Handwritten Character Recognition by Alternately Trained Relaxation Convolutional Neural Network

Chunpeng Wu, Wei Fan, Yuan He, Jun Sun, Satoshi Naai
Fujitsu Research & Development Center Co. Ltd., Beijing, 100025, China
{wuchunpeng, fanwei, heyuan, sunjun, naai}@cn.fujitsu.com

Abstract—Deep learning methods have recently achieved impressive performance in the area of visual recognition and speech recognition. In this paper, we propose a handwriting recognition method based on relaxation convolutional neural network (R-CNN) and alternately trained relaxation convolutional neural network (ATR-CNN). Previous methods regularize CNN at full-connected layer or spatial-pooling layer, however, we focus on convolutional layer. The relaxation convolution layer adopted in our R-CNN, unlike traditional convolutional layer, does not require neurons within a feature map to share the same convolutional kernel, endowing the neural network with more expressive power. As relaxation convolution sharply increase the total number of parameters, we adopt alternate training in ATR-CNN to regularize the neural network during training procedure. Our previous C-NN took the 1st place in ICDAR’13 Chinese Handwriting Character Recognition Competition, while our latest ATR-CNN outperforms our previous one and achieves the state-of-the-art accuracy with an error rate of 3.94%, further narrowing the gap between machine and human observers (3.87%).

Keywords-handwritten character recognition; convolutional neural network; relaxation convolution; alternate training

I. INTRODUCTION

Despite considerable efforts have been devoted to handwriting recognition, it is still a challenging problem due to the presence of cursive writing, touching strokes, and confusion in shapes. For handwritten digit recognition, methods based on different classifiers are proposed, such as KNN [1], [2], SVM [3] and Boosting [4]. As to handwritten Chinese character recognition, modified quadratic discriminant function (MQDF) and its variants are the mainstream classification methods for the past decades [5], [6], [7]. Hand-crafted features (e.g., 8-direction gradient [8]), and feature selection algorithms are key components of the above methods [9]. Recently, deep neural network (DNN) has been proposed for handwritten English word recognition by automatic feature learning [10]. DNN also yields the best recognition accuracy on handwritten digit dataset [11] and handwritten Chinese character dataset [12].

DNNs are breaking records and improving state-of-the-art recognition performance for many difficult visual tasks. Bissacot et al. [13] propose a HoG-input DNN for recognizing characters in natural scenes, and achieve the best performance in ICDAR’13 Robust Reading Competition. Sermanet et al. [14] present an integrated framework for adopting DNN for image classification, localization and detection, taking the 1st place in ImageNet’13 Competition. Wang et al. [15] train a stacked auto-encoder to learn generic image features for video tracking, achieving the state-of-the-art on some hard benchmark video sequences. Ciresan et al. [16] are several champions of neuron membrane segmentation, traffic sign classification, handwriting recognition and etc., based on their multi-column DNNs (MCDNN). The above encouraging progress of DNN can be attributed to factors as follows: (1) High performance computing techniques (e.g., CPU clusters and GPU farms); (2) Flexible structure of neural networks (e.g. Lin. [17]); (3) Availability of larger training datasets; (4) Effective learning algorithm, especially model regularization method (e.g., dropout [18]).

A significant difference between previous neural networks and current DNNs is the scale of the model. Recent success on various visual and acoustic tasks show that scaling up neural networks could dramatically improve the recognition performance [19]. However, two problems brought by the scaling up are slow convergence and over-fitting.

To accelerate the training convergence, it is popular to train DNNs with unsupervised layer-wise pre-training followed by back-propagation fine-tuning, according to Hinton et al.’s breakthrough [20]. But more recent experiments [21], [22] show that unsupervised pre-training may be less necessary when more training data and careful weight initialization are available [23], [24]. In practice, high performance computing techniques, i.e., CPU clusters and GPU farms, have made it possible to train large-scale neural networks with stochastic gradient descent (SGD) in several days, instead of months or even years [25], [26], [27]. But because of SGD’s inherent problem—large variance [28], the multi-core CPU and GPU will not sharply speed up the training procedure unless the DNN structure (e.g., convolutional layer and spatial-pooling layer) and the scale of training dataset are appropriate for the related task [19], [23].

