A novel LSTM-RNN decoding algorithm in CAPTCHA recognition

Chen Rui, Yang Jing, Hu Rong-gui, Huang Shu-guang
Department of network
Electronic Engineering Institute
Hefei, China
e-mail: captchafun@163.com

Abstract—**LSTM-RNN has been succeeded in applying in offline handwritten recognition. The paper used two-dimensional LSTM-RNN to recognize text-based CAPTCHA. Aiming at the problem that traditional decoding algorithm cannot obtain satisfactory results. The paper proposed a novel decoding algorithm based on the multi-population genetic algorithm. The experimental results showed that the novel decoding algorithm for the LSTM-RNN can improve the recognition rate of merged-type CAPTCHA.**

Keywords-Network Security; CAPTCHA recognition; decoding algorithm; Recurrent Neural Network

I. INTRODUCTION

Completely Automated Public Turning test to tell Computers and Humans Apart (CAPTCHA) provides a way for automatically distinguishing a human from a computer program, thereby avoiding abuse of the network resource by the computer program. CAPTCHA is a kind of network security mechanism based on hard artificial intelligence problems. Currently the research on CAPTCHA is focused on the design and recognition technology[1-3]. In this paper, the recognition technology of the text-based CAPTCHA is chiefly studied. Nowadays the researchers mainly use the technology of pattern recognition for CAPTCHA recognition[4] such as SVM, Neural Network, HMM and so on. These methods are all based on segmentation, however once character in the CAPTCHA touching or merged are encountered, the segmentation becomes difficult. A Segmentation-free strategy based on two dimension Long-short Term memory Recurrent neural network(2DRNN) [5] is used for merged-type CAPTCHA recognition without segmentation in this paper. Aiming at the problem that traditional decoding algorithm cannot obtain satisfactory results, A decoding algorithm based on the multi-population genetic algorithm is proposed in this paper, and we call it decoding algorithm based on GA for short.

This paper is partially supported by Natural Science Foundation of AnHui Province under grant no. 1208085QF107

II. RELATED WORK

The recognition technology of text-based CAPTCHA normally include these steps: Firstly at preprocess stage the noise and complex background in the CAPTCHA image are cleared, then machine learning algorithm is used for the character string recognition. The recognition technology can be divided into segmentation-based strategy and segmentation-free strategy. In the segmentation-based recognition systems, the boundary of the character is determined first, and then the CAPTCHA image is segmented into individual character image, finally the character image is recognized. The key of the strategy is whether the CAPTCHA image can be segmented correctly. Segmentation is proved to be a very difficult work in handwritten recognition, while explicitly segment can be avoided in segmentation-free strategy. Traditional hidden markov model(HMM) can finish the recognition with segment at the same time. It has been widely used in offline handwritten recognition. It can get high recognition rate with lexicon and linguist model. While there is no lexicon or linguist model in CAPTCHA recognition. Recently a novel HMM called recurrent HMM can finish recognition without lexicon, while the results are not good enough.

LSTM-RNN(1DRNN) is another Segmentation-free strategy, it has been used for CAPTCHA recognition successfully. Vanishing gradient problem is one of the shortcomings of the traditional RNN[6]. With LSTM structure ,1DRNN solved the problem, and it can keep the long context in the network.

978-0-7695-5122-7/13 $26.00 © 2013 IEEE
DOI 10.1109/IMCCC.2013.171

III. A LSTM-RNN DECODING ALGORITHM BASED ON GENETIC ALGORITHM

2DRNN used for CAPTCHA recognition in this paper has made significant improvements on 1DRNN. 1DRNN can learn context in the horizontal direction, while in the vertical direction it relies on the feature selection by human. 2DRNN can learn not only horizontal context, but also the surrounding context automatically. For this reason, 2DRNN should be more adaptable to CAPTCHA recognition. For the decoding algorithm of 2DRNN is the focus of this paper, The details of 2DRNN can be found in the document [5]. The decoding algorithm based on GA is proposed after studying the Connectionist Temporal Classification(CTC) output layer of 2DRNN[7]. So the rest of chapter is organized as follows. The CTC output layer is presented first, and then the decoding algorithm based on the multi-population genetic algorithm proposed is introduced.

