A Discriminative Cascade CNN Model for Offline Handwritten Digit Recognition

Shulan Pan, Yanwei Wang, Changsong Liu, Xiaoqing Ding
State Key Laboratory of Intelligent Technology and
Tsinghua National Laboratory for Information Science
and Technology, Department of Electronic Engineering,
Tsinghua University
Beijing, China
{pansl, wangyw, lcs, dxq}@ocrserv.ee.tsinghua.edu.cn

Abstract

This paper presents a high-performance two-stage cascade CNN model. The main idea behind the cascade CNN model is complementary classification objectives between Stage I and Stage II. Discriminative learning is introduced to train Stage II by feeding back poorly recognized training samples. Experiments have been conducted on the competitive MNIST handwritten digit database. The cascade model achieved the best state-of-the-art performance with an error rate of 0.18%.

1 Instructions

Current automatic handwriting recognition algorithms are good at recognizing handwritten characters. Numerous results have been achieved by researchers using different algorithms, such as k-nearest-neighbor (kNN), support vector machine (SVM), modified quadratic discriminant function (MQDF), neural network (NN), convolutional neural network (CNN), etc. kNN is a simple and reasonable effective algorithm. It has achieved an error rate of 0.94% on the well-known MNIST digit recognition benchmark(Martin Renqiang Min et al., 2009[1]). Both SVM and MQDF have achieved lower error rates which is around 0.50% on the same dataset (Dennis Decoste et al., 2002[2])(Sargur N. Srihari et al., 2007[3]). NN and CNN are among the high-performance algorithms. They have been intensely studied in recent years. For a single network, a six-layer neural network has achieved a mentionable result of a 0.35% error rate (Dan C. Ciresan et al., 2010[4]). Owing to the complementarity between different CNN architectures, combining several CNNs could achieve better results. Later researches concentrated mainly on multi-network integration.

CNN has some natural weakness. One is that they do not make a good use of misclassified samples. It limits a single network's performance as well as its complementarity with other networks. The ordinary multi-network integration rules also meet the bottleneck on increasing the number of fusion networks to improve performance. This paper committed to seek a better fusion rules with relatively few networks. Finally, the discriminative cascade model is proposed for offline handwritten character recognition problem.

2 Related works

CNN has a long history in computer vision. An early example is that LeCun et al. successfully used supervised back-propagation networks to perform digit recognition in 1989. More recently, Dan C. Ciresan et al. proposed a committee of seven CNNs by averaging each network's output probability. It obtained an error rate of 0.27%±0.02% on MNIST [5]. In 2012, Dan C. Ciresan et al. successfully formed multi-column deep neural networks (MCDNN) where thirty-five pretrained CNNs were equally divided into five columns. The final output was generated by averaging outputs of the five columns as well. MCDNN achieved the first human competitive result of 0.23% error rate on MNIST [6]. Starting with LeNet-5[7], CNN has a typically standard structure while variants of the basic design is prevalent in computer vision. Li Wan et al. introduced DropConnect to CNN architectures and obtained a 0.21% error rate[8]. DropConnect is a generalization of Dropout[9] for regularizing large fully-connected layers. Min Lin et al. proposed Network in Network (NIN) in order to increase the representational power of NN[10].

3 The Cascade Model

CNN is hierarchical neural network including two parts: feature extraction layers and classification layers. The feature extraction part is composed of alternating the convolutional layer and the sub-sampling layer. A convolutional layer is parametrized by: the number of maps (M), the size of maps (M_x, M_y), kernel sizes (K_x, K_y), and skipping factors (S_x, S_y)[11]. The output map size is defined as:

$$M_x^n = \frac{M_x^{n-1} - K_x^n}{S_x^n + 1} + 1; \quad M_y^n = \frac{M_y^{n-1} - K_y^n}{S_y^n + 1} + 1 \quad (1)$$

where n indicates the layer index. Let L^n indicates the nth layer in network. Each map in L^n is connected to M^{n-1} maps in L^{n-1} at most. Neurons in the same map share their weights, but have different receptive field.