As to the second problem, over-fitting, large-scale neural networks with millions or billions of parameters are prone to over-fitting, especially when they are trained on small training sets. Therefore regularizing DNNs has become a popular research subject [29]. Dropout [18] randomly drops activations, and dropconnect [11] randomly drops weights are two typical stochastic regularizing techniques. These
two methods are only suitable for full-connected layers. Zeiler et al. [30] propose a method which makes spatial-pooling a stochastic process. However, as the experiments shown in [11], regularization methods such as dropout and dropconnect achieve superior performance but slow down the convergence. Dropconnect is even slower than dropout.

We propose relaxation convolution and alternate training to solve the above two problems. Our relaxation convolution used in R-CNN is to enhance the learning capability of the neural networks. Our alternate training used in ATR-CNN is to regularize the neural network by alternately training a subset of layers of a CNN, which also helps to improve the recognition accuracy. We evaluate our proposed R-CNN and ATR-CNN for recognizing handwritten digits and handwritten Chinese characters. Our experimental result shows that our ATR-CNN achieves the state-of-the-art for both tasks.

The rest of this paper is organized as follows: The proposed R-CNN and ATR-CNN are described in Section II. Then Section III presents the experimental results on datasets of handwritten digits and handwritten Chinese characters. Section IV analyzes the CNN structures used for different handwriting tasks. Section V draws the conclusions.

II. PROPOSED METHOD

A typical CNN consists of convolutional layer, spatial-pooling layer and full-connected layer. The convolutional layer is responsible for extracting features from feature maps at lower layer. A new operation—relaxation convolution is proposed to expand the complexity of the convolutional layer, enhancing the learning capability of the neural networks.

Traditionally, all layers of a CNN are trained together using back-propagation algorithm. However, a strategy of alternating training a subset of layers of a CNN is proposed to regularize the neural network.

A. R-CNN: Relaxation Convolution

Fig. 1 compares the convolution adopted in traditional CNN and the relaxation convolution adopted in our R-CNN. The difference between these two types of connection is whether neurons within a feature map share the same convolutional kernel. As shown in the left part of Fig. 1, two neurons $n_1$ and $n_2$ share the same weight matrix $W_1$ (or $W_2$) for traditional convolution. While these two neurons $n_1$ and $n_2$ use different weight matrices $W_1$ and $W_2$ for relaxation convolution as shown in the right part of Fig. 1.

Specifically, given a CNN, a feature map at $i$th layer are defined as $F^i_{i,j}$ ($i = 1, 2, ..., L$, and $j = 1, 2, ..., N_i$), where $L$ is the number of layers, and $N_i$ is the total number of feature maps at $i$th layer. The height and width of the feature map $F^i_{i,j}$ are respectively $H$ and $W$.

Neurons within the feature map $F^i_{i,j}$ are then defined as $Neuron^s_{i,j}$ ($s = 1, 2, ..., H$, and $t = 1, 2, ..., W$). The corresponding convolutional kernel used by each $Neuron^s_{i,j}$ is defined as $Kernel^s_{i,j}$. Traditional convolution requires that all kernels $Kernel^s_{i,j}$ related to the feature map $F^i_{i,j}$ are the same, i.e., there is only one convolutional kernel for each feature map $F^i_{i,j}$. However, our relaxation convolution removes this requirement. In our method, all these kernels $Kernel^s_{i,j}$ are independently randomly initialized and adjusted by back-propagation during training.

Relaxation convolution can be considered to enhance the learning ability of the neural network. Each neuron at upper layer uses different kernels to extract features from lower layer, therefore various deformations of handwriting caused by cursive writing, touching strokes and uncontrolled oscillations of the hand muscles [31] are more probable to be captured by R-CNN. The nature of relaxation convolution, endows the neural networks with more expressive power, comparing to traditional convolution. In our R-CNN, relaxation convolution layers are used to replace some traditional convolutional layers and full-connected layers, as shown in Section III.

