Research of ID Card Recognition Algorithm Based on Neural Network Pattern Recognition

Wang Naiguo,Zhu Xiangwei*
Suzhou Industrial Park Branch
Jiangsu Union Technical Institute
Suzhou, China
zxwook@163.com *corresponding author

Zhang Jian
School of computer Science & Technology
Soochow University
Suzhou, China
zhangjian2012@suda.edu.cn

Abstract—How to quickly and correctly identify the second generation ID card code has important practical significance in real life. The algorithms have been presented about the recognition and verification code of ID cards. In order to let computer own the knowledge of code of character image, firstly, transform the image of ID cards to binary image ,then segment image character of ID card code and display image character using image processing technique, secondly manual input corresponding correct code of character image, thirdly character image is classification stored in the corresponding folder according to the corresponding correct code. These character images are training samples, which are trained using the matlab toolbox of the neural network pattern to generate training model. Using the generated the training model to identify code of the ID cards. The validation algorithm is based on the generated the training model to identify code of ID cards. Compare the consistent of code of ID cards to identify the code, if recognition accuracy rate is not high, the training model need to be trained again, until the recognition accuracy rate is more than 95%. Neural network pattern recognition and verification algorithm can take advantage of the training model to perfectly realize identification and validation the code of the second-generation ID card, the algorithms have application value in the real work.

***Keywords-** neural network; pattern recognition; image processing; ID card; segmentation*

I. INTRODUCTION

ID is a unique identifier resident status, identity cards are widely used in identity testing department needs the service industry, transportation systems and public security system, it has become essential for the life of the certificates, which plays an important role in the personnel management. At present the registration ID card usually adopts manual fill or manual entry, the work efficiency is low, and the number of erroneous input possibility is big, and the use of computers to automatically identify entry ID can save a lot of manpower and resources, which has broad application prospects.

The algorithm has been proposed using the image processing technology and neural network pattern recognition technology in this paper. It realizes the automatic recognition of the identity card code. The result of experiment illustrates recognition accuracy rate more than 97% if the image ID code is clear. The algorithm fully meets the actual demand of automatic recognition ID code.

II. BP NEURAL NETWORK

In machine learning and cognitive science, artificial neural networks (ANNs) are a family of statistical learning algorithms inspired by biological neural networks and are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Artificial neural networks are generally presented as systems of interconnected "neurons" which can compute values from inputs, and are capable of machine learning as well as pattern recognition thanks to their adaptive nature.

The BPNN (Back Propagation Neural Network) is a special feed-forward network including an input layer, a hidden layer and an output layer as shown in Fig .1.

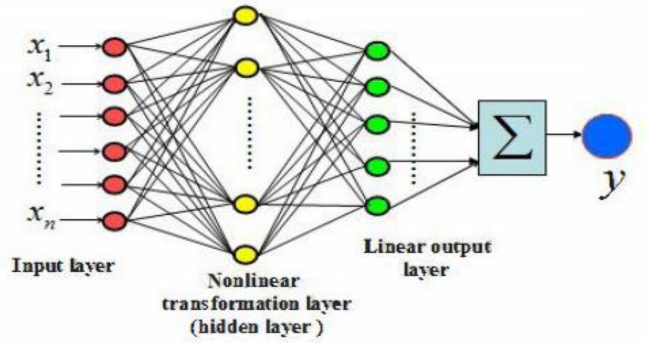


Figure 1. Back Propagation Neural Network architecture

The input layer passes the input signal to the hidden space. There is only one hidden node and its basis function is Gaussian kernel function. It realizes the non-linear conversion from input space to hidden space. The mapping from hidden space to output space is linear and the output of the network is linear weighed sum of the output of hidden unit.

Fig .1 presents part of a typical multilayer network workflow. For more information and other steps, see Multilayer Neural Networks and Back Propagation Training.

When the network weights and biases are initialized, the network is ready for training. The multilayer feed forward network can be trained for function approximation (nonlinear regression) or pattern recognition. The training process requires a set of examples of proper network behavior—network inputs and target outputs.

