Deep Neural Networks for Handwritten Chinese Character Recognition

Gendong Zhang, Xiao Zhang, Lantao Mei
Stanford University

Motivation

Chinese character recognition is a challenging task. Firstly, there are a lot more categories for Chinese characters than for digits and English characters. Secondly, Chinese handwritten to printed draft conversion also has really high applicable value. Therefore, precise recognition of Chinese character and sentence is a really worthwhile task.

Data Preprocessing

For Chinese character recognition, we use offline handwritten data from Institute of Automation of Chinese Academy of Sciences (CASIA). More specifically, HWDB1.1trn_gnt (5.3GB) as training set; HWDB1.1tst_gnt (1.4GB) as validation set and competition_gnt (1.4GB) as test set.

A python script is written to first decode the binary data in .gnt file using gb2312 which is the official character set of the People's Republic of China and then convert the binary data into .png image. Sample conversion result is shown in Figure 1:

Figure 1: Sample Chinese Character preprocessing result

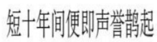

Figure 2: Synthetic Dataset from Swordsman

For Chinese Optical Character recognition, we generate our own dataset based on a Chinese Famous Novel, ***Swordsman***. We take 10 consecutive characters from the novel, and generate it as JPG file with labels. One example is shown in Figure 2.

Reference

[1] B. Shi, X. Bai, C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition", *CoRR*, vol. abs/1507.05717, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of CVPR*, pages 770–778, 2016.

Models

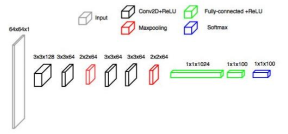

Figure 3: Simple CNN model with 6 weight layers

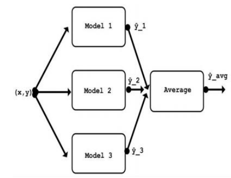

Figure 6: Models Ensembling

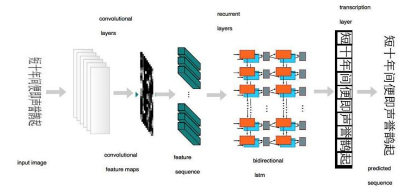

Figure 7: CRNN Model[1]

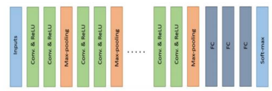

Figure 4: VGG16 Architecture

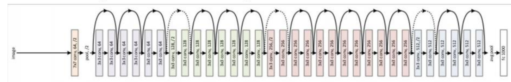

Figure 5: ResNet50 Architecture [2]

Results and Discussion

Best Performance Model——ResNet 50 on whole dataset

	ResNet 50
Training acc.	98.82%
Validation acc.	95.19%
Testing acc.	94.37%

Table 1: ResNet 50

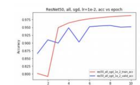

Figure 8: ResNet 50

ResNet50 parameter tuning——different optimizer & learning rate
(on a portion of whole dataset)

	ResNet 50 learning rate le-2 SGD	ResNet 50 learning rate le-2 Adam
Training acc.	99.44%	43.70%
Validation acc.	91.55%	40.94%
Testing acc.	87.31%	39.38%

Table 2: ResNet parameter tuning

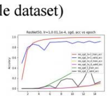

Figure 9: Different learning rate

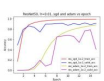

Figure 10: Different optimizer

Model comparison——ResNet 50 vs. VGG 16

	ResNet 50	VGG 16
Training acc.	99.44%	81.14%
Validation acc.	91.55%	70.79%
Testing acc.	87.31%	72.66%

Table 3: ResNet vs VGG

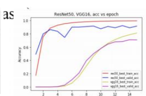

Figure 11: Different model

Preliminary test—— VGG 16 vs. CNN
(on even smaller portion of dataset)

	VGG 16	CNN
Training acc.	97.05%	95.11%
Validation acc.	93.69%	89.11%
Testing acc.	94.48%	88.68%

Table 4: VGG vs. CNN

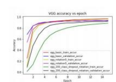

Figure12: VGG

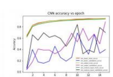

Figure 13: CNN

Analysis

After training on a small dataset, we find ResNet50 outperforms VGG16 and CNN. And with 0.01 learning rate and sgd optimizer, resnet50 performs the best.

The oscillation in the validation error can be caused by the dropout or other overfitting-proof methods, while the deeper the network, the less oscillation

OCR

The ocr CRNN model does not work well on our synthetic dataset, even using pre-trained weight. we only got 25.78% accuracy for testing.

1.More suitable training dataset

2.Use character segmentation and use our best character recognition model