CSEE 5590 - Special Topics
Deep Learning – Lab 2

Author:
Kalyan Kilaru
Class Id: 22

Configuration:
IDE: pycharm
python: version 3

Introduction: By using the CNN model, we need to implement the text classification with the new data sets.

Objective: The main objective is to make us familiar with TensorFlow and by implementing the Test classification with the CNN model.

Approaches: To achieve Text classification with CNN Model, I used 5 new data sets. Then I have written the code that accepts the data sets after which I run it to get the accuracy and the loss. Then I checked the accuracy and the loss continuously by changing the hyper-Parameters, i.e., to know how these factors impact the loss and accuracy.

Workflow:

- To get the 5 different data sets.
- To write the code that accepts the data or the input data files.
- To train the system or model with the new data sets.
- Then we need to evaluate the model by checking the accuracy and loss.
- To modify the factors and check how it effects loss and accuracy.

Datasets: The datasets I used are as follows
fashion_7000, finance_7000, law_7000, lifestyle_7000 and test_data these are the data sets given in the class.

Parameters:

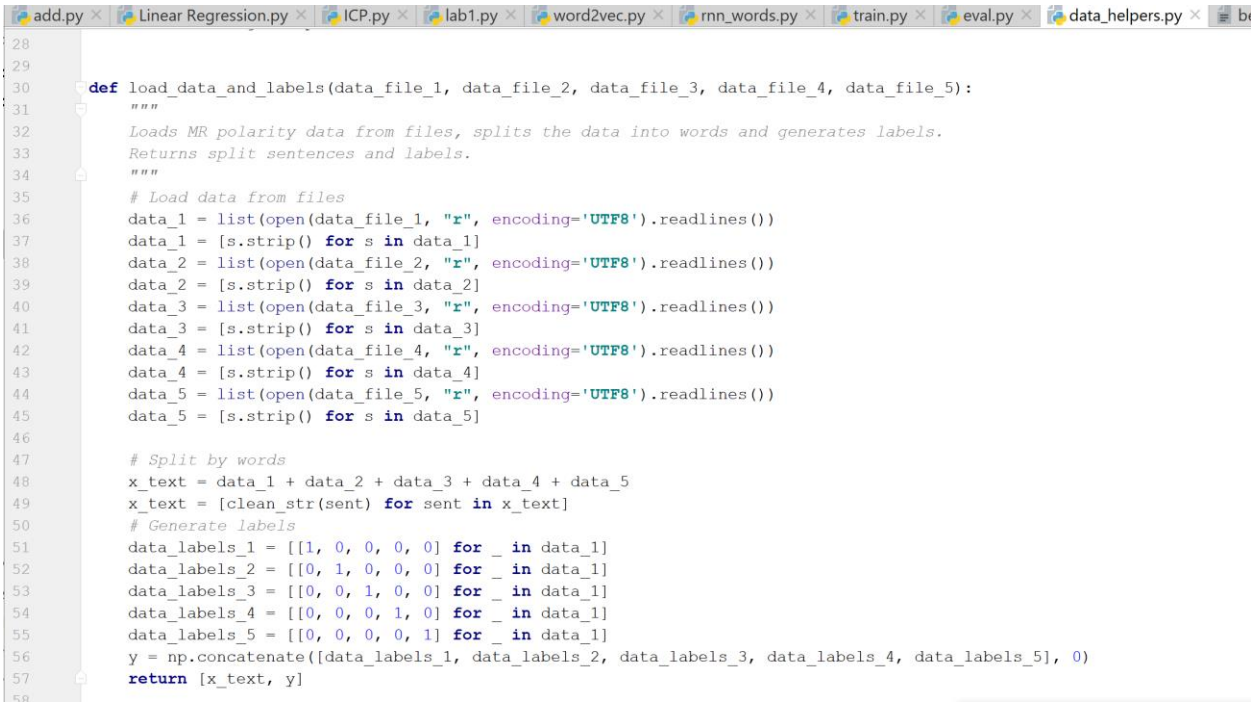
- Initially I have taken the embedding_dim to the default value of 128 and later I made it to 64 to check the accuracy and loss.

- The default num_filter is taken as 128 and then it is reduced to half to check how it effects the accuracy and loss.

- To evaluate Model on Dev set after number of steps, the default is 100. Also checked with 200 steps.

Evaluation:

I have modified the code of data_helpers.py by giving the new data sets and the below code accepts the new data.



```
def load_data_and_labels(data_file_1, data_file_2, data_file_3, data_file_4, data_file_5):
    """
    Loads MR polarity data from files, splits the data into words and generates labels.
    Returns split sentences and labels.
    """
    # Load data from files
    data_1 = list(open(data_file_1, "r", encoding='UTF8').readlines())
    data_1 = [s.strip() for s in data_1]
    data_2 = list(open(data_file_2, "r", encoding='UTF8').readlines())
    data_2 = [s.strip() for s in data_2]
    data_3 = list(open(data_file_3, "r", encoding='UTF8').readlines())
    data_3 = [s.strip() for s in data_3]
    data_4 = list(open(data_file_4, "r", encoding='UTF8').readlines())
    data_4 = [s.strip() for s in data_4]
    data_5 = list(open(data_file_5, "r", encoding='UTF8').readlines())
    data_5 = [s.strip() for s in data_5]

    # Split by words
    x_text = data_1 + data_2 + data_3 + data_4 + data_5
    x_text = [clean_str(sent) for sent in x_text]
    # Generate labels
    data_labels_1 = [[1, 0, 0, 0, 0] for _ in data_1]
    data_labels_2 = [[0, 1, 0, 0, 0] for _ in data_1]
    data_labels_3 = [[0, 0, 1, 0, 0] for _ in data_1]
    data_labels_4 = [[0, 0, 0, 1, 0] for _ in data_1]
    data_labels_5 = [[0, 0, 0, 0, 1] for _ in data_1]
    y = np.concatenate([data_labels_1, data_labels_2, data_labels_3, data_labels_4, data_labels_5], 0)
    return [x_text, y]
```