The process of training a neural network involves tuning the values of the weights and biases of the network to optimize network performance, as defined by the network performance function net(performFc). The default performance function for feed forward networks is mean square error mse(the average squared error) between the network outputs a and the target outputs t. It is defined as following equation (1):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 \qquad (1)$$

There are two different ways in which training can be implemented: incremental mode and batch mode. In incremental mode, the gradient is computed and the weights are updated after each input is applied to the network. In batch mode, all the inputs in the training set are applied to the network before the weights are updated. Incremental training with the adapt command is discussed in Incremental Training with adapt. For most problems, when using the Neural Network Toolbox™ software, batch training is significantly faster and produces smaller errors than incremental training.

For training multilayer feed forward networks, any standard numerical optimization algorithm can be used to optimize the performance function, but there are a few key ones that have shown excellent performance for neural network training. These optimization methods use either the gradient of the network performance with respect to the network weights.

The gradient is calculated using a technique called the back propagation algorithm, which involves performing computations backward through the network. The back propagation computation is derived using the chain rule of calculus.

III. Training Algorithm and Result

A. Sample Collection of Identity Card

In order to generate the training mode, identity card samples should be collected. One of the ID samples is shown in Fig .2.

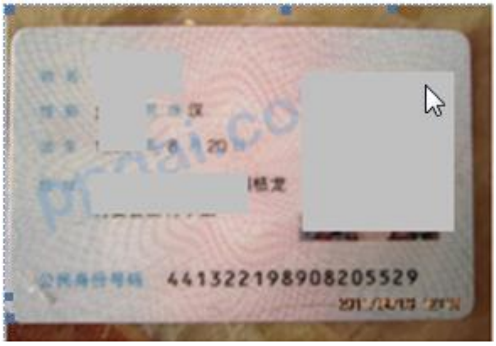
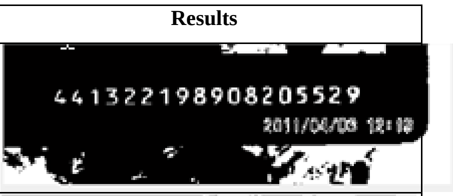
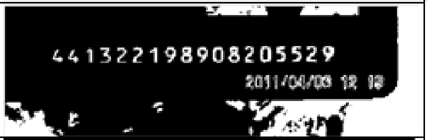
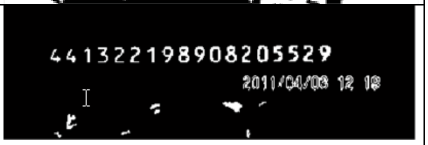
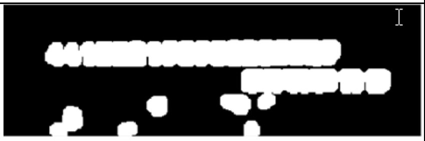
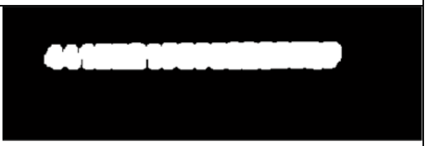
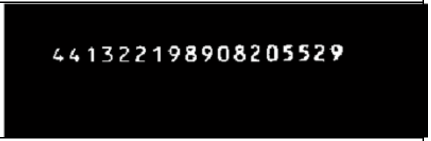
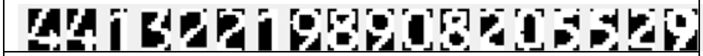
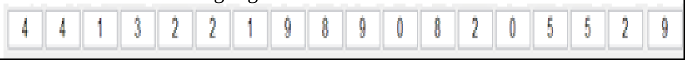
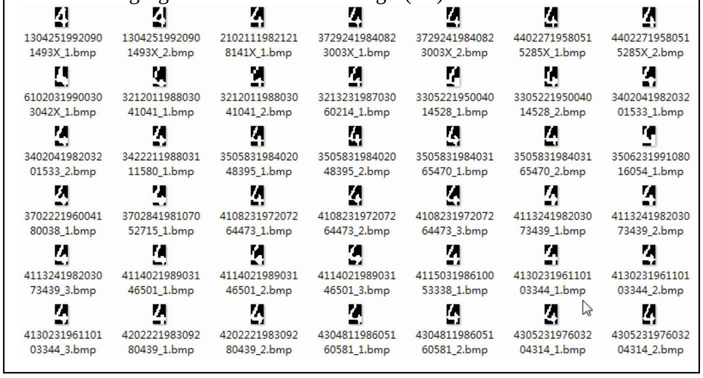


Figure 2. one of the ID card samples

B. Genreate Segmentation Character of ID Code

TABLE I illustrates the steps and results of segmentation character of ID code.

