CS5590 APS - Deep Learning Programming

ASSIGNMENT 2

Abdoh Jabbari

Introduction

In deep learning CNN is models has inputs of layers to start the processing to outputs layers. In between those two steps there are also hidden layers that are convolutional layers such as pooling and fully connected layers. In this lab we are using **Convolution Neural Networks** CNN for text file classification.

Objectives

In this lab we are going to use TensorFlow with CNN model to analyze text file. This method is helpful to use in Natural Language Processing NLP to analyze sentences to come up good results. Our objective is to use multiclass datasets to analysis the sentences in that file to get to higher accuracy and less loss.

Approaches/Methods

The approach used in this lab is using TensorFlow with CNN model. The method is using multiclass with word embedding.

- loading the datasets
- Read the datasets and analyze it.
- Determining the number of features, and classes on the datasets.
- Defining variables and placeholders.
- Setting the parameters of the model.
- Training and testing the model.
- Optimizing to minimize the loss or error
- Optimizing to increase the accuracy

Datasets

The dataset used in this project is Kaggle Consumer Finance Complaints. The datasets are multiclass data

It has the following columns:

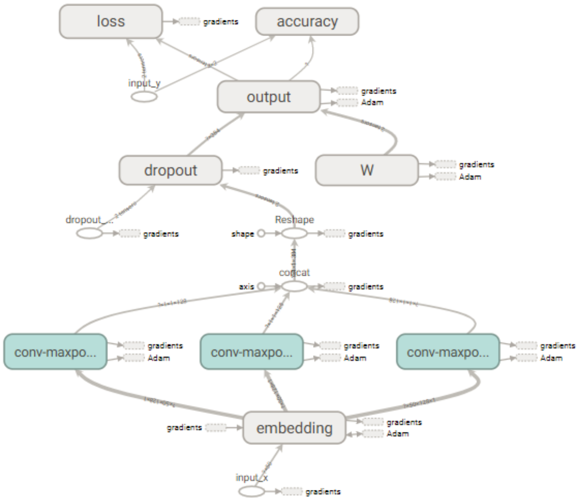
- date_received - date the complaint was received
- product
- sub_product
- issue
- sub_issue
- consumer_complaint_narrative
- company_public_response
- company
- state
- zipcode
- tags
- consumer_consent_provided
- submitted_via
- date_sent_to_company
- company_response_to_consumer
- timely_response
- consumer_disputed
- complaint_id

The data are very large that made my computer to crash multiple time, therefore I had to delete some of the data to allow the CNN model to analysis the datasets without crashing my system. Now, it takes around 15 minutes to finish the analysis.

Workflow

The workflow of this CNN model in steps:

- Input
- placeholders
- Setup the parameters
- Embedding layers
- Conv and MaxPo Layers
- Dropout
- Output
- Loss and accuracy

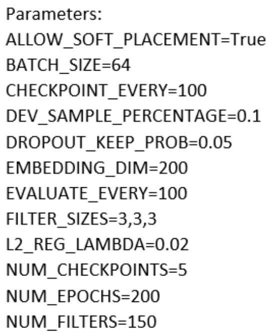


Parameters

```
BATCH_SIZE=64
CHECKPOINT_EVERY=100
DEV_SAMPLE_PERCENTAGE=0.1
DROPOUT_KEEP_PROB=0.5
EMBEDDING_DIM=128
EVALUATE_EVERY=100
FILTER_SIZES=3,4,5
L2_REG_LAMBDA=0.0
NUM_CHECKPOINTS=5
NUM_EPOCHS=200
NUM_FILTERS=128
```

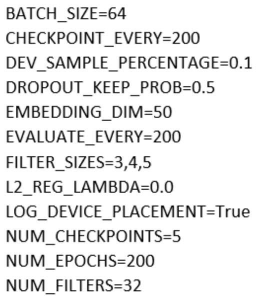
Evaluation & Discussion

First results after deleting more than ¾ of the datasets to allow my system to run **loss 0.783131, accuracy 0.7**