A sub-sampling layer reduces the number of neurons, i.e. the redundant parameters needed to be calculated, which benefits the training. Max-pooling is a form of nonlinear down-sampling. A max-pooling layer is parametrized by rectangular region size (K_x, K_y)[11]. The outputs of a max-pooling layer is generated by the most active neuron within the non-overlapping pooling region.The kernel size of convolutional layers, subsampling or pooling rectangles as well as skipping factors can be chosen to guarantee only one pixel per

output map of the last convolutional layer after down-sampling.

Fully connected classification layers combines the outputs of the last feature extraction layer into a one-dimension feature vector. A Multi-layer perceptron (MLP) classifier with softmax activation function $f_i(x) = P(y = i\,|x, W, b) = \frac{e^{W_i x + b_i}}{\sum_j e^{W_j x + b_j}}$, input x and label y minimizes the negative log-likelihood loss [12]:

$$L(f, (x, y)) = -\sum_i \log f_i(x) \tag{2}$$

Note that $\sum_i f_i(x) = 1$ and $0 < f_i(x) < 1$.

SVM is a good substitute for MLP. SVM with different kernel functions can transform a nonlinear separable problem into a linear separable problem by projecting original data into high-dimension feature space[13]. The soft margin SVM tends to find the optimal separate hyperplane where dataset $D = \{(x_i, y_i)\}$ can be linearly separated by solving the following primal problem:

$$\min P(\theta = \{W, b\}, \xi) = \frac{1}{2} W^T W + C \sum_{i=1}^{|D|} \xi_i \tag{3}$$

$$subject\ to: \begin{cases} y_i(W^T \phi(x_i) + b) \geq 1 - \xi_i \\ \xi_i \geq 0, \quad i = 1, 2, \cdots |D| \end{cases} \tag{4}$$

where C and ξ represent the penalty parameter and slack variables. $|D|$ is the size of dataset. The CNN-SVM hybrid model can be formed in several ways, i.e. [13][14][15].

The cascade model is composed of eleven different networks. The diagram of the two-stage cascade model for offline handwritten digit recognition system is given in Fig.1.

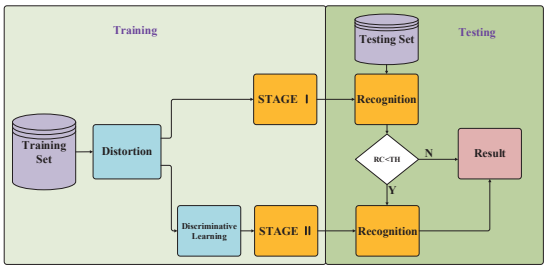

Figure 1. The block diagram of the cascade model for offline handwritten digit recognition system.

Stage I is a committee of four CNN-MLP networks and two CNN-SVM networks. This stage is aimed at well writing samples. It has a relatively high recognition accuracy to guarantee the whole performance. Stage II is a committee of three CNN-MLP networks and two CNN-SVM networks. Stage II, on the contrary, focuses on the minority poorly writing samples. Discriminative learning method is applied to improve this stage's recognition capability of poorly writing samples. Detailed information about discriminative learning method is placed in the next section.

When using softmax as the last layer's activation function, each neuron's output represents one class probability. Linear confidence accumulation[16] is considered within each stage to form the committee. Each class's combination probability $p(w_j\,|x)$ is calculated as a linear function of the N networks' predict probability $p(w_{ij}\,|x)$.

$$p(w_j\,|x) = \alpha_j \cdot \frac{1}{N} \sum_{i=1}^{N} p(w_{ij}\,|x) \quad j = 0, 1, \cdots, 9 \tag{5}$$

where α_j is the weighting parameter calculated by all networks' voting. Let $v_j(x)$ represents the jth class's frequency voted by the N networks' recognition results.

$$\alpha_j = \frac{v_j(x) + 1}{\sum_{i=0}^{9}(v_i(x) + 1)} = \frac{v_j(x) + 1}{N + 10} \quad j = 0, \cdots, 9 \tag{6}$$

Normalization is applied to the above probability afterwards.

