

Fast License Plate Recognition Method Based on Competitive Neural Network

Jinxing Qian and Bo Qu

School of Electronic and Information Engineering, Soochow University, Suzhou, Jiangsu, China

Abstract—Vehicle license plate recognition is an important part of Intelligent Transportation Network. Generally speaking, the traditional LPR mainly consists of three parts: location, segmentation and recognition. Recognition is the most important part of LPR and the effect of recognition is closely related to the result of location segmentation. The accuracy of recognition is the most important index to test the effect of recognition. This paper introduces the recognition method of license plate characters by competitive neural network, and discusses a fast recognition method of the number and letter of vehicle license plate under natural background by using an improved competitive neural network. The results show that the character recognition rate of the images taken under the natural landscape is 91.3%.

Keywords—License plate recognition; neural network; character detection)

I. INTRODUCTION

Artificial neural network includes supervised neural network and unsupervised neural network, also known as tutor neural network and untutor neural network. The corresponding supervised learning and unsupervised learning are also two learning modes of human beings. Supervised learning provides a target output that updates weight by comparing the error between the actual output and the target output. This learning method can quantify the error between the expected and actual output. Unsupervised learning does not have any target output, but this network structure can conduct self-learning and find rules in input data. Traditional competitive neural networks belong to the category of unsupervised learning. Unsupervised learning algorithm is mainly used to find hidden models between data when there is no clear target. The application domain includes machine learning such as data and pattern recognition.

II. COMPETITIVE NEURAL NETWORK

Competitive neural network algorithm is a network model of "winner is king" or "winner-take-all", that is, comparing the output value between output neurons to determine which one is the winner. Essentially, it is an algorithm to find the hidden model between data without a clear target. The data is passed in the from input layer, after the hidden layer neurons process the data, the result is transmitted to the the output layer neurons, neurons in the output layer is to find the largest neurons in the output value, the neuron is the winning one. Compared with other supervised learning algorithm, the difference of competitive learning can only update the winning neuron's weights, while other neurons remain the same, allowing the neurous closer to the input to win the competition[1]. For this

kind of learning, the network will not consider the bias, so the neuron only needs to change the corresponding weight. Figure 1 is the competition neural network model.

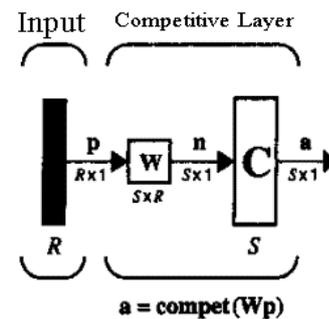


FIGURE I. COMPETITIVE NEURAL NETWORK MODEL

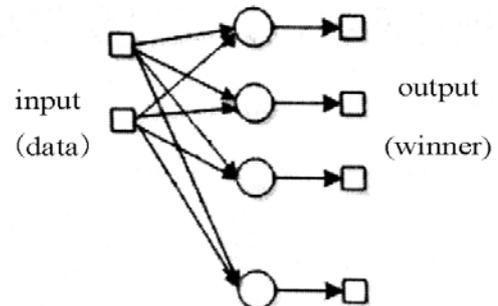


FIGURE II. TWO-INPUT-FOUR-OUTPUT NEURAL NETWORKS

Figure 2 is a single-layer neural competition network with two inputs and four outputs. The input vector is X , All the input neurons and the middle neurons have a weight matrix W , where:

$$X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad W = \begin{pmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \\ w_{13} & w_{23} \\ w_{14} & w_{24} \end{pmatrix}$$

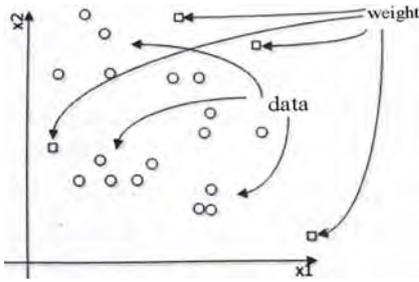


FIGURE III. DATA AND WEIGHT DISTRIBUTION BEFORE TRAINING

The dimension of the input data is the same as the weights. In Figure 3, the circle represents the data point, and the small square represents the weight of the neuron. Among them, some data points are very close to the weight square, some data points are very far away from the weight square, some data points are very close to this weight value but far away from others[2]. The distance $O_j(x)$ between the data of the neural network and the weight value is calculated as follows:

$$o_j(x) = f_j(\sum w_{ij} x_i) \quad (1)$$

The calculated results represent the competition of each sample, and the winning neurons will update the weight by the following formula:

$$\Delta w_{ij} = \alpha(x_i - w_{ij}) \quad (2)$$

where α represents the learning rate. After several iterations, as shown in Figure 4, the weight approaches a fixed value. The centroid formed by neurons around the data points can make it more competitive than other neurons[3].