A. CTC output layer

Although 2DRNN can address vanishing gradient problem, it cannot make the alignment between input sequence and output sequence directly. The input sequence should be segmented before inputted into 2DRNN, so that every segment sequence can be aligned with individual output label. This problem limits the availability of 2DRNN, so 2DRNN with CTC output layer has been used to address this problem. It output the probability of sequential labels without data pre-segmentation. The CTC output layer includes one more unit than the number of output label alphabet, the extra unit outputs the probability of a 'blank' or no label. The activations of the output layer is softmax function, since the softmax function can normalize the output values of the network.

$$y_m^t = \frac{e^{a_m^t}}{\sum_{m'} e^{a_{m'}^t}} \tag{1}$$

In (1), a_m^t is the activation value of unit m at time t before normalization, e refers to the base of natural logarithms, the range of m' is from zero to the number of output unit. y_m^t is the final output value of unit m at time t note that the normalized output value depends on the whole input sequence.

Linking output label at each step sequentially can get a special output path δ. when the input sequence is x and the output sequence is l, Calculating the posterior probability need the mapping from the path into the label sequence. *Compression* function which deletes first the repeated labels and then the blanks from the paths is used for mapping. For example, The corresponding label sequence of the path （$a,a,-,b,c,c$） and（$-,a,a,b,b,-,c$）is the same (a,b,c). For training data of 2DRNN is the raw data without segment, the alignment between input sequence and output labeling is not clear. Calculating the probability of the output label sequence need sum the probability of all the paths which are mapping to the the same label sequence by *Compression* function.

$$p(l \mid x) = \sum_{\delta \in Com^{-1}(l)} p(\delta \mid x) \tag{2}$$

The number of path which mapped into the label sequence grows exponentially with the length of the input sequence, so a method based on graph called CTC forward-backward algorithm is used. For there is 'blank' label in the path, blank labels are inserted into the head ,end and the inside of the consecutive labels. After that, the original label sequence l was changed into label sequence l', and the length of the new labeling l' is $2|l|+1$.

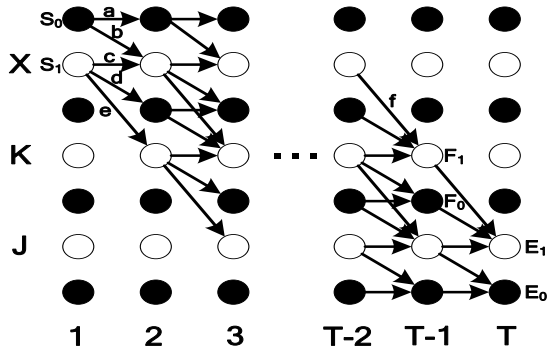

Figure 1. The diagram of forward-backward algorithm

The calculation formulas of the forward variable and backward variable are explained according to the diagram below. In the diagram the white circles represent

non-blank label, while the black circles represent blank label. So the label sequence in the diagram is l=(X,K,J), and l'=(-,X,-,K,-,J). Every column in the diagram correspond to a time step t, so the whole time step T correspond to a path from the upper left circle S_0 or S_1 to lower right circle E_0 or E_1. The transition relation between the two adjacent time steps is defined: the transition of the blank label includes that (1) next time step outputs blank label, as shown by the arrow a. (2) next time step outputs non-blank label, as shown by the arrow b. while the transition of the non-blank label include that (1) next time step outputs the same non-blank label, as shown by the arrow c. (2) next time step outputs blank label, as shown by the arrow d, (3) next time step outputs another non-blank label, as shown by the arrow f.

