Dealing with the HEp-2 Cell Image Classification Based on Deep CNN .

Ting Li (litingszy@qq.com)

1 HEp-2 Cell Image Classification

1.1 Construction of the network

In this assignment, a deep CNN is constructed to deal with the HEp-2 Cell Image Classification. I use the TensorFlow library, and implement it using python. Because of the huge computational complexity, I run the code on Colaboratory which is provided by google. The GPU on it can be used to speed up the process of training.

This CNN contains 8 layers: 3 convolutional layers alternated with 3 pooling layers, and the final two are full-connected layers.

C1(the first convolutional layer) contains 6 feature maps which have size of 72×72, C2(the second convolutional layer) has 16 features which has size of 33×33, and C3(the third convolutional layer) has 32 9×9 feature maps. And the final two layers are all full-connected, the first one has 150 feature maps and the final one(is also the output layer) has 6 features which also means 6 types of cells.

The network structure can be shown as follows:

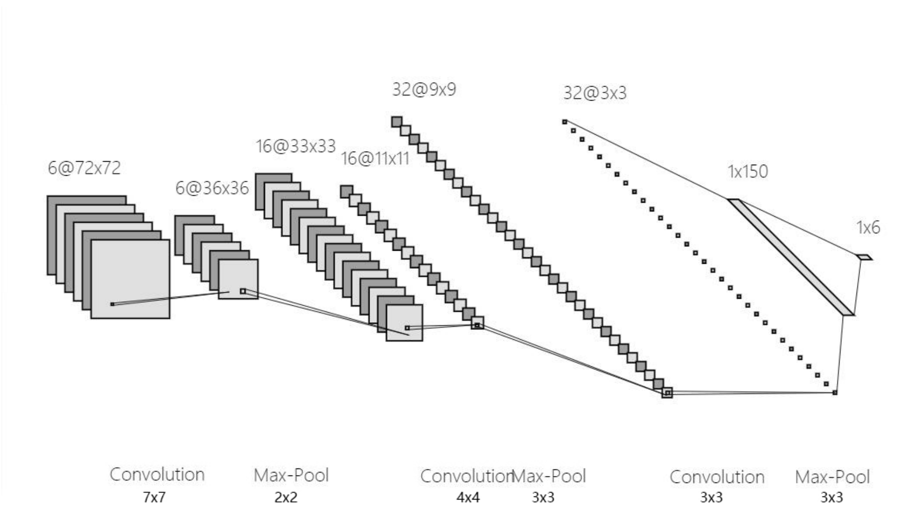

Figure 1: The structure of CNN

1.2 Image data preprocessing

In this task, the image preprocessing mainly contains two steps:

Fiestly, because the brightness and contrast of HEp-2 cell images differ greatly. In order to reduce this difference and enhance the contrast, each image is normalized.

Secondly, in order to increase the accuracy, more different examples are needed, so here those images are rotated by three different angles. Therefore training examples are four times as many as before to achieve better performance.

1.3 Parameters and accuracy

In this assignment, the main parameters which affect the performance are learning rate and batch size. These two parameters should chose properly.

Then batch size should be a proper value. Its value should not be too small, because accuracy will be unstable and fluctuate greatly. At the same time, it can not be too big, otherwise accuracy will increase too slowly, it needs more epoch to achieve enough accuracy. This conclusion has been obtained in the experiment I did before in the logistic regression model and CNN. And the professor's research also demonstrates it. Therefore, the batch size in this task is 120.

And from the professor's journal paper, the learning rate should not be too small or too large, when it is small, the test accuracy will increase quite slowly. While it is too large, the accuracy will unstable then we cannot get ideal result. So here I will try two learning rates: 0.01 and 0.001. And here is the experiment result:

	Learning Rate	
Value	0.01	0.001
Training Accuracy	0.8543	0.9587
Test Accuracy	0.8549	0.9116
Epoch	120	70