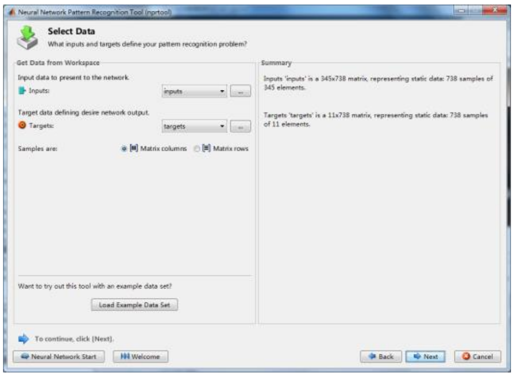
TABLE I. Steps Segmentation Character of ID Code

Steps	Results
1.Part image of ID card is converted to binary image by threshold	
2.All connected components have fewer than 11 pixels should be removed from a binary image.	
3.Clear the image border by suppressing light structures that are lighter than their surroundings and that are connected to the image border.	
4.It is dilated using each structuring element in succession, which is created according to specifies the radius.	
5.Find maximum dilated structuring element that specifies the actual number of pixels in the region.	
6.Under the image mask perform morphological reconstruction using 4-connected neighborhood.	
7.Locate the positioning of the two generation ID card code, which is segmented to the two value characters image. Each character image is shown below in the text box. 	
8.According to code of ID sample, manual input corresponding correct code as shown in the following Figure.  4 4 1 3 2 2 1 9 8 9 0 8 2 0 5 5 2 9	
9.Character images(step 7) are classification stored in the corresponding folder according to the corresponding correct code(step 8). The following figure shows character images( ) which stored in the folder 4  1304251992090 1493X_1.bmp 1304251992090 1493X_2.bmp 2102111982121 8141X_1.bmp 3729241984082 3003X_1.bmp 3729241984082 3003X_2.bmp 4402271958051 5285X_1.bmp 4402271958051 5285X_2.bmp 6102031990030 3042X_1.bmp 3212011988030 41041_1.bmp 3212011988030 41041_2.bmp 3213231987030 60214_1.bmp 3305221950040 14528_1.bmp 3305221950040 14528_2.bmp 3402041982032 01533_1.bmp 3402041982032 01533_2.bmp 3422211988031 11580_1.bmp 3505831984020 48395_1.bmp 3505831984020 48395_2.bmp 3505831984031 65470_1.bmp 3505831984031 65470_2.bmp 3506231991080 16054_1.bmp 3702221960041 80038_1.bmp 3702841981070 52715_1.bmp 4108231972072 64473_1.bmp 4108231972072 64473_2.bmp 4108231972072 64473_3.bmp 4113241982030 73439_1.bmp 4113241982030 73439_2.bmp 4113241982030 73439_3.bmp 4114021989031 46501_1.bmp 4114021989031 46501_2.bmp 4114021989031 46501_3.bmp 4115031986100 53338_1.bmp 4130231961101 03344_1.bmp 4130231961101 03344_2.bmp 4130231961101 03344_3.bmp 4202221983092 80439_1.bmp 4202221983092 80439_2.bmp 4304811986051 60581_1.bmp 4304811986051 60581_2.bmp 4305231976032 04314_1.bmp 4305231976032 04314_2.bmp	

C. Training Model According to the Character Images

According to character images are classification stored in the corresponding folder perform to train model.

1) Run the toolbox of the neural network pattern recognition, and then fill neural network parameters of inputs and outputs as shown Fig .3.



Select Data

What inputs and targets define your pattern recognition problem?

Get Data from Workspace

Input data to present to the network.

Inputs: inputs

Target data defining desire network output.

Targets: targets

Samples are: ● Matrix columns ○ Matrix rows

Want to try out this tool with an example data set?

Load Example Data Set

Summary

Inputs 'inputs' is a 345x738 matrix, representing static data: 738 samples of 345 elements.