So the forward variable α_s^t is defined as:

$$\alpha_s^t = \sum_{\substack{\delta: \\ Com(\delta_{1:t})=l_{1:s/2}}} \prod_{t'=1}^{t} y_{\delta_{t'}}^{t'} \tag{3}$$

The formula can be calculated recursively. And the backward variable defined as:

$$\beta_s^t = \sum_{\substack{\pi: \\ Com(\pi_{t:T})=l_{s/2:|l|}}} \prod_{t'=t+1}^{T} y_{\pi_{t'}}^{t'} \tag{4}$$

The details of calculating forward and backward variable can be found in the document[7].

At last, the conditional probability of the CTC layer output sequence l when input sequence is x can be calculated by multiply the forward variable and backward variable in any time step t. and $s \in 1,...,|l'|$.

$$p(l \mid x) = \sum_{s=1}^{|l'|} \alpha_s^t \beta_s^t \tag{5}$$

B. Decoding algorithm based on GA

The decoding algorithm for the CTC layer of the 2DRNN transforms the original output sequence into the label sequence. Through adding the CTC output layer into the 2DRNN, and training with BPTT algorithm. 2DRNN output the label sequence l* corresponding to the maximum posterior probability when input sequence is x.

$$l^* = \arg\max_{l} p(l \mid x) \tag{6}$$

Currently there is no decoding algorithm which can promise getting the best results rapidly. So many easy method are used to get similar results, such as the best path decoding algorithm. It is supposed that the label sequence which respond to the path of maximum probability is the most probability label sequence.

$$l^* \approx Com(\arg\max_{\delta} p(\delta \mid x)) \tag{7}$$

While the best path decoding algorithm cannot guarantee the optimal result.

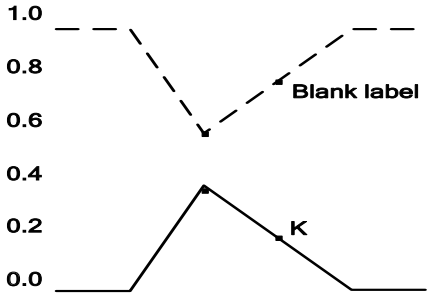

Figure 2. The problem of best path decoding algorithm

As shown in Figure 2, It is supposed that there is only label K and blank label in the label alphabet. The probability of outputting blank label sequence is $p(l=blank)=p(blank,blank)$=0.6*0.8=0.48. and the probability of outputting label sequence K is p(l=K)=p(K, K)+p(K, blank)+p(blank, K)=0.4*0.2+0.4*0.8+0.6*0.2=0.52. If the best path decoding algorithm used for decoding, the output label sequence should be blank. While the optimal result should be label sequence K according to the calculation.

So the document [8] proposed a decoding algorithm based on neighborhood search(DA-NS), while the method only considered the path which appear only one time suboptimal label in the T time steps. The path (K, K) is the exception in the diagram which is not included in the paths considered by the DA-NS, while the output label sequence respond to the path (K, K) is the optimal labeling. Aiming at this problem a decoding algorithm based on the multi-population genetic algorithm is proposed in this paper. A new neighborhood is defined.

The paths considered are allowed to appear more than one time suboptimal label. Suppose the optimal path is $\delta_{\max} = \delta^1_{\max}, \delta^2_{\max}, ..., \delta^T_{\max}$,the neighborhood paths are given as follow:

$$\delta_{sel} = \delta^1_{sel}, \delta^2_{sel}, ..., \delta^T_{sel}, \text{ and } \delta^i_{sel} = \delta^i_{\max} \text{ or } \delta^i_{\sec}, \ i=1,\ldots T \quad (8)$$

In (8), δ^i_{sel} is the output label in time step *i*, it should be the label which correspond to the maximum or the secondly max probability. The neighborhood path is composed with such label through the whole time steps. the number of neighborhood is 2^T, when *T* is larger, decoding process will be slow. So genetic algorithm is applied in the new decoding process.

Genetic algorithm is a global random search algorithm based on the natural selection thought and genetic mechanisms[9]. To prevent the population convergence to local optimal, the multi-population genetic algorithm is used in this paper. Every individual is coded as a binary vector. If the label in time step *i* correspond to the maximum probability, the *i*th value of the vector is 1. If the label correspond to the second max probability, the *i*th value of the vector is 0.

