Deep Learning
Lab Assignment – 2

Introduction:

Here we are implementing CNN with the help of a few datasets named "battles.csv, character-deaths.csv, character-predictions.csv" for text classification in which the data is about all the battles in Game of Thrones. Basically CNN consists of multiple convolutional layers. The CNN is designed to analyze 2D structure of an input image. The CNN is better to train as they have many fewer parameters than fully connected networks which comes as an advantage here.

Objectives:

Here we are using Game of thrones datasets for implementation of text classification on CNN through which the graphs are plotted using tensor flow and then we also need to evaluate the changes by modifying some hyper parameters and find out the accuracy and loss.

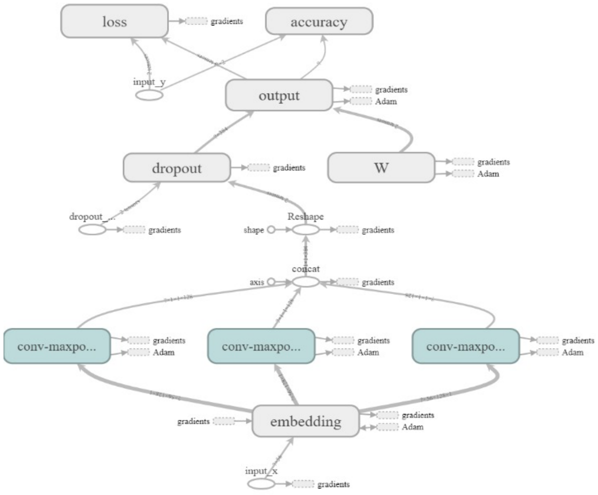
Approaches/Methods:

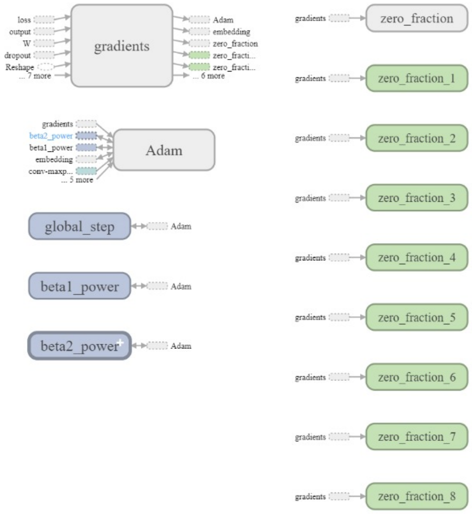
- Consider the following datasets namely "battles.csv, character-deaths.csv, character-predictions.csv
- We are using Adam optimizer in order to construct the model
- Initially the data from the dataset is read and build model
- Then the data is converted into json format and create placeholders for X,Y
- Then split the data into two parts train data and test data
- The train data and test data need to be in the ratio 20:80
- Now store the summary of the data session into a folder

Workflow:

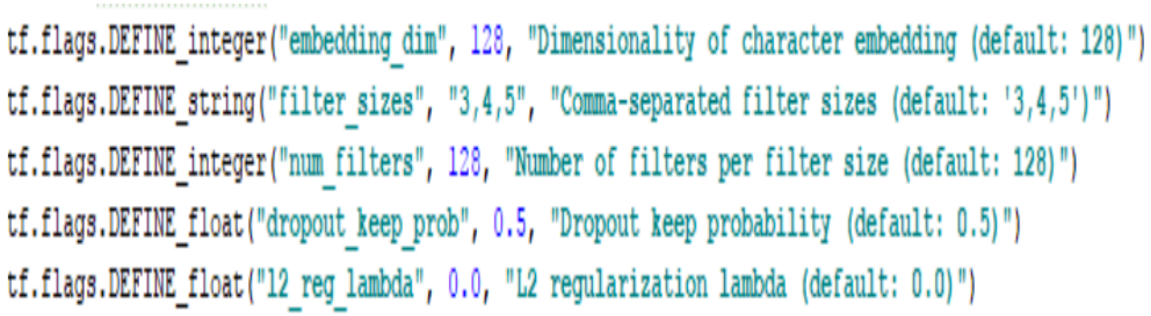
- Consider the datasets through which we perform the text classification over CNN.
- Modify some of the hyper parameters and identify the accuracy and loss