All testing samples will be sent to Stage I and get a recognition confidence RC. If RC is above the threshold Th_1, the system will use the recognition result of Stage I as the prediction. Otherwise, the testing samples will be passed to Stage II and the system's prediction will be the recognition result of Stage II. The final decision of the cascade CNN model is derived as follow:

$$w(x) = \begin{cases} \underset{j=0,\cdots,9}{\arg\max}\ p(w_j^1\,|x) & if\ RC > Th_1 \\ \underset{j=0,\cdots,9}{\arg\max}\ p(w_j^2\,|x) & if\ RC \leq Th_1 \end{cases} \tag{7}$$

where w^1, w^2 represent candidate results of Stage I and Stage II.

4 Discriminative Learning

CNN minimizes the Empirical risk [17] on the training set denoted by the following regularized loss function [12].

$$E(\theta = \{W, b\}, D) = -\sum_{i=0}^{|D|} \log P^{(i)} + \lambda \|\theta\|_p^p \tag{8}$$

$P^{(i)} = P(Y = y^{(i)} \mid x^{(i)}, \theta)$, θ denotes set of all parameters for a given model including weights matrix W and bias vector b. $D = \{(x^{(i)}, y^{(i)})\}$ is the dataset. $|D|$ is the size of dataset. Adding regularization parameter $\|\theta\|_p^p$ is an effective way to combat overfitting. λ controls the importance of the regularization parameter. CNN tends to perform better if testing sets have more similar primal features with the training set. In this paper, the cascade model feeds mis-classified and poorly recognized samples whose output recognition confidence RC is below the artificially set threshold Th_2 back to the input to adjust the sample proportion of the training set. In particular, the cascade model only feeds the original training samples back rather than the distorted samples in consideration of the small size of MNIST training set.

Properly setting Th_2 depends on the type of networks. For CNN, each sample's output probability is very close to"1". Accordingly Th_2 is high and close to"1" too. Fig.2 shows the number of samples feeding back with different thresholds.

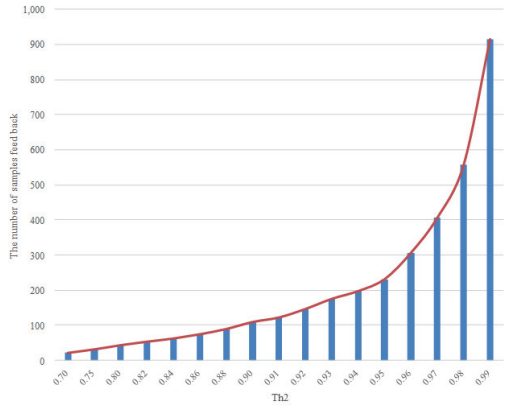

Figure 2. The number of characters feeding back with different threshold.

It is necessary to redouble the selected samples several times before feeding them back in order to efficiently adjust the proportion of the training set. In the CNN cascade model, Th_2 is set to be 0.95. All the selected samples are redoubled three to six times for each network to get about one thousand samples. They are around 2% of the original fifty thousand training samples. If Th_2 is too low, e.g. below 0.85, there are only a handful of samples selected which will make the network focus on fixed minority samples thus reducing the network's generalization. On the contrary, if Th_2 is too close to one, many well writing samples will be selected as well and the discriminative learning will be less efficient.

5 Experiments

The cascade model has been verified on MNIST dataset. MNIST contains 70000 digit samples: 50000 samples for training, 10000 samples for validation and 10000 samples for testing.

We first train different networks separately and then combine them to build the cascade system. Before training each separate network, three types of distortion are applied to expand the original training set. They are rotation distortion, elastic distortion, shearing and local resizing distortion. The distortion parameters are chosen basing on previous experiments:

(1) σ and α : real-valued parameters for elastic distortion (Y.Simard et.al., 2003[18]). σ is the standard deviation of the Gaussian filter. α is the scaling parameter controlling the amplitude of the distortion. Here we set $\sigma = 6.0$ and $\alpha = 36.0$. In this paper, a derivation named gradient-based elastic distortion is proposed to reform rationality and controllability of the original elastic distortion.

(2) a and k : random values for shearing and local resizing distortion (K.C.Leung et.al., 2009[19]). a is the resizing range and k is the shearing slope. The deformation effect is like bending the rubber band. We set $\|a\| \leq 2$ and $\|k\| \leq 0.04$.