The data loading parameters are given as below that takes 5 new data sets.

```
tf.flags.DEFINE_string("fashion1_7000", "./ICP_data/fashion_7000.txt",
"fashion_7000.txt")
tf.flags.DEFINE_string("finance2_7000", "./ICP_data/finance_7000.txt",
"finance_7000.txt")
tf.flags.DEFINE_string("law3_7000", "./ICP_data/law_7000.txt", "law_7000.txt")
tf.flags.DEFINE_string("lifestyle4_7000", "./ICP_data/lifestyle_7000.txt",
"lifestyle_7000.txt")
tf.flags.DEFINE_string("TestData5", "./ICP_data/TestData", "TestData")
```

The data is then loaded or passed to the data_helpers class as shown below.

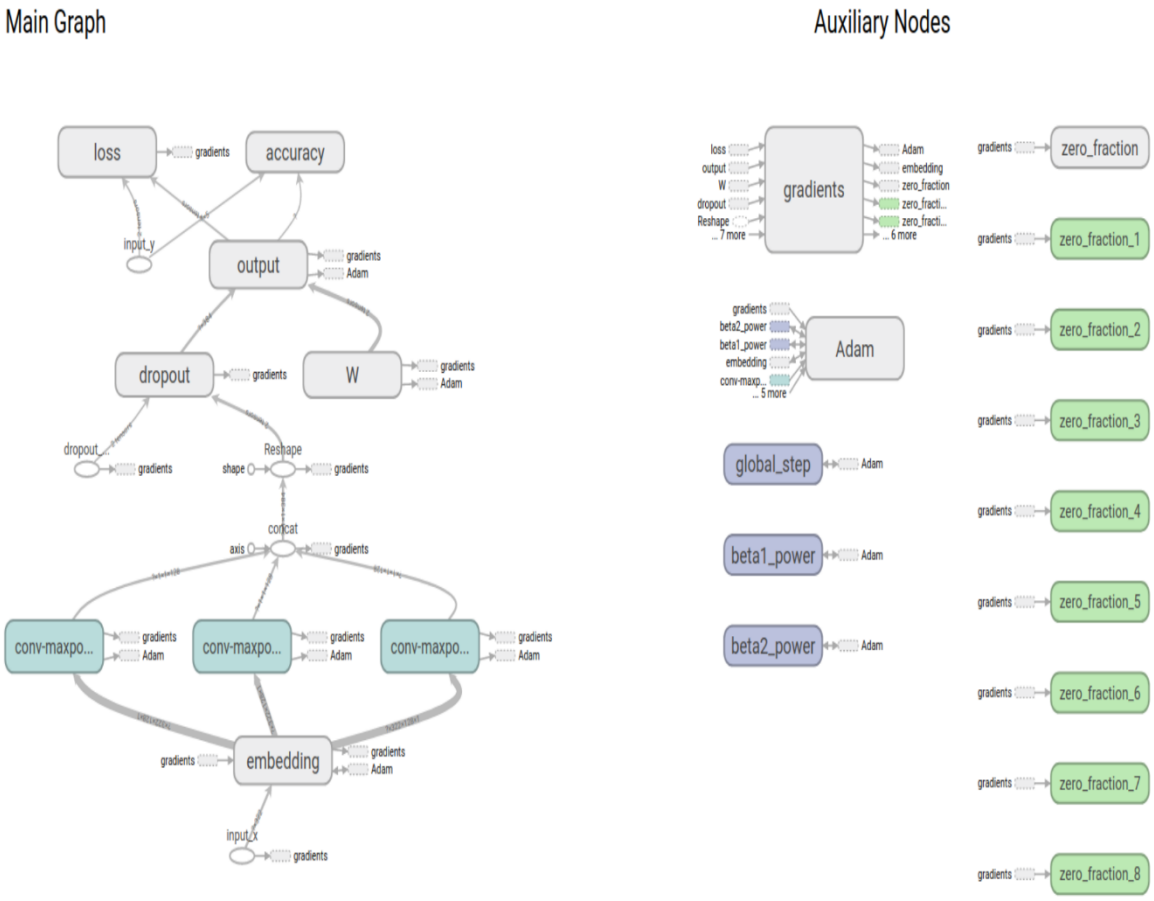
```
x_text, y = data_helpers.load_data_and_labels(FLAGS.fashion1_7000,FLAGS.finance2_7000,
                                              FLAGS.law3_7000, FLAGS.lifestyle4_7000,
                                              FLAGS.TestData5)
```

The data is loaded here as shown below in the eval.py in order to evaluate the result or check the loss and accuracy.

```
if FLAGS.eval_train:

    x_raw, y_test = data_helpers.load_data_and_labels(FLAGS.fashion1_7000,
FLAGS.finance2_7000,
                                                      FLAGS.law3_7000,
FLAGS.lifestyle4_7000, FLAGS.TestData5)
    y_test = np.argmax(y_test, axis=1)

else:

    x_raw = ["master-piece 4 years making", "evrthng off."]
    y_test = [1, 0]
```

