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 proposed 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.

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 $\text{net}(\text{performFc})$. The default performance function for feed forward networks is mean square error $\text{mse}$ (the average squared error) between the network outputs $a$ and the target outputs $t$. It is defined as following equation (1):

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

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\textsuperscript{TM} 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.

![ID sample](image)

Figure 2. one of the ID card samples

B. Generate Segmentation Character of ID Code

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

<table>
<thead>
<tr>
<th>Steps</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Part image of ID card is converted to binary image by threshold</td>
<td>441321989082085529</td>
</tr>
<tr>
<td>2. All connected components have fewer than 11 pixels should be removed from a binary image.</td>
<td><img src="image" alt="Binary image" /></td>
</tr>
<tr>
<td>3. Clear the image border by suppressing light structures that are lighter than their surroundings and that are connected to the image border.</td>
<td><img src="image" alt="Clear image border" /></td>
</tr>
<tr>
<td>4. It is dilated using each structuring element in succession, which is created according to specifies the radius.</td>
<td><img src="image" alt="Dilated image" /></td>
</tr>
<tr>
<td>5. Find maximum dilated structuring element that specifies the actual number of pixels in the region.</td>
<td><img src="image" alt="Find maximum dilated structuring element" /></td>
</tr>
<tr>
<td>6. Under the image mask perform morphological reconstruction using 4-connected neighborhood.</td>
<td><img src="image" alt="Morphological reconstruction" /></td>
</tr>
<tr>
<td>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.</td>
<td><img src="image" alt="Character images" /></td>
</tr>
<tr>
<td>8. According to code of ID sample, manual input corresponding correct code as shown in the following Figure.</td>
<td><img src="image" alt="Input corresponding correct code" /></td>
</tr>
<tr>
<td>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</td>
<td><img src="image" alt="Character images" /></td>
</tr>
</tbody>
</table>

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

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.

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.

   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):

$$x_{k+1} = x_k - \alpha_k g_k$$  \hspace{1cm} (2)

Where $x_k$ is a vector of current weights and biases, $g_k$ is the current gradient, and $\alpha_k$ 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>
<thead>
<tr>
<th>Function</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>trainlm</td>
<td>Levenberg-Marquardt</td>
</tr>
<tr>
<td>trainbr</td>
<td>Bayesian Regularization</td>
</tr>
<tr>
<td>trainbfg</td>
<td>BFGS Quasi-Newton</td>
</tr>
<tr>
<td>trainrp</td>
<td>Resilient Back-propagation</td>
</tr>
<tr>
<td>trainscg</td>
<td>Scaled Conjugate Gradient</td>
</tr>
<tr>
<td>traincgb</td>
<td>Conjugate Gradient with Powell/Beale Re-starts</td>
</tr>
<tr>
<td>traincfg</td>
<td>Fletcher-Powell Conjugate Gradient</td>
</tr>
<tr>
<td>traincg</td>
<td>Polak-Ribiere Conjugate Gradient</td>
</tr>
<tr>
<td>trainoss</td>
<td>One Step Secant</td>
</tr>
<tr>
<td>traingdx</td>
<td>Variable Learning Rate Gradient Descent</td>
</tr>
<tr>
<td>traingdm</td>
<td>Gradient Descent with Momentum</td>
</tr>
<tr>
<td>trained</td>
<td>Gradient Descent</td>
</tr>
</tbody>
</table>

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

### REFERENCES


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