Targets 'targets' is a 11x738 matrix, representing static data: 738 samples of 11 elements.

To continue, click [Next].

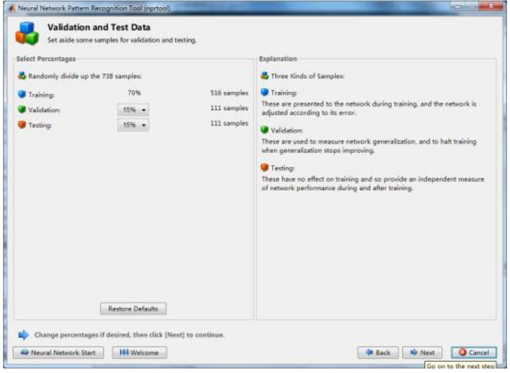
Figure 3. neural network parameters

2) Validation and test data sets are each set to 15% of the original data. With these settings, the input vectors and target vectors will be randomly divided into three sets as shown Fig .4.

a) 70% are used for training;

b) 15% are used to validate that the network is generalizing and to stop training before over fitting;

c) The last 15% are used as a completely independent test of network generalization.



Validation and Test Data

Set aside some samples for validation and testing.

Select Percentages

Randomly divide up the 738 samples:

Training:	70%	516 samples
Validation:	15%	111 samples
Testing:	15%	111 samples

Restore Defaults

Explanation

Three Kinds of Samples:

Training:
These are presented to the network during training, and the network is adjusted according to its error.

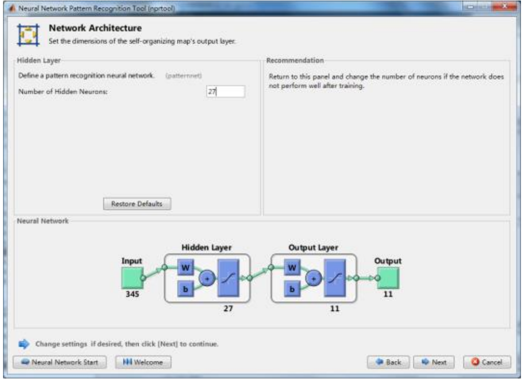
Validation:
These are used to measure network generalization, and to halt training when generalization stops improving.

Testing:
These have no effect on training and so provide an independent measure of network performance during and after training.

Change percentages if desired, then click [Next] to continue.

Figure 4. Training, Validation and Test data sets

3) Hidden number of neurons have a great impact on the generation of the training model, so select the appropriate value in the course of the experiment is very important. Through repeated experiments, define number of hidden neurons to 27 as shown Fig .5.



Network Architecture

Set the dimensions of the self-organizing map's output layer.

Hidden Layer

Define a pattern recognition neural network. (patternnet)

Number of Hidden Neurons: 27

Restore Defaults

Recommendation

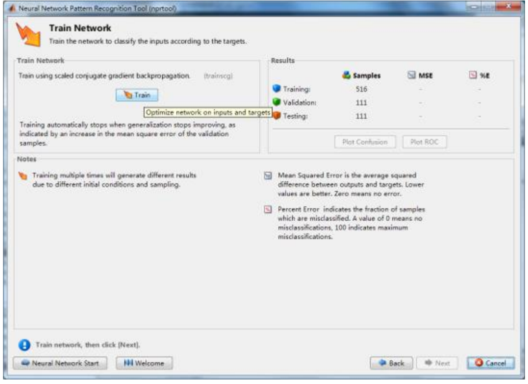
Return to this panel and change the number of neurons if the network does not perform well after training.

Neural Network

Change settings if desired, then click [Next] to continue.

Figure 5. define number of hidden neurons

4) Train the network to classify the inputs according to the targets using scaled conjugate gradient back propagation, and then generate the training model as shown Fig .6.


Neural Network Pattern Recognition Tool (nprtool)
Train
Optimize network on inputs and targets
Plot Confusion
Plot ROC
Neural Network Start
Welcome
Back
Next
Cancel

Train Network

Train the network to classify the inputs according to the targets.

Train Network

Train using scaled conjugate gradient backpropagation. (trainscg)

Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

Results

	Samples	MSE	%E
Training:	516	-	-
Validation:	111	-	-
Testing:	111	-	-

Notes

Training multiple times will generate different results due to different initial conditions and sampling.

