2017-11-1

Homework 2

By yangfan sun

1. Introduction

Implement Text classification with CNN model to train a new dataset into 10 different classes.

1. Objectives

Dataset downloaded from a set group of THUCTC. We only applied 10 classes among them and each classes include 6,500 samples.

* cnew.train.txt: training set (50,000 items).
* cnew.val.txt: validation set (5,000 items).
* cnew.test.txt: testing set (10,000 items).

Pre-conditioning dataset, data/cnew\_loader.py is the program pre-treat the three sets of data.

* Read\_file(): read the txt files.
* Build\_vocab(): Build the vocabulary sheet.
* Read\_vocab(): Read the sheet above, transform into dictionary: {word: id}.
* Read\_category(): Transform into dictionary: {class: id}.
* To\_words(): Transform data represented by id into words.
* Process\_file(): Transform words into a fixed length id.
* Batch\_iter(): For CNN training batch size.

Classify all samples into 10 different classes, including “Sport, Finance, Real estate, Living, Education, Technology, Fashion, Politics, Game, Entertainment”.

1. Approaches

Spilt dataset into training, validation and test sets. Fill in the CNN machine, including an embedding layer, a convolution layer, a max pooling layer and a full connection layer, applying drop-out method. Relu activation function with full connection layer and a softmax as classifier at last to compare prediction value and labels in dataset. Loss function is cross entropy and adam optimizer for adjusting weights.