B. ATR-CNN: Alternate Training

Given a R-CNN, the weight matrices between adjacent layers are defined as $W_i$ ($i = 1, 2, ..., L$), and the corresponding learning rates are $\eta_i$ ($i = 1, 2, ..., L$). Randomly choosing one weight matrix $W_i$, and fix its learning rate $\eta_i$ to zero. After training this R-CNN using back-propagation algorithm for pre-defined SE epochs, the learning rate of weight matrix $W_i$ is reverted to its original value $\eta_i$, while the learning rate of another weight matrix $W_j$ ($j = \ldots$, $W_L$).
1, 2, ..., L) is randomly selected to fix to zero, and repeat the above procedure. The details are described in Table I. The condition not converging in Table I can be measured by error rate on validation set. The variable sub-epoch is a counter for a loop during which a certain learning rate \( \eta \) is fixed to zero. The variables \( C_i (i = 1, 2, ..., L) \) are predefined constant value. The standard forward-propagation and back-propagation are adopted.

The advantage of adopting alternate training is that the neural network is regularized, though relaxation convolution sharply increases the number of parameters which makes the learning problem more difficult. We will show in Section III that alternate training also contributes to the improvement of recognition accuracy, as weight parameters are finely tuned by using alternate training, comparing with traditional altogether training.

### III. EXPERIMENTAL RESULTS

Six types of layers are used in our R-CNN and ATR-CNN: input layer (In), convolutional layer (Conv), max-pooling layer (MaxP), full-connected layer (Full), relaxation convolution layer (RX) and output layer (Out). The parameter settings of one layer are denoted by “XYZ” in the subsections below. X is the number of feature maps, Y is the type of current layer and Z is the kernel size, e.g., “64Conv5” indicates a convolutional layer with 64 feature maps and 5x5 kernels. Some traditional convolutional layers and full-connected layers are replaced by relaxation convolution layers in our R-CNN and ATR-CNN, which is described in the following subsections.

All experiments use mini-batch SGD with each batch including 128 samples. The activation function is Rectified Linear Units (ReLUs) as described in [32]. All parameters of our R-CNN and ATR-CNN are randomly initialized using a (0,1) uniform distribution. The training datasets are augmented by introducing random 5%-15% translation, rotation and scaling. Cropping images [11] is not used for distorting training samples in our method. As to our alternate training, the constant value \( SE \) is fixed to 10, and \( C_i (i = 1, 2, ..., L) \) are fixed to 0.001.

Totally 10 R-CNNs (or ATR-CNNs) are trained for each recognition task. We use the averaged recognition accuracy of these 10 models as our recognition performance. We will also give experimental results based on voting committee of our R-CNNs (or ATR-CNNs), in order to compare with the results reported in [11], [22]. We implement our R-CNN and ATR-CNN using NVIDIA GeForce GTX 690, with Intel Xeon E5-2620 (2.0GHz) and 64GB RAM. It takes us 5-7 days to train a R-CNN or an ATR-CNN.

#### A. Handwritten Digits

Handwritten digit dataset MNIST consists of a training set of 60,000 samples, and a test set of 10,000 samples. Each digit is normalized to fit in a 20x20 rectangle, and is centered in a 28x28 image by computing the center of mass of the pixel.

In our method, the parameter settings of R-CNN or ATR-CNN are In-32Conv5-32MaxP2-64Conv3-64MaxP2-64RX3-64RX3-Out. Totally two relaxation convolution layers, instead of traditional full-connected layers, are adopted between the second max-pooling layer and the final 10-class soft-max output layer. Table II compares the error rate of different methods [1], [2], [3], [11], [18], [31], [33] on MNIST. Our ATR-CNN achieves the lowest error rate. To make the results comparable, the error rates of a single CNN are reported for the last five CNN based methods in Table II, according to the respective papers.

Fig. 2 shows the misclassified samples by one of our ATR-CNN whose error rate is 0.24%. The lower label in Fig. 2 is “ground-truth\( \rightarrow \)prediction”. The misclassified samples are due to cursive writing, missing strokes and stroke touching. Most of these samples are difficult for a machine to make a correct prediction.