Mean Squared Error is the average squared difference between outputs and targets. Lower values are better. Zero means no error.

Percent Error indicates the fraction of samples which are misclassified. A value of 0 means no misclassifications, 100 indicates maximum misclassifications.

Train network, then click [Next].

Figure 6. train the network to generate training model

The algorithm of scaled conjugate gradient back propagation is the simplest optimization algorithm. It updates the network weights and biases in the direction in which the performance function decreases most rapidly, the negative of the gradient. One iteration of this algorithm can be written as following equation (2):

$$\mathbf{x}_{k+1}=\mathbf{x}_{k}-\alpha_{k}\mathbf{g}_{k} \qquad (2)$$

Where $\boldsymbol{x}_k$ is a vector of current weights and biases, $\boldsymbol{g}_k$ is the current gradient, and α_k is the learning rate. This equation is iterated until the network converges.

A list of the training algorithms that are available in the Neural Network Toolbox software and that use gradient method is shown in the following table II.

TABLE II. GRADIENT METHOD

Function	Algorithm
trainlm	Levenberg-Marquardt
trainbr	Bayesian Regularization
trainbfg	BFGS Quasi-Newton
trainrp	Resilient Back propagation
trainscg	Scaled Conjugate Gradient
traincgb	Conjugate Gradient with Powell/Beale Restarts
traincgf	Fletcher-Powell Conjugate Gradient
traincgp	Polak-Ribi ére Conjugate Gradient
trainoss	One Step Secant
traingdx	Variable Learning Rate Gradient Descent
traingdm	Gradient Descent with Momentum
traingd	Gradient Descent

5) Click the "Train" button of Fig .6, generate the training model, which performance as shown Fig .7.

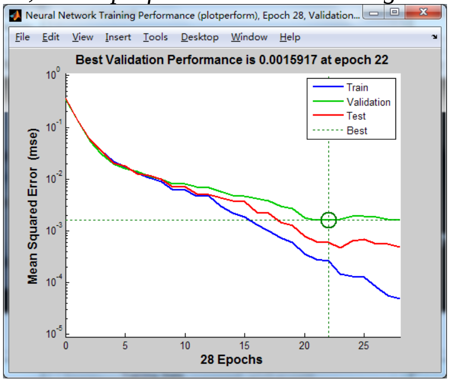


Figure 7. the result of training model performance

IV. RECOGNITION ALGORITHM AND RESULT

A. Segment the Character Image of ID Code Accordig to the Steps of TABLE I.

In order to verify algorithm of the training mode, identified ID card should be segment the character image according to the steps of Table I.

Fig .8 shows the characters image result of segmentation of ID card code. Each character image is shown below in the text box.

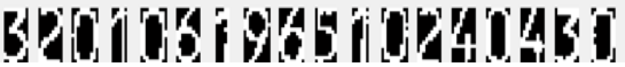
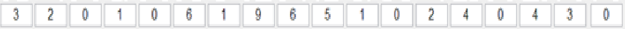


Figure 8. the characters image result

B. Using the Generated the Training Model to Identify Code of the ID Card

The recognition algorithm is based on the generated the training model to identify code of ID cards. Fig .9 shows the recognition result of segmentation of ID card code. Each recognition result is shown below in the text box.



3 2 0 1 0 6 1 9 6 5 1 0 2 4 0 4 3 0

Figure 9. the recognition result

V. VALIDATION ALGORITHM AND RESULT

In order to verification convenience, filename of ID card image is named as same as code of ID card. If character of text box in Fig .9 is not same as code of ID card, the character will be display by red color. Fig .10 shows the verification result of ID card code, which has error code marked by red color.