Table 1: Accuracy under different learning rate

Clearly, here learning rate can set as 0.001 because it has better performance in both two accuracy, and it can have great classification result. The detailed result is shown bellow:

```
Epoch 40
Training accuracy:  0.9425287324806739
Test accuracy:  0.9025000002649095
Epoch 45
Training accuracy:  0.947068963790762
Test accuracy:  0.9072222226195865
Epoch 50
Training accuracy:  0.9524137891572098
Test accuracy:  0.9088888883590698
Epoch 55
Training accuracy:  0.9508908033370972
Test accuracy:  0.903981484969457
Epoch 60
Training accuracy:  0.9554022978092062
Test accuracy:  0.9089814835124546
Epoch 65
Training accuracy:  0.9566091940320771
Test accuracy:  0.9055555548932818
Epoch 70
Training accuracy:  0.9586781592204653
Test accuracy:  0.9115740729702844
```

Figure 2: Accuracy under 0.001 learning rate

In this experiment, the train set has 34804 examples, test set has 10880 examples and validation set has 8700 examples. The number of epochs is 100, learning rate is 0.001, batch size is 70, and the result shows every 5th epoch. The accuracy figure can be shown as follows:

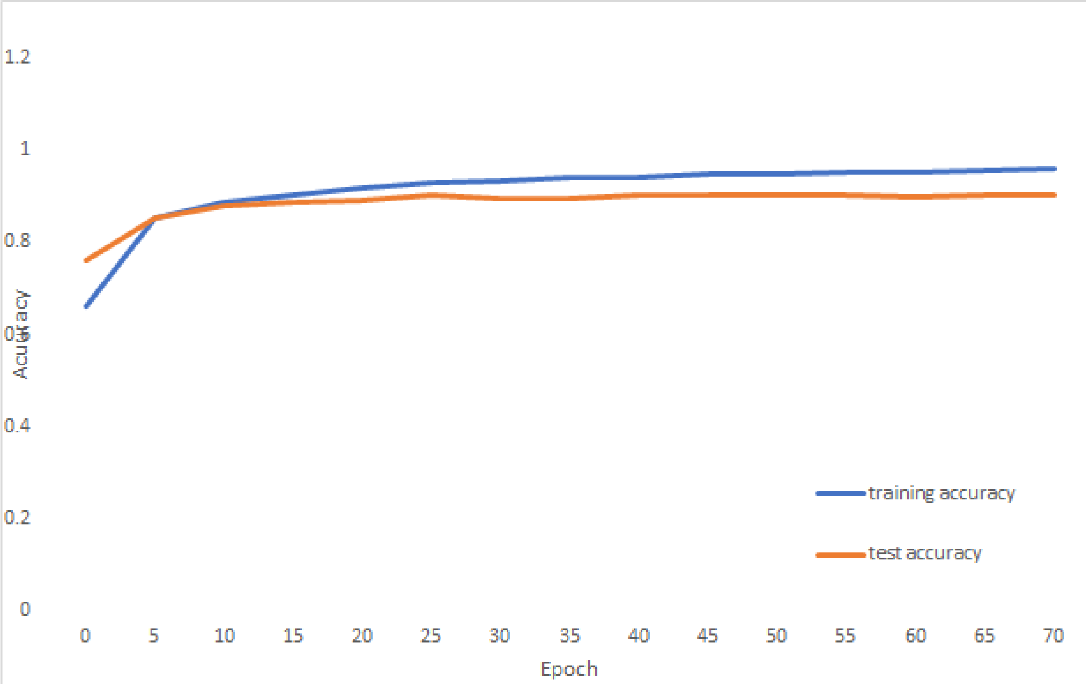

Figure 3: Training and test accuracy

As shown above, firstly, the training accuracy is lower than test accuracy. Then training accuracy exceed the test accuracy after 5 epochs. And finally, they all tend to be stable. Then the test accuracy keep lower than training accuracy, but it reaches 91.16% in the 70th epoch, which can be a great result.

1.4 Observation and analysis

During this assignment, I tried two experiment with different learning rate, the first time I set learning rate as 0.01 which has the final classification accuracy of 85.50%, when learning rate is 0.001, the final classification accuracy reach to 90.55%, which is higher than the first experiment.

However, this result is a bit different from some research. According to some articles, the learning rate should not be so small. And it is shown in some articles, when the learning rate is 0.01, it has better performance. But in this experiment, it is clear that 0.001 achieve higher accuracy, the reason may be: When the learning rate is 0.001, it fits well with other parameters, so it can get better results.

In conclusion, CNN is very suitable for dealing with HEp-2 Cell Image Classification, as it has great performance. Thanks to professor's research, it helps me a lot in constructing CNN structures, reprocessing images and choosing suitable parameters.