The workflow is as follows





Model Hyper Parameters are



```
tf.flags.DEFINE_integer("embedding_dim", 128, "Dimensionality of character embedding (default: 128)")
tf.flags.DEFINE_string("filter_sizes", "3,4,5", "Comma-separated filter sizes (default: '3,4,5')")
tf.flags.DEFINE_integer("num_filters", 128, "Number of filters per filter size (default: 128)")
tf.flags.DEFINE_float("dropout_keep_prob", 0.5, "Dropout keep probability (default: 0.5)")
tf.flags.DEFINE_float("l2_reg_lambda", 0.0, "L2 regularization lambda (default: 0.0)")
```

Configuration:

- PyCharm Editor 3.4
- Tensor Board

DataSets:

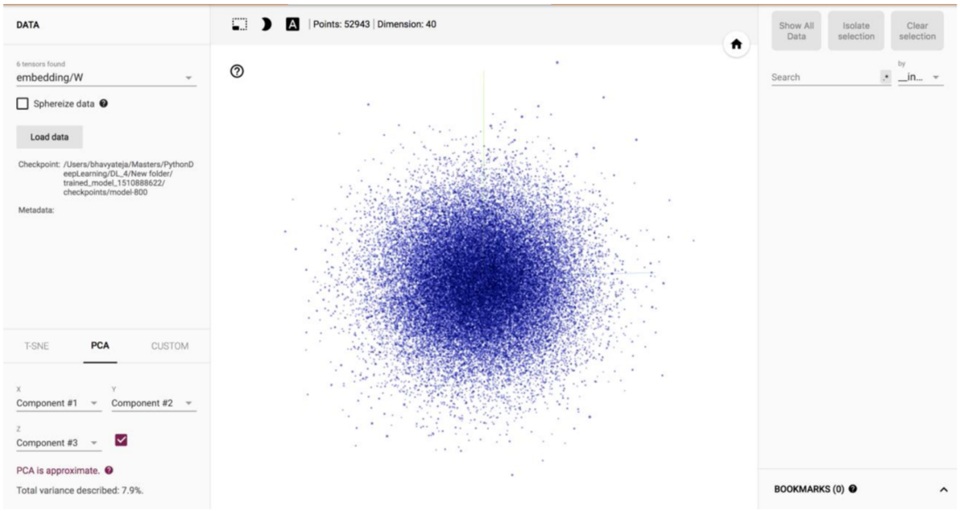
We considered Game of Thrones dataset which has different csv files.

Parameters:

We considered the parameters no.of epochs, learning rate and so on in order identify the accuracy and loss.

Evaluation and Discussion:

The built model using CNN can be viewed using the tensor board by considering the user data. Then by using an python file named eval.py we can use the data that is only needed and we can exclude the remaining data