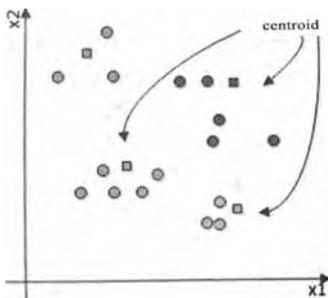


FIGURE IV. DATA AND WEIGHT DISTRIBUTION AFTER TRAINING

III. IMPROVED ALGORITHM OF COMPETITION NETWORK IN LICENSE PLATE RECOGNITION

Competitive neural network is an unsupervised learning scheme. This design of neural network refers to the neural network with supervised learning, which combines the advantages of both to simplify the network and reduce the computational complexity. The weight modification scheme of the design remains the same, but a supervisor is introduced in the training process.

When designing a network, the number of output neurons is equal to the number of classification, so that each neuron points to a classification. After that the input data are classified and then sent to the network training, and after the training, the result will be compared with the prior classification. If the output is different from the previous classification, indicating that the weights of the neuron is far away from the data, then the weights of neuron is modified to be close to the input data, and weight change rule is the same as the above discussion. After several iterations, the neuron will point to the data clustering of corresponding classification to achieve the purpose of classification. At the same time, in order to speed up the training, a normalized certain data instead of a random value as the initial value of the neurons. In training, the inner product of the normalized weight and normalized data is made, which is equivalent to calculating the Euclidean distance between the small squares and the small circle in the figure above. The maximum inner product means the minimum Euclidean distance. Since all the data have been normalized, all the results obtained are reasonable and accurate.

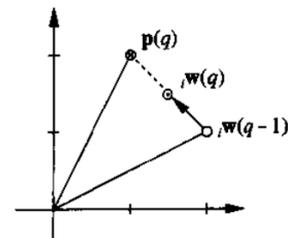


FIGURE V. NEURON WEIGHT MODIFICATION DIAGRAM

As can be seen from Figure 5, a competitive network classifier can be designed by setting the row vector of the matrix as the ideal prototype vector (the number of rows of W is the number of neurons, and the number of columns is the dimension of the sample). Each time we need to modify the weight, only the winning neuron, that is, the data of the corresponding row, is modified[3]. The rule of weight modification is:

$$W_i(q) = W_i(q-1) + \alpha[P_i - W_i(q-1)] \quad (3)$$

Where W_i is the neuron pointing to class i , α is the learning rate, and P_i is the sample belonging to class i . Every time the weight value is modified, it must be normalized and then written to the network.

The training process is shown in Figure 6.

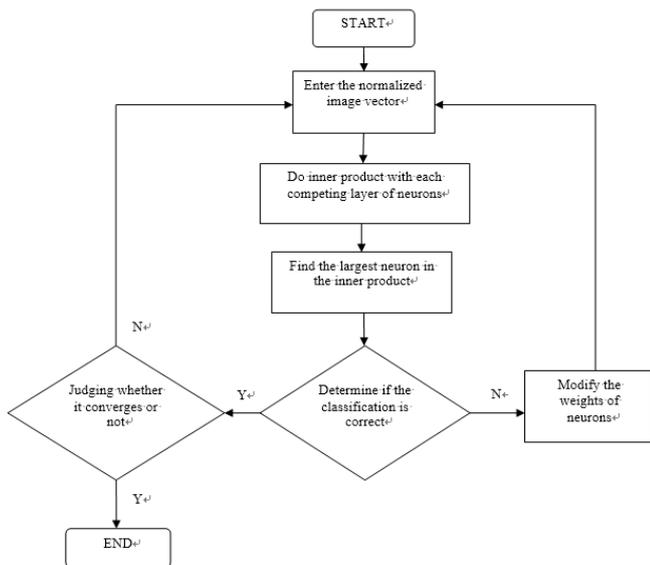


FIGURE VI. TRAINING PROCESS

IV. SPECIFIC IMPLEMENTATION STEPS

The network parameters of this paper are discussed in detail below, as well as the training and recognition of characters. Objects are identified for all Numbers and letters, a total of 34 characters (where the letters I and O are the same as Numbers 1 and 0).