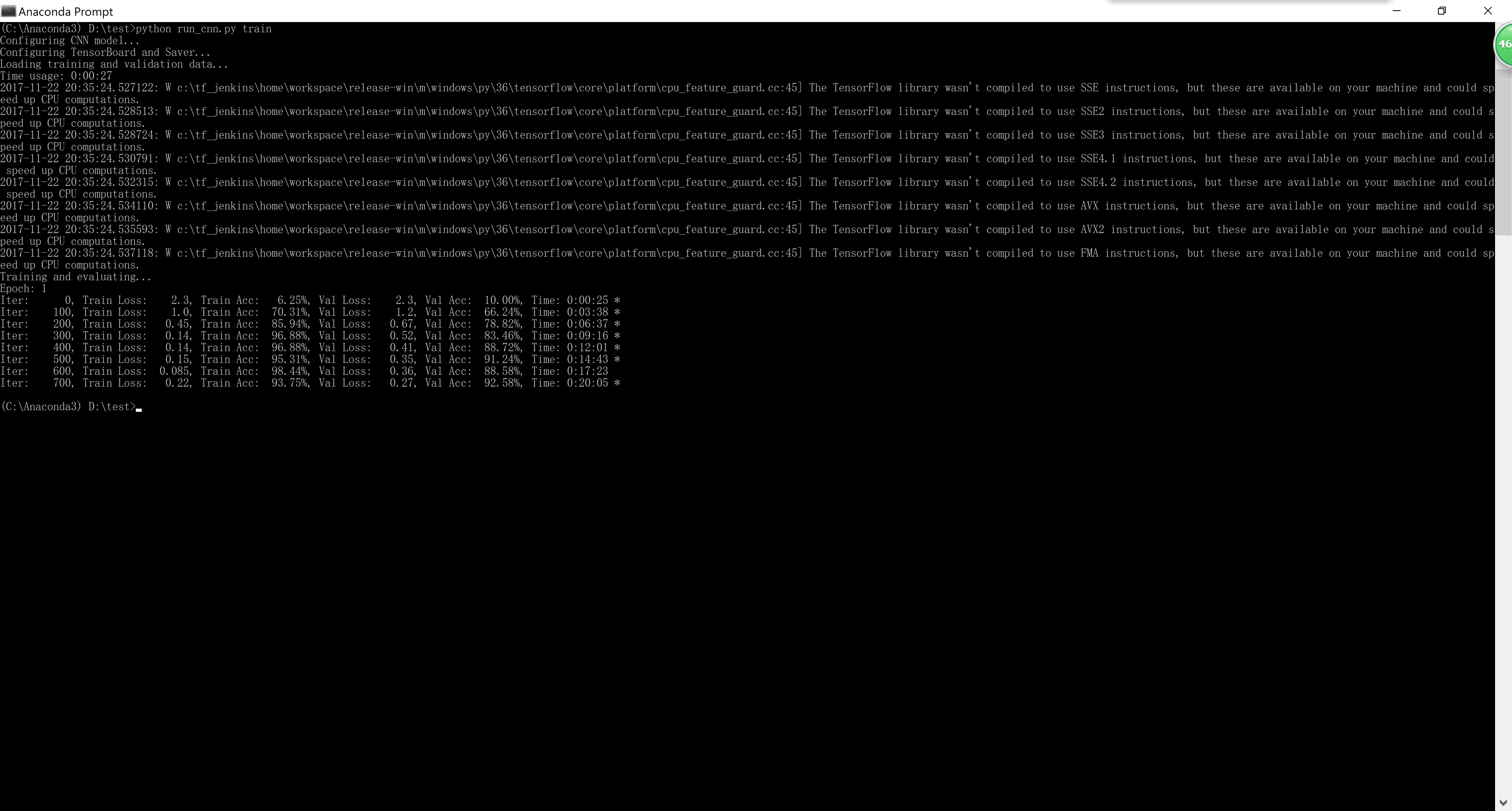
1. Parameters

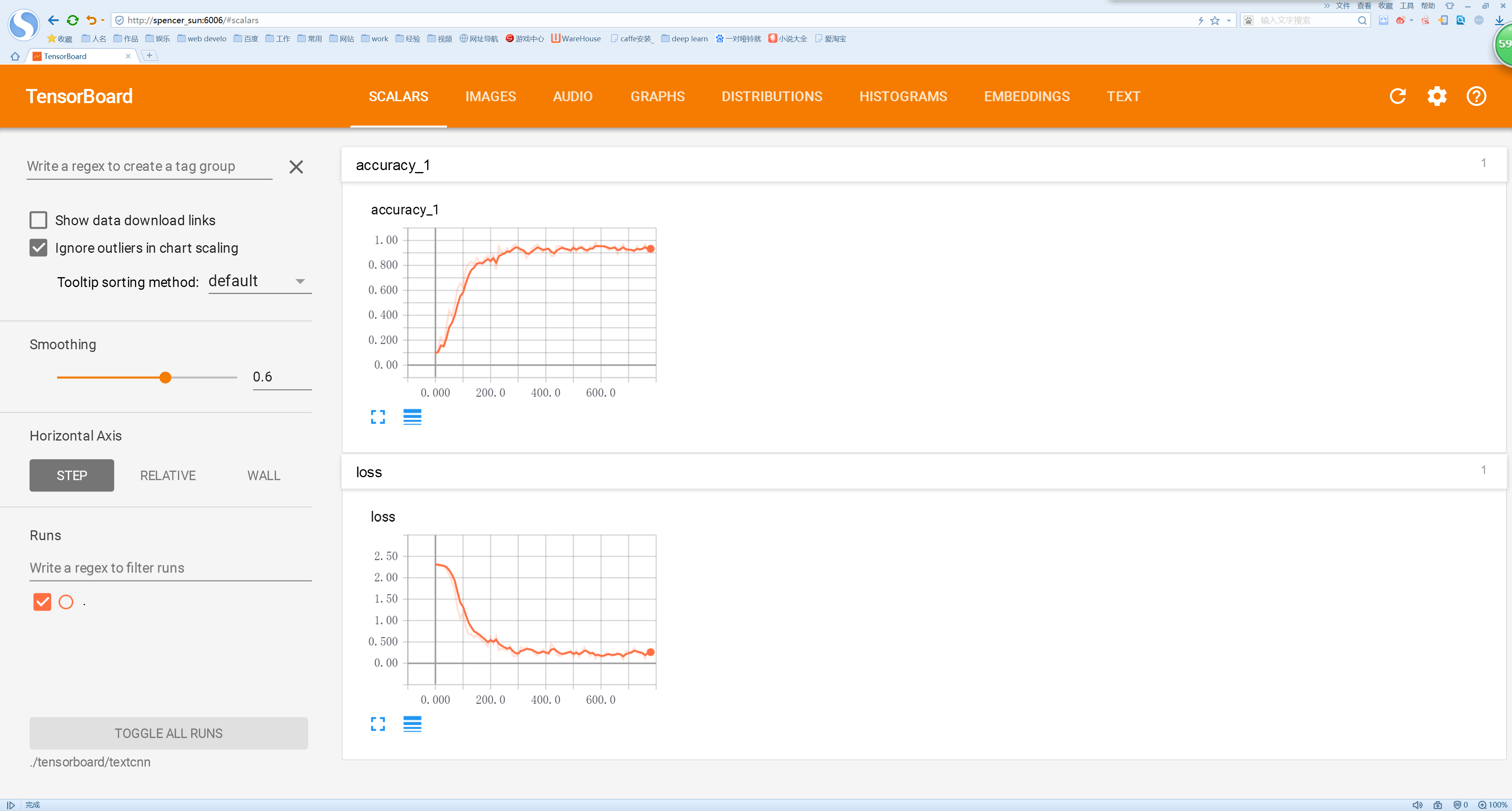
CNN configuration parameters is in cnn\_model.py.