Points: 52943 | Dimension: 40

6 tensors found
embedding/W

□ Sphereize data

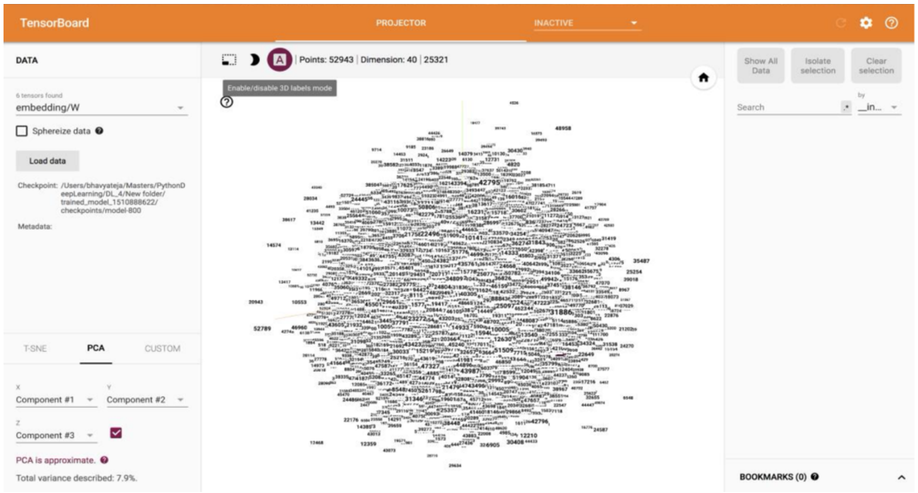
Checkpoint: /Users/bhavyateja/Masters/PythonDeepLearning/DL_4/New folder/trained_model_1510888622/checkpoints/model-800

Metadata:

X Component #1
Y Component #2
Z Component #3 ☑

PCA is approximate.

Total variance described: 7.9%.



A | Points: 52943 | Dimension: 40 | 25321

6 tensors found
embedding/W

□ Sphereize data

Checkpoint: /Users/bhavyateja/Masters/PythonDeepLearning/DL_4/New folder/trained_model_1510888622/checkpoints/model-800

Metadata:

X Component #1
Y Component #2
Z Component #3 ☑

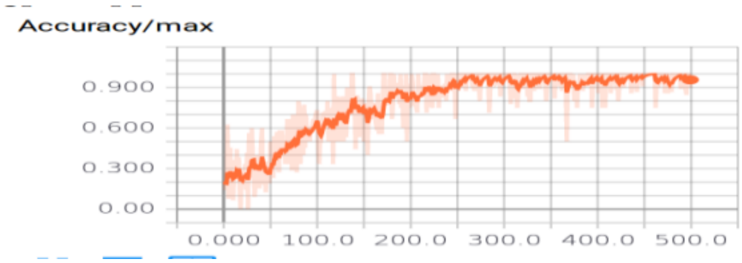
PCA is approximate.

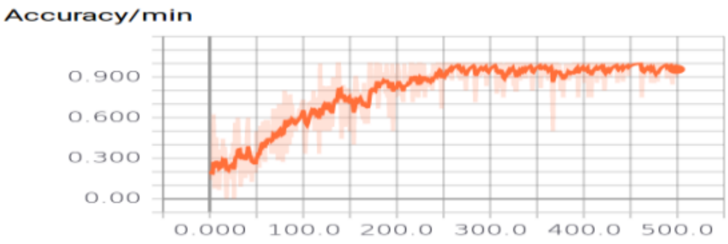
Total variance described: 7.9%.

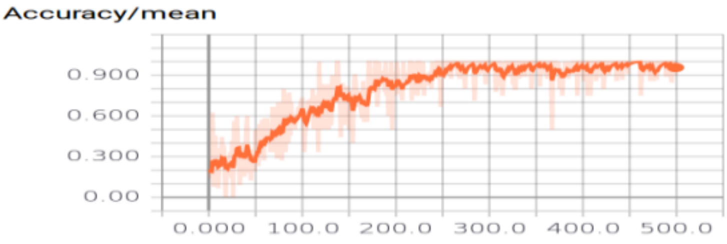
Search __in...

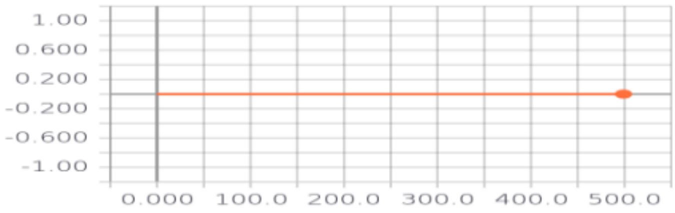
Conclusion:

By using this training set we obtained better results. But we can observe better accuracy results by using RNN than CNN.









Accuracy/stddev_1