#### B. Handwritten Chinese Characters

Table III lists all datasets of handwritten Chinese characters involved in this subsection. Each dataset covers at least GB2312-80 level-1 Chinese characters (which contains 3,755 classes in total). Only the samples from these 3755 classes are used in the experiments of this subsection. The ICDAR’13 Competition Dataset shown in Table 2 is testing set, while different methods select subset of first five
Table II  
RECOGNITION ACCURACY ON MNIST

<table>
<thead>
<tr>
<th>Method</th>
<th>Model</th>
<th>Error Rate(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecun et al. [33]</td>
<td>Boosted Letnet-4</td>
<td>0.70</td>
</tr>
<tr>
<td>Mizukami et al. [2]</td>
<td>KNN</td>
<td>0.57</td>
</tr>
<tr>
<td>Lauer et al. [3]</td>
<td>TFE-SVM</td>
<td>0.54</td>
</tr>
<tr>
<td>Keysers et al. [1]</td>
<td>KNN</td>
<td>0.52</td>
</tr>
<tr>
<td>Simard et al. [31]</td>
<td>CNN</td>
<td>0.40</td>
</tr>
<tr>
<td>Wan et al. [11]</td>
<td>CNN+DropConnect</td>
<td>0.280±0.032</td>
</tr>
<tr>
<td>Hinton et al. [18]</td>
<td>CNN+DropOut</td>
<td>0.280±0.016</td>
</tr>
<tr>
<td>Our Method</td>
<td>R-CNN</td>
<td>0.274±0.021</td>
</tr>
<tr>
<td>Our Method</td>
<td>ATR-CNN</td>
<td>0.254±0.014</td>
</tr>
</tbody>
</table>

1 To make the results comparable, the error rates of a single CNN are reported for the last five CNN based methods, according to respective papers.

Figure 2. Misclassified digit samples by one of our ATR-CNN whose error rate is 0.24%. The lower label is “ground-truth→prediction”. The misclassified samples are due to cursive writing, missing strokes and stroke touching. Most of these samples are difficult for a machine to make a correct prediction.

datasets as training set or validation set. CASIA-HWDB 1.0-1.1 are offline handwriting datasets of isolated characters, CASIA-HWDB 2.0-2.2 are offline text datasets, and isolated characters can be obtained by segmentation of text lines and characters based on ground-truth. The ICDAR’13 Competition Dataset is written by 60 writers who do not contribute to the above CASIA-HWDB datasets. Details of these datasets can be found in [5].

In our method, the parameter settings of our R-CNN or ATR-CNN are In-64Conv5-64MaxP2-128Conv3-128MaxP2-128RX3-128MaxP2-256RX3-256Full1-Out. The first relaxation convolution layer is adopted between two max-pooling layer, and the second one is before a full-connected layer. CASIA-HWDB 1.1 is our training set, and subset of CASIA-HWDB 1.0 is our validation set. The character images are binarized using Ostu method, and resized to 40x40 pixels and placed in the center of a 48x48 image with linear moment normalization [34]. The training algorithm is stopped if no significant improvement is observed on the validation set.

Table IV compares the error rate of different methods [5], [6], [7], [12], [22] on ICDAR’13 Competition Dataset. We achieve the lowest error rate 3.94% by 4 ATR-CNNs voting, though our training set consists of only CASIA-HWDB 1.1. Furthermore, our performance is almost the same as that of human observers (3.87%) [12]. We previously achieved the 1st place in ICDAR’13 Competition [12] as shown in Table IV, the error rate then was 5.23% by 4 CNNs voting. To make the results comparable, we report Ciresan et al.’s error rate 4.35% using 4 MCDNNs voting committee, and the lowest error rate of a single MCDNN is 5.53% [22]. Our single ATR-CNN (4.96±0.08%) is better than a single MCDNN.