(3) β : a random angle in $[-7.5°, 7.5°]$ for rotation distortion.

For CNN-SVM, this paper selects RBF kernel with parameter $\sigma = 2^{-6}$ and penalty coefficient $C = 32$.

All networks were completed using Theano. All the experiments were performed on the same computer using GPU to speed up the training.

Table 1. Each network's structure and error rate on test set.

Network	Structure	Error rate%
Net 1	20C-MP-50C-MP-500N-10N	0.40
Net 2	20C-MP-60C-MP-1000N-10N	0.44
Net 3	25C-MP-50C-MP-1000N-10N	0.39
Net 4	30C-MP-60C-MP-1000N-10N	0.42
Net 5	30C-MP-50C-MP-1000N-SVM	0.40
Net 6	30C-MP-60C-MP-1200N-SVM	0.44
Net 7	20C-MP-50C-MP-900N-10N	0.42
Net 8	25C-MP-50C-MP-900N-10N	0.42
Net 9	30C-MP-50C-MP-900N-10N	0.43
Net 10	30C-MP-50C-MP-900N-SVM	0.44
Net 11	30C-MP-60C-MP-1000N-SVM	0.40
Average		0.23

Net 1 to Net 6 belong to Stage I; Net 7 to Net 11 belong to Stage II. Discriminative learning is applied to Net 7 to Net 11. For a single network, discriminative learning is sometimes a trade-off. It will naturally debase the performance on the whole test set in most cases because it turns network to pour more attention on poor writing styles. However, networks trained with discriminative learning demonstrate higher complementarity with other networks.

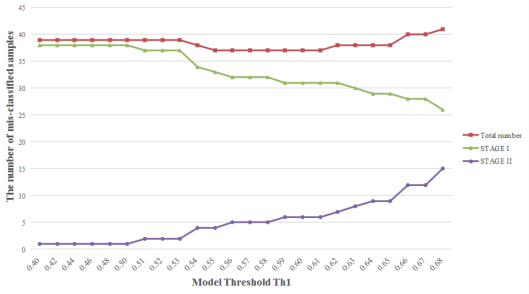

Figure 3. The number of mis-classified samples with different thresholds.

The threshold Th_1 in the cascade model was selected on MNIST validation set. The total number of misclassified samples firstly goes down to the minimum point and then rolling up. When Th_1 is less than 0.40, no validation sample is passed to Stage II. As Th_1 gradually increases, a few poorly writing validation samples are passed to Stage II and they are correctly recognized. So the system's error rate goes down. When Th_1 is in [0.55, 0.61], the whole system achieves the best performance. As Th_1 continues increasing, more validation samples are passed to Stage II including a multitude of well writing samples. Since Stage II focuses on poorly writing samples, the system's error rate goes up. The cascade model does not "see" validation

samples formerly thus the average RC of validation set is lower than the training set. That's the reason why best performance is achieved when Th_1 is around 0.60 lower than Th_2 in discriminative learning.

In this paper, Th_1 is set to be 0.60. We tested the performance of the cascade model on MNIST test set and obtained an error rate of 0.18%. Comparing with simply averaging the eleven CNNs (0.23%), the discriminative cascade model achieves better results (0.18%). It benefits from high complementarity of the two stages classification and the discriminative learning.

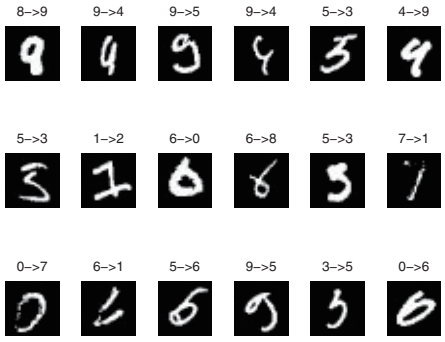

Figure 4. Mis-classified samples on test set: correct label → predict label.

6 Conclusion

This paper presents a discriminative cascade model for offline handwritten digit recognition problem. The recognition problem has been divided into two stages. Stage I aims at well writing samples while Stage II aims at poorly writing samples. The compensation between the two stages makes their cascade combination outperforming each of them alone. In the future work, directly applying discriminative learning on CNN architectures will be explored and studied comprehensively.