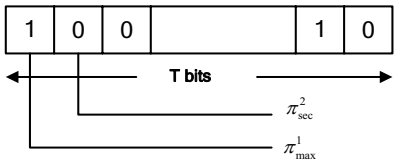

Figure 3. An example of neighborhood encoding by means of a bit string

The goal of the genetic algorithm is to find the output label sequence of the maximum conditional probability. So the fitness function is the posterior probability of the output label sequence which got from the compression function calculated by forward-backward algorithm. The fitness function is given as follow:

$$F(I) = P_{fb}(Com(Path(I))) \quad (9)$$

In (9), *Path*(*I*) is the path of the individual, P_{fb} is the probability of the label sequence calculated by the forward-backward algorithm. After the encode mode and fitness function are selected. The results can be got by the GA algorithm. Suppose sub-population is *M*, the number of individual in the sub-population is *N*, the current generation is *curGen*, the max generation is *maxGen*, the number of individual migrated is *nMig*. The smaller of the array subscript, the better fitness of the individual. The process of the CTC decoding algorithm based on the genetic algorithm is described below.

Step 1: the *M* sub-populations are initialized randomly. The size of each sub-population is *N*.

Step 2: while curGen<maxGen

{

For i=1:M

{the fitness value of each individual in the *i*th sub-population is caculated, new sub-populations are generated by copy, crossing and variation}

For j=1:M-1

{ the *nMig* biggest individuals of the *m*th sub-population are stored into a temporary array, the *nMig* bigger fitness individuals of sub-population *j* are migrated into the $j+1_{th}$ sub-population}

The individuals stored in the temporary array migrate into the first sub-population}

Step 3:the algorithm output label sequence corresponding to the best individual.

In order to accelerate the speed of convergence, the individuals corresponding to the best path decoding algorithm (all the values are zero in the vector)and DA-NS (only a value is one in the vector) are added into the initial population. Through the analysis, the search space of the decoding algorithm is larger than the DA-NS. for the multi-population genetic algorithm is used, the population convergence to local optimal is avoided. So the decoding algorithm based on GA should be better than the best path decoding and DA-NS.

IV. EXPERIMENTAL RESULTS

The aim of the experiment is to evaluate the performance of CAPTCHA recognition method based on 2DRNN, and the new decoding algorithm based on GA.

A. Experimental data

For there are no available public data sets for CAPTCHA recognition. A CAPTCHA generation program was used in this paper for generating experimental data. As shown in the figure 4, the characters in the CAPTCHA are distorted and merged seriously.

Figure 4. Samples for CAPTCHA recognized by proposed method

B. Experimental setting

This paper used 2DRNN for CAPTCHA recognition, the gray value of the CAPTCHA were the feature selected, the feature of the picture was input into 2DRNN from up to down and from left to right sequentially. 4000 CAPTCHA pictures were generated as the experimental sample, the size of train data was 2000, the size of validation data was 1000, and the size of test data was 1000.

Firstly the CAPTCHA images were pre-processed, including cutting the white frame, normalizing the height into 25 pixel, and keeping the ratio between width and height unchanged. 2DRNN had seven layers. There were another three hidden layers except the input and output layers, the unit number of the hidden layers were 2, 10, and 50. And there were a forward layer between the any two hidden layer, the unit number of the forward layer were 6 and 20. Forward layer could decrease the number of the weigh between the two hidden layers. Subsample were used between hidden layer and forward layer. The window of the sample were 3*3 and 2*3. There was no subsample between the third hidden layer and output layer. 2DRNN was trained by BPTT with learning rate of and momentum of 0.9.

The settings of decoding algorithm based on genetic algorithm are shown follows: the number of the sub-population was 5, the size of each population was 80, the max generation was 100, The probability of mutation was 0.03, the probability of crossover was 0.6, the probability of migration was 10.

The program of feature exaction, recognition and decoding algorithm based on GA are coded by C++, compiled with VS2008. The experimental platform is on a machine with the memory 2G, and CPU core i5 2.6 G.

