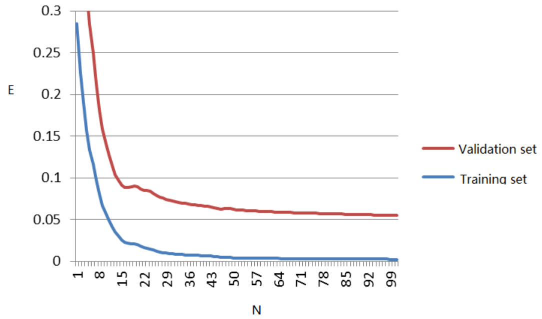


Figure 4. Number of Update for Relative Weight of Error

From Figure 3 it can be seen that the algorithm proposed in this thesis only contains 3 hidden units, but the precision can reach 90%. Thus, a minimum number of hidden unit is needed to study the objective function. And the more the hidden units there are, the more precise the learning is. And cross-checking method is used to decide how many times gradient descent iteration should conduct. Without cross- checking, adding the number of hidden unit usually result in increasing overfitting training data and undermining the precision. Figure 4 shows the change of the number of update for relative weight of error. The error of training set decreases monotonously. But the validation set generates more errors when it updates too many times. This experiment has updated 100 times, and the error of validation set is steady and reaches the minimum in the end. If it updates more than 100 times, there will be more errors occurred.

Table 3 and Figure 5 describe the output unit weight of 960×3×4 network obtained through training and show weight visualization of hidden unit after 100 times of iteration.

Table 3. Output Unit Weight of 100 Times of Iteration

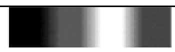
No. of output unit :	Output Unit 1(let)	Output Unit 2(front)	Output Unit 3(right)	Output Unit 4(up)
Weight Image:				
Weight:				

Figure 5. Hidden Unit Image after 100 Times of Iteration

In Table 3, every rectangular describes a weight of one of the four output units in the network. Four small squares in each rectangular represent four weights related to the output unit. The square on the left side has a unit threshold of w0, followed by three weights connecting three hidden units and the output unit. The luminance degree of square represents weight. The lighter, the bigger the positive weight. And the darker, the bigger the negative weight. The grey shade represents middle weight. Figure 5 shows the network weight of gradient descent after 100 times of iteration. It can be seen that the algorithm is sensitive to face and body when deciding weight. So, it can recognize facial image accurately.

5. Conclusions

This thesis studies and analyzes key issues in facial image recognition and proposes a facial image recognition algorithm based on BP neural network. It improves on traditional BP neural network and discusses the input code, output code, network structure and algorithm steps with regard to facial image recognition. Finally, this thesis verifies the algorithm through practical cases and proves that it is precise and efficient with high accuracy and simple calculation.

References

[1] C. Zhao, X. Li and Y. Cang, "Bisecting k-means clustering based face recognition using block-based bag of words model", [J]. Optik - International Journal for Light and Electron Optics, vol. 126, no. 19, **(2015)**, pp. 1761-1766.

[2] L. A. Cament, F. J. Galdames, K.W. Bowyer and C. A. Perez, "Face recognition under pose variation with local Gabor features enhanced by Active Shape and Statistical Models", [J]. Pattern Recognition, vol. 48, no. 11, **(2015)**, pp. 3371-3384.

[3] S.-B. Chen, C. H.Q. Ding and B.Luo, "Extended linear regression for undersampled face recognition", [J]. Journal of Visual Communication and Image Representation, vol. 25, no. 7, **(2014)**, pp. 1800-1809.

[4] J. Lu, J. Zhao and F. Cao, "Extended feed forward neural networks with random weights for face recognition", [J]. Neurocomputing, vol. 136, **(2014)**, pp. 96-102.

[5] S. Li, D. Gong and Y. Yuan, "Face recognition using Weber local descriptors", [J]. Neurocomputing, vol. 122, **(2015)**, pp. 272-283.

[6] R. Kumar, R.K. Aggarwal and J.D. Sharma, "Comparison of regression and artificial neural network models for estimation of global solar radiations", [J]. Renewable and Sustainable Energy Reviews, 2015, 52: 1294-1299.

[7] S. Kang and S. Cho, "Approximating support vector machine with artificial neural network for fast prediction", [J]. Expert Systems with Applications, vol. 41, no.10, **(2014)**, pp. 4989-4995.

[8] A.K. Tsadiras, C.T. Papadopoulos and M.E.J. O'Kelly, "An artificial neural network based decision support system for solving the buffer allocation problem in reliable production lines", [J]. Computers & Industrial Engineering, vol. 66, no. 4, **(2013)**, pp. 1150-1162.

[9] Y. Zhang, X. Gao and S. Katayama, "Weld appearance prediction with BP neural network improved by genetic algorithm during disk laser welding ", [J]. Journal of Manufacturing Systems, 2015, 34: 53-59.

[10] K. Cai, J. Xia, L. Li, Z. Gui. Analysis of the electrical properties of PZT by a BP artificial neural network [J]. Computational Materials Science, vol. 34, no. 2, **(2005)**, pp. 166-172.

[11] J. Liu, H. Chang, T.Y Hsu, X. Zuyao and X. Ruan, "Prediction of the flow stress of high-speed steel during hot deformation using a BP artificial neural network", [J]. Journal of Materials Processing Technology, vol. 103, no. 2, **(2000)**, pp. 200-205.
[12] W. Wu, J.Wang, M. Cheng and Z. Li, "Convergence analysis of online gradient method for BP neural networks", [J]. Convergence analysis of online gradient method for BP neural networks,vol. 24, no. 1, **(2011)**, pp. 91-98.