* embedding\_dim = 64 # dimension of embedding layer
* seq\_length = 600 # length of sequence
* num\_classes = 10 # number of classes
* num\_filters = 256 # number of filters
* kernel\_size = 5 # kernel size
* vocab\_size = 5000 # size of vocabulary sheet
* hidden\_dim = 128 # dimension of hidden layer
* dropout\_keep\_prob = 0.5 # probability of dropout
* learning\_rate = 1e-3 # learning rate
* batch\_size = 64 # batch size
* num\_epochs = 10 # number of epoch
* print\_per\_batch = 100 # print results after running number of batches
* save\_per\_batch = 10 # save in tensorboard after running number of batches

1. Evaluation & Discussion

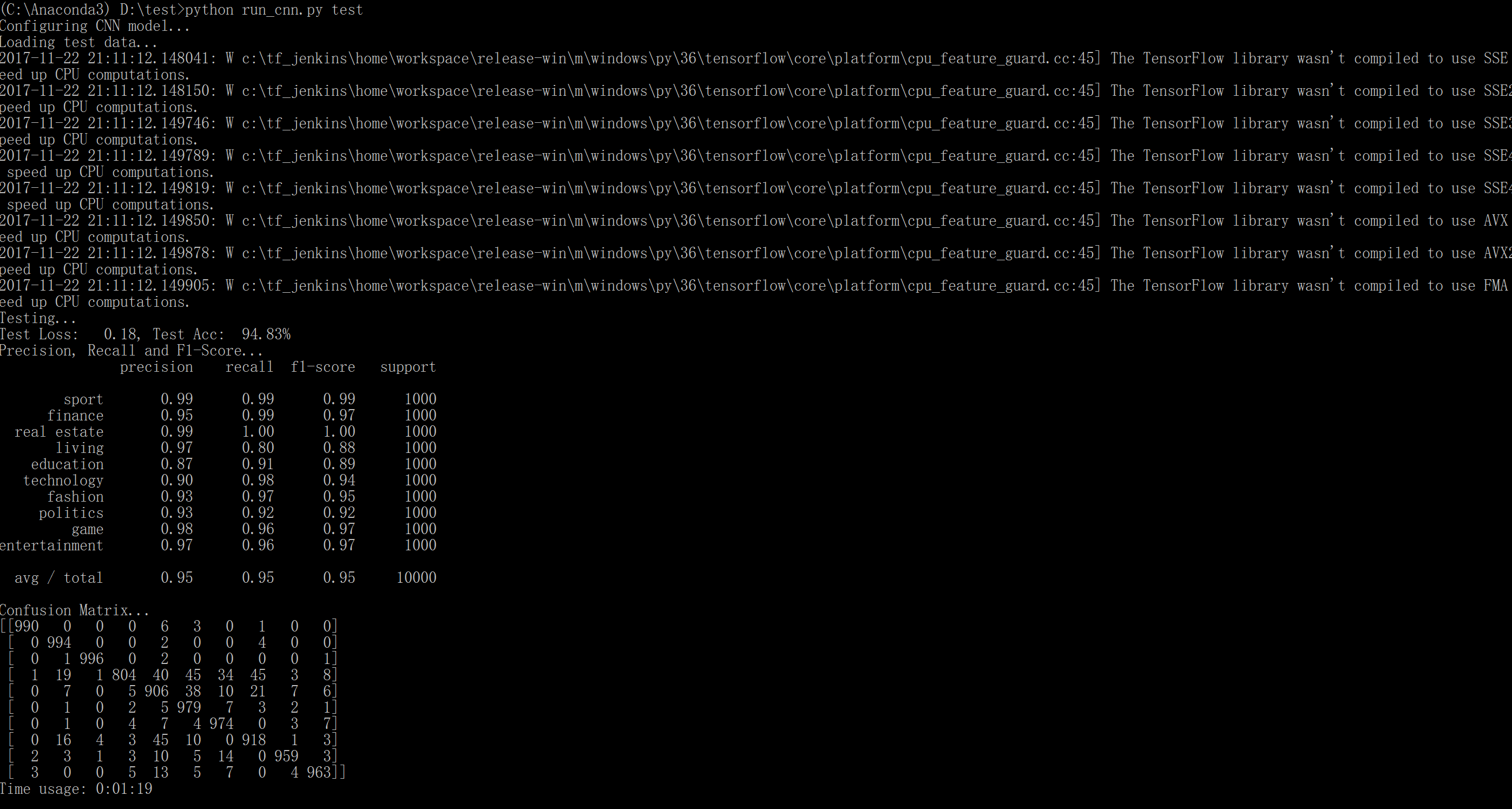
Training:





1. Conclusion

Testing:



The result of test set is 94.83% with only 1 epochs training, and the average of precision, recall and f1-score as followed, at high accuracy. The confusion matrix also can show the same conclusion.