C. Experimental results

TABLE I. THE RECOGNITION RATE OF VARIOUS DECODING ALGORITHMS

Decoding algorithm	Recognition rate on validation set（%）	Recognition rate on test set（%）
The best path deco ding algorithm	50.0	51.2
DA-NS	52.3	52.9
our algorithm	54.9	55.2

The 2DRNN recognition rate with various decoding algorithm are shown in table1. As shown in the table, no matter which kind decoding algorithm used, the merged-type CAPTCHA recognition rate using 2DRNN can reach more than 50%. The recognition rate of the Decoding algorithm based on GA and DA-NS is higher than the recognition rate of best path decoding algorithm. And the decoding algorithm based on GA gets the best results. Otherwise, through calculating the shortest edit distance between recognition result and sample labeling (including delete error, insert error and substitute error), we found the delete error of DA-NS was much higher. The decoding algorithm based on GA increases the recognition rate through decreasing the delete error. Experimental results show that the proposed recognition method based on 2DRNN is efficient, furthermore, the proposed decoding algorithm based on genetic algorithm improves the CAPTCHA recognition rates.

V. CONCLUSION

The study of the CAPTCHA recognition will accelerate the development of the CAPTCHA security technology. The contribution of this paper included two aspects:

1)Two dimensional LSTM-RNN are applied into CAPTCHA recognition. the method can learn vertical context automatically, which is more suitable for CAPTCHA recognition.

2)The proposed decoding algorithm based on genetic algorithm is best than DA-NS and best path decoding. It can improve the recognition rate further.

As shown in the experimental results, the recognition rate of merge-typed CAPTCHA can be improved further. The reliability of CAPTCHA recognition is another research direction[10]. After some results with low confidence rejected, the CAPTCHA recognition rate of the remained sample can be increased. We will study the reject strategy in the further research.

REFERENCES

[1]Haichang G, Wei W, Ye F, "Divide and Conquer: An Efficient Attack on Yahoo! CAPTCHA," Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on, 2012. pp.9-16

[2]Liu P, Shi J, Wang L, Guo L. "An Efficient Ellipse-Shaped Blobs Detection Algorithm for Breaking Facebook CAPTCHA," In: Yuan Y, Wu X, Lu Y, eds. Trustworthy Computing and Services: Springer Berlin Heidelberg, 2013,pp. 420-428

[3]Shu-Guang H, Liang Z, Peng-Po W, Hong-Wei H, "A CAPTCHA Recognition Algorithm Based on Holistic Verification". Instrumentation, Measurement, Computer, Communication and Control, 2011 First International Conference on: IEEE, 2011. pp.525-528

[4]Bursztein E, Martin M, Mitchell J. "Text-based CAPTCHA strengths and weaknesses," Proceedings of the 18th ACM conference on Computer and communications security: ACM, 2011,pp.125-138

[5]Graves A, Fernández S, Schmidhuber J. "Multi-dimensional recurrent neural networks," Artificial Neural Networks–ICANN 2007, 2007,pp. 549-558

[6]Hochreiter S, Schmidhuber J. "Long short-term memory. Neural computation," 1997, vol .9,pp.1735-1780

[7]Graves A, Liwicki M, Fernández S, Bertolami R, Bunke H, Schmidhuber J."A novel connectionist system for unconstrained handwriting recognition," Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2009, vol .31,pp.855-868

[8] Shu-Guang H, Liang Z, Zhao-Xiang S, Rong-gui H. "CAPTCHA recognition method based on RNN of LSTM". Pattern Recognition and Artificial Intelligence, 2011, vol .1,pp.40-47

[9]De Stefano C, Fontanella F, Marrocco C, Scotto di Freca A. "A GA-based feature selection approach with an application to handwritten character recognition," Pattern Recognition Letters, 2013, in press.

[10]Koerich A L, Sabourin R, Suen C Y. "Recognition and verification of unconstrained handwritten words," Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2005, vol.27 , pp.1509-1522