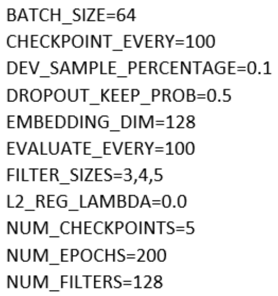
```
Parameters:
ALLOW_SOFT_PLACEMENT=True
BATCH_SIZE=64
CHECKPOINT_EVERY=100
DEV_SAMPLE_PERCENTAGE=0.1
DROPOUT_KEEP_PROB=0.05
EMBEDDING_DIM=200
EVALUATE_EVERY=100
FILTER_SIZES=3,3,3
L2_REG_LAMBDA=0.02
NUM_CHECKPOINTS=5
NUM_EPOCHS=200
NUM_FILTERS=150
```

Second results are **losses 0.29969, accuracy 0.89**



```
BATCH_SIZE=64
CHECKPOINT_EVERY=200
DEV_SAMPLE_PERCENTAGE=0.1
DROPOUT_KEEP_PROB=0.5
EMBEDDING_DIM=50
EVALUATE_EVERY=200
FILTER_SIZES=3,4,5
L2_REG_LAMBDA=0.0
LOG_DEVICE_PLACEMENT=True
NUM_CHECKPOINTS=5
NUM_EPOCHS=200
NUM_FILTERS=32
```

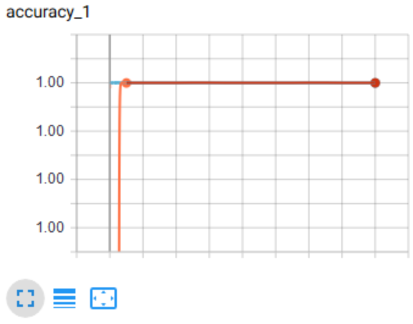
Third trial we get to run through 1600 steps, to get to **losses 0, accuracy 1**

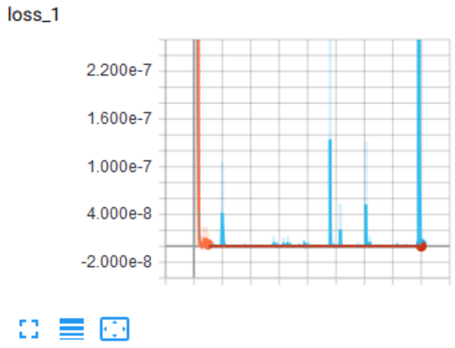


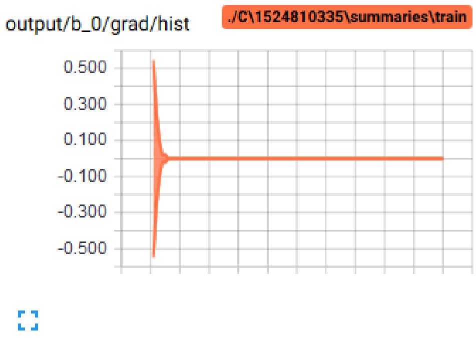
```
BATCH_SIZE=64
CHECKPOINT_EVERY=100
DEV_SAMPLE_PERCENTAGE=0.1
DROPOUT_KEEP_PROB=0.5
EMBEDDING_DIM=128
EVALUATE_EVERY=100
FILTER_SIZES=3,4,5
L2_REG_LAMBDA=0.0
NUM_CHECKPOINTS=5
NUM_EPOCHS=200
NUM_FILTERS=128
```

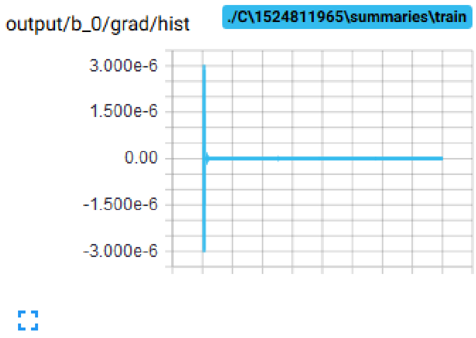
Conclusion

After multiple trial we get to good accuracy, the large file and the model to analysis text in the file. At first it took more than half hour without finishing the analysis, then I stopped the code from executing. Then, after reducing the datasets the system was ok to handle the code execution to run the model with CPU reaching 99% utilization.









References

http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/

https://github.com/dennybritz/cnn-text-classification-tf/tree/master/data/rt-polaritydata

https://github.com/jiegzhan/multi-class-text-classification-cnn

https://towardsdatascience.com/how-to-do-text-classification-using-tensorflow-word-embeddings-and-cnn-edae13b3e575