References

[1] Min M R, Stanley D A, Yuan Z, et al.: "A Deep Non-linear Feature Mapping for Large-Margin kNN Classification," *Ninth IEEE International Conference on Data Mining(ICDM 09)*, IEEE Press, Dec. 2009, pp.357–366, doi: 10.1109/ICDM.2009.27.

[2] Decoste D, Scholkopf B.: "Training invariant support vector machines," *Machine Learning*, vol.46, Jan.2002, pp.161–190, doi: 10.1023/A:1012454411458.

[3] Srihari S N, Yang X, Ball G R.: "Offline Chinese handwriting recognition: an assessment of current technology," *Frontiers of Computer Science in China*, vol.1, May.2007, pp.137–155, doi:10.1007/s11704-007-0015-2.

[4] Claudiu Ciresan D, Meier U, Gambardella L M, et al.: "Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition," *Neural Computation*, vol.22, Dec.2010, pp.3207–3220, doi: 10.1162/NECO_a_00052.

[5] Ciresan D C, Meier U, Gambardella L M, et al.: "Convolutional neural network committees for handwritten character classification," *ICDAR 2011*, IEEE Press, Sept.2011, pp.1135–1139, doi:10.1109/ICDAR.2011.229.

[6] Ciresan D, Meier U, Schmidhuber J.: "Multi-column deep neural networks for image classification," *IEEE Conference on.Computer Vision and Pattern Recognition (CVPR 2012)*, IEEE Press, June.2012, pp.3642–3649, doi: 10.1109/CVPR.2012.6248110.

[7] LeCun Y, Boser B, Denker J S, et al.: "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol.1, Dec.1989, pp.541–551, doi: 10.1162/neco.1989.1.4.541.

[8] Wan L, Zeiler M, Zhang S, et al.: "Regularization of neural networks using dropconnect," *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp.1058–1066.

[9] Hinton G E, Srivastava N, et al.: "Improving neural networks by preventing co-adaptation of feature detectors,"

[10] Min Lin, Qiang Chen, et al.: "Network in network," *International Conference on Learning Representations(ICLR 2014)*,

[11] Ciresan, Dan, et al.: "A committee of neural networks for traffic sign classification," *The 2011 International Joint Conference on Neural Networks (IJCNN)*, IEEE Press, Aug.2011, pp.1918–1921, doi: 10.1109/IJCNN.2011.6033458.

[12] Deep Learning: `http://deeplearning.net/`

[13] Niu X X, Suen C Y.: "A novel hybrid CNNCSVM classifier for recognizing handwritten digits," *Pattern Recognition, 2012*, vol.45, April 2012, pp.1318–1325, doi: 10.1016/j.patcog.2011.09.021.

[14] Nagi, Jawad, et al.: "Convolutional Neural Support Vector Machines: Hybrid Visual Pattern Classifiers for Multi-robot Systems," *11th International Conference on Machine Learning and Applications (ICMLA)*, vol.1, Dec. 2012, pp.27–32, doi: 10.1109/ICMLA.2012.14.

[15] Tang, Yichuan.: "Deep learning using linear support vector machines," *Workshop on Challenges in Representation Learning, ICML*, 2013.

[16] Xin Li, et al.: "An MQDF-CNN Hybrid Model for offline handwritten Chinese character recognition," *14th International Conference on Frontiers in Handwriting Recognition (ICFHR 14)*.

[17] Lauer, Fabien, Ching Y. Suen, and Grard Bloch.: "A trainable feature extractor for handwritten digit recognition," *Pattern Recognition*, vol.40, June.2007, pp.1816–1824, doi: 10.1016/j.patcog.2006.10.011.

[18] Simard, Patrice, David Steinkraus, and John C. Platt.: "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis," *ICDAR. 2003*, IEEE Press, Aug.2003, pp.958–963, doi: 10.1109/ICDAR.2003.1227801.

[19] Leung K C, Leung C H.: "Recognition of handwritten Chinese characters by combining regularization, Fisher's discriminant and distorted sample generation," *ICDAR 2009*, IEEE Press, July.2009, pp.1026–1030, doi: 10.1109/ICDAR.2009.48.