Image Captcha Recognition

Progress Report

Xiaoguang Wang
A53246911
xiw503@ ucsd.edu

Hao Zhang
A53234732
haz351@ ucsd.edu

Qian Wang
A53235276
qiw018@ ucsd.edu

Abstract

At this point, we have built an applicable network for 4-digits captcha image recognition with TensorFlow and built the captcha image dataset to train and test the network. Below are what we have achieved so far and what left to be improved on this system.

1   Introduction to the network

The basic idea here is we employed a pre-trained deep neural network to learn features from the captcha images, then combined with *softmax* layers to makes predictions for each digit of the captcha code separately. Below is the network structure for our system:
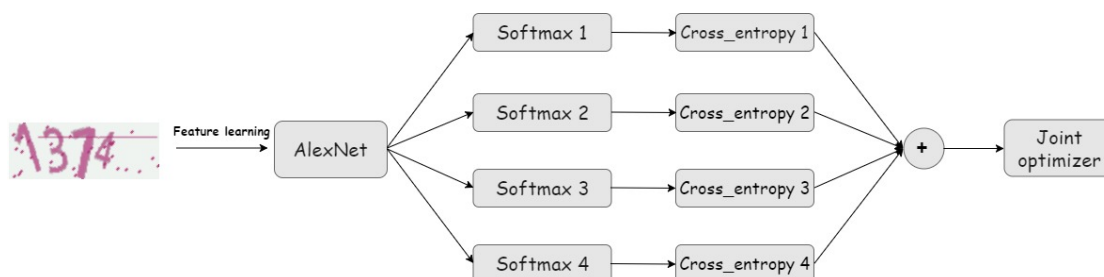

Figure 1: Network structure for the recognition system (4-digits captcha)

Below are some details for the network:

1.1   Image feature learning

Here we used the pre-trained AlexNet for image feature learning since it has been well trained on a large image dataset and proved to represent images features well with lower dimension feature vectors. We leave the net parameters untouched during the training process.

Also, we explored some more modern networks like GoogLeNet, but unluckily, the GPU server seems out of resources to run this large network well. So, we chose AlexNet at last and believe this network can extract necessary features well from the simple captcha images.

1.2 Training

As is shown in the network structure above, we used one *softmax* layer to predict each digit separately (here we set the captcha length to be 4, so we implemented 4 softmax layers). To train this model, we use a method called **multi-task learning**.

On each softmax layer, we have an output and get a cross-entropy loss value. Instead of backpropagating these loss values four times separately, we backpropagate the sum or the mean value of these loss values and assume that we can get the best prediction if we can minimize this 'complex loss value' through some methods like gradient descent.

2 Some preliminary results and conclusions

At this moment, we only used digits from 0 to 9 to compose a 4-digits captcha image. Below are the accuracies on test set and mean loss value on training set we get during the training process.
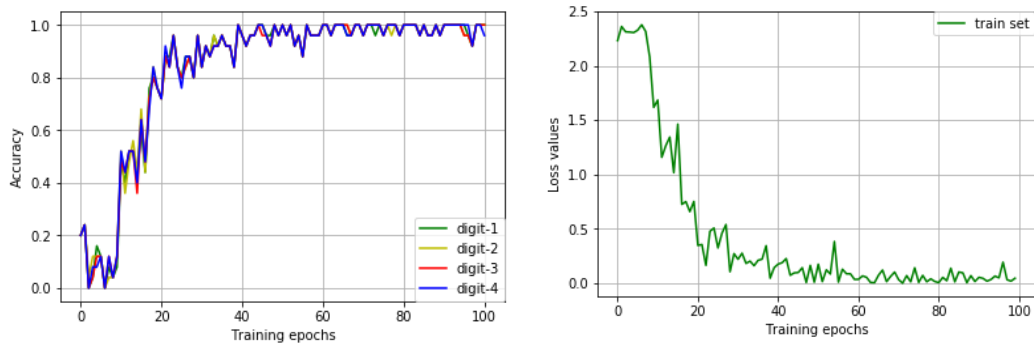

Figure 2: Accuracies and loss value during the training process

We can see, during the training process, accuracies on each digit keeps going up and mean loss value on training set keeps going down. We have achieved a satisfactory accuracy on 4-digits captcha image recognition. This result can prove the feasibility of our model and the training method we are using now.

3 Some difficulties and potential improvements

Now we have realized the initial implementation of our captcha image system, there still left with some difficulties and we will still have to make some improvement on this system.

3.1 Robustness

Now the captcha is simple and consist od only digit from 0 to 9. We can make the system more robust based on the below two aspects:

(1) **Complexity of the code**: We are going to explore a larger char set including digits, lower-case and upper-case characters to enrich the captcha image set.

(2) **Noise**: We will try to add more noise to the captcha images and potentially to introduce some noise cancellation operations from traditional CV like Gaussian Low Pass Filter before feeding the images into CNN to compensate for the noise.

3.2 Some improvements to make the system more user-friendly

This part is just some naïve thoughts about the system and definitely not the main focus of this course and the project. Captcha is an acronym for **"C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part". So, if time permitted, we can store the well-trained network model and combined with some web crawlers to improve their performance, since they cannot get access to some web sites with captcha protection.

Github Link：https://github.com/wxgsdy/Captcha_Recognition