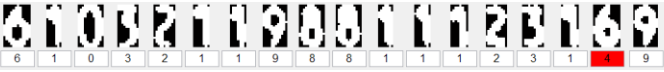
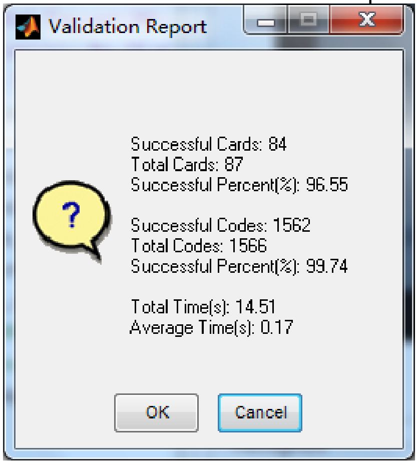


Figure 10. the verification result has error code

In order to verify recognition accuracy rate, Fig .11 shows the verify recognition accuracy rate. ID cards successful rate and codes successful are computed.



Validation Report

Successful Cards: 84
Total Cards: 87
Successful Percent(%): 96.55

Successful Codes: 1562
Total Codes: 1566
Successful Percent(%): 99.74

Total Time(s): 14.51
Average Time(s): 0.17

Figure 11. verify recognition accuracy rate

There are total 87 second-generation ID cards which have been verified. ID cards of 84 are correctly identified. The correct recognition rate is 96.55 percent.

There are total 1566 code characters which have been verified. Characters of 1562 are correctly identified. The correct recognition rate is 99.74 percent.

The total processing time is 14.51 seconds. The average processing time for each second-generation ID card image is 0.17 seconds.

VI. CONCLUSION

The three algorithms of training, recognition and validation have been depicted in the paper. Because the training model is base of ID cards recognition, so the important algorithm is training algorithm which is basis of other algorithms.

Training model is involving in parameters configuration of neural network function. If ID card has been wrongly identified, you need to mark the wrong recognition character, which should be stored in the sample library of character images. The sample data has been added in order to generate the excellent training model which can enhance the recognition accuracy rate of ID cards. This process can continue to cycle until the train recognition rate model substantially the required accuracy.

On the other hand, if the origin ID cards is scanned clearly and distinctly, the recognition accuracy rate can extremely be raised using training model.

The algorithms have been proposed can take advantage of the training model to perfectly realize identification and validation the code of the second-generation ID card; the algorithms have application value in the real work.

REFERENCES

[1] Gao Ju, Ye Hua, Yang Yan Lan. Android platform identification ID number [J]. of information technology and informationization based on 2013,04:116-119+122.

[2] Le Kai, Chen Lian, Cao Jichang. Based on the gray multi value ID card number recognition[J].computer engineering and applications,:.

[3] Hong Yamin. The two generation ID card photo identification [J]. Journal of Lanzhou industry college, 2013,06:18-22.

[4] Zhao Xingwang, Li Tianyang, Wang Liang, Zhou Jing. The two generation ID card number recognition system of [J]. computer and modernization based on digital equipment, 2014,06:132-136.

[5] Qiu Dao Yin, Liu Qingxia. The recognition algorithm MATLAB. Journal of the identity card numbers of North China Institute of water conservancy and hydroelectric power based on [J]., 2012,02:91-94.

[6] Junhuan Zhou (Wuhan Univ. of Technol., Wuhan, China); Jingling Yuan; Haizheng Tao; Hui Huang; Wu Qiu Source: 2013 8th International Conference on Computer Science & Education (ICCSE 2013), p 769-72, 2013

[7] Supardi, J. (Inf. Eng. Dept., Sriwijaya Univ., Sriwijaya, Indonesia); Utami, A.S. Source: International Journal of Machine Learning and Computing, v 4, n 1, p 110-13, Feb. 2014

[8] Graditi, G. (ENEA - Res. Center, Italian Nat. agency for new Technol., Portici, Italy); Ferlito, S.; Adinolfi, G.; Tina, G.M.; Ventura, C. Source: 2014 5th International Renewable Energy Congress (IREC), p 6 pp., 2014

[9] Sivanandan, S., Saiyyad, Y.: Automatic vehicle identification using license plate recognition for Indian vehicles. Int. J. Comput. Appl.2012, 23–28 (2012)

[10] Ibrahim, N.K., Kasmuri, E., Norazira, A.: License plate recognition (LPR): a review with experiments for Malaysia case study. Int. J. Soft Comput. Softw. Eng. 3 (3), (2013)

[11] Dandu, B.R., Chopra, A.: Vehicular number plate recognition using edge detection and characteristic analysis of national number plates. Int. J. Comput. Eng. Res. 2 (3), 795–799 (2012)