Fig. 3 shows top 10 errors (ordered by descending) made by our ATR-CNN for recognizing handwritten Chinese characters on ICDAR’13 Competition Dataset. The fist column in Fig. 3 is “ground-truth→prediction”. The ground-truth and prediction are indeed confusing in shape, particularly the fifth row. The presence of many cursive and confusing samples in the dataset makes it hard to further improve the recognition performance.

Fig. 4 respectively illustrates the contribution of relaxation convolution (blue curve with plus signs) and alternate training (red curve with circles) to recognition accuracy on validation dataset, comparing to our ICDAR’13 (green curve

Table IV  
RECOGNITION ACCURACY ON ICDAR’13 COMPETITION DATASET

<table>
<thead>
<tr>
<th>Method</th>
<th>Training Set</th>
<th>Error Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>THU [6]</td>
<td>CASIA-HWDB 1.0-1.1, 2.0-2.2</td>
<td>7.44</td>
</tr>
<tr>
<td>HIT [7]</td>
<td>CASIA-HWDB 1.0-1.1</td>
<td>7.38</td>
</tr>
<tr>
<td>Liu et al. [5]</td>
<td>CASIA-HWDB 1.0-1.1</td>
<td>7.28</td>
</tr>
<tr>
<td>Our ICDAR’13 [12]</td>
<td>CASIA-HWDB 1.1</td>
<td>5.23¹</td>
</tr>
<tr>
<td>MCDNN [22]</td>
<td>CASIA-HWDB 1.1</td>
<td>5.53</td>
</tr>
<tr>
<td>MCDNNs Voting [22]</td>
<td>CASIA-HWDB 1.1</td>
<td>4.35²</td>
</tr>
<tr>
<td>R-CNN</td>
<td>CASIA-HWDB 1.1</td>
<td>5.32±0.09</td>
</tr>
<tr>
<td>R-CNNs Voting</td>
<td>CASIA-HWDB 1.1</td>
<td>4.45³</td>
</tr>
<tr>
<td>ATR-CNN</td>
<td>CASIA-HWDB 1.1</td>
<td>4.96±0.08</td>
</tr>
<tr>
<td>ATR-CNNs Voting</td>
<td>CASIA-HWDB 1.1</td>
<td>3.94⁴</td>
</tr>
<tr>
<td>Human [12]</td>
<td>-</td>
<td>3.87</td>
</tr>
</tbody>
</table>

¹ The result is achieved by 4 CNNs voting. We took 1st place in ICDAR’13 Competition based on this result.  
² To make the results comparable, we report their error rate using 4 MCDNNs voting committee.  
³ This result is achieved by 4 R-CNNs voting.  
⁴ This result is achieved by 4 ATR-CNNs voting.
with triangles) as baseline. The vertical axis of Fig. 4 is the error rate on validation dataset (the subset of CASIA-HWDB 1.0) at each training epoch. Only initial 43 epochs are shown in Fig. 4, however the generalization capability of the three corresponding methods can be reflected by these 43 epochs. The ATR-CNN curve in Fig. 4 decreases slower during the first 10 epochs comparing to the other two curves, but its drop become sharp during the second 10 epochs. The R-CNN curve and ATR-CNN curve both decrease faster than Our ICDAR’13 curve does, indicating that relaxation convolution and alternate training both contribute to the improvement of recognition accuracy.

V. Conclusion

A handwritten character recognition method based on R-CNN and ATR-CNN have been presented in this paper. The differences between traditional CNN and our proposed R-CNN and ATR-CNN are as follows: (1) Our R-CNN consists of relaxation convolution layers whose neurons within a feature map do not share the same convolutional kernel as convolutional layers; (2) Our ATR-CNN further adopts alternate training strategy, i.e., the weights parameters of a certain layer do not change by back-propagation algorithm given a training epoch. Relaxation convolution can be considered to enhance the learning ability of the neural network, while alternate training is to regularize the network during training because relaxation connections sharply increase the number of parameters. Our ATR-CNN achieves state-of-the-art both on handwritten digit dataset MNIST and ICDAR’13 Competition Dataset. The handwritten digits misclassified by our ATR-CNN are even difficult for human observers.
REFERENCES