The details are as follows:

The input layer neurons were set to 200, corresponding to the input vector of 200 dimensions.

The hidden layer neurons are set to 34, corresponding to 34 categories.

The output layer neuron is set to 1, corresponding to the winning neuron.

The number of competing neurons is the same as the number of subclasses to be classified. The input data of each neuron is the same as the dimension of the sample without setting the bias vector. So each neuron points to a subclass[4]. When training samples, each neuron and sample are both unitized, and when weight values are modified, they are also unitized. For each sample input, the inner product of the sample and each neuron is calculated separately, and the maximum inner product of the neuron is found. If the neuron points to the correct classification, the weight value is not modified, otherwise, the weight value is modified to offset the neuron vector to the sample direction.

A. Preprocessing License Plate Image

Letters and numbers are mainly detected on the characters of the license plate. The whole construction of letters and numbers is simpler than that of Chinese characters, and the whole pretreatment is easier to implement. First, each character is uniformly resize into a 20*10 pixels image, then the entire image is grayed out and binarized, so as to obtain a 20*10 matrix, and finally, the matrix is classified as a column matrix

with a size of 200*1. At this point each column represents a character image[5].

B. Training for Each Character

During the training, 6 samples were selected for each character for training. First, the first sample of each character's sample set was unitized as the initial value of the weight matrix. Then, the image vector is continuously unitized and input into the network, and make inner product with the weight of each hidden layer neuron. Finally the winning neuron is compared with the target classification, if the winning neuron is not the target neuron, the weight of the corresponding target neuron is modified, so that the weight vector of the target neuron is closer to the input vector.

C. Character Recognition Process

The setting of the target matrix is essentially encoding the output. In order to simplify the model, we used a simple model in the competitive neural network in our experiment: after normalization of input vector and perform an inner product operation with each neuron, the calculation result is the distance between input vector and neuron, and if the nearest one is found, the neuron wins the competition[6]. The winning neuron represents the corresponding classification, more detailed algorithm is as described above.

V. CHARACTER RECOGNITION RESULTS

This paper tested 115 images, most of which were taken by the camera in the natural landscape and partly from the Internet. These images have different resolutions and different illumination conditions. The recognition results are shown in table 1.

TABLE I. RECOGNITION RESULTS

Total pictures	Total characters	Success	Fail
115	690	629	61
(%)	(%)	91.3%	8.7%

VI. CONCLUSION

The advantages and disadvantages of this paper are summarized as follows:

- A. All the images in this test were taken within 2 meters of the license plate under natural light, so the images were relatively clear and did not require much adjustment.
- B. The main advantage of this design network is that the network is simple and the computation is less. In addition, the sample required is far less than traditional competition neural network.
- C. Since the training process is carried out with standard and clear images, it will affect the recognition of tilted license plates and those with impurities.
- D. The system needs to be paid special attention to the situation of overlearning, sometimes there will be shock convergence, so the training completion condition should be set according to the actual situation.

REFERENCES

- [1] Vivek Shrivastava, Navdeep Sharma. Artificial Neural Network Based Optical Character Recognition. Signal and Image Processing: An International Journal (SIPIJ), Vol.3, No.5, October 2012.
- [2] Qeethara K. Al-Ahaya. Artificial Neural Networks in Medical Diagnosis [M]. International Journal of Computer Science Issues, Vol.8, Issue 2, March 2011, pp.145-160.
- [3] Alan M.F. Souza, Carolina M. Affonso, Fábio M. Soares, Roberto C.L. De Oliveira. Soft Sensor for Fluoridated Alumina Inference in Gas Treatment Centers. Intelligent Data Engineering and Automated Learning 2012, Lecture Notes in Computer Science v.7435, Springer Verlag Berlin Heidelberg, 2012, pp.290-315.
- [4] K. Deb, H. U. Chae, and K. H. Jo, "Vehicle license plate detection method based on sliding concentric windows and histogram. Journal of Computers, vol. 4, No.8.2009, pp.353-361.
- [5] Hongliang Bai, Junmin Zhu, Changping Liu, "A Fast License Plate Extraction Method on Complex Background", the 2003 IEEE International Conference on Intelligent Transportation System", pp.985-987, 2003.
- [6] Yoo, S. H., Oh, S. K., & Pedrycz, W. (2015). Optimized face recognition algorithm using radial basis function neural networks and its practical applications. Neural Networks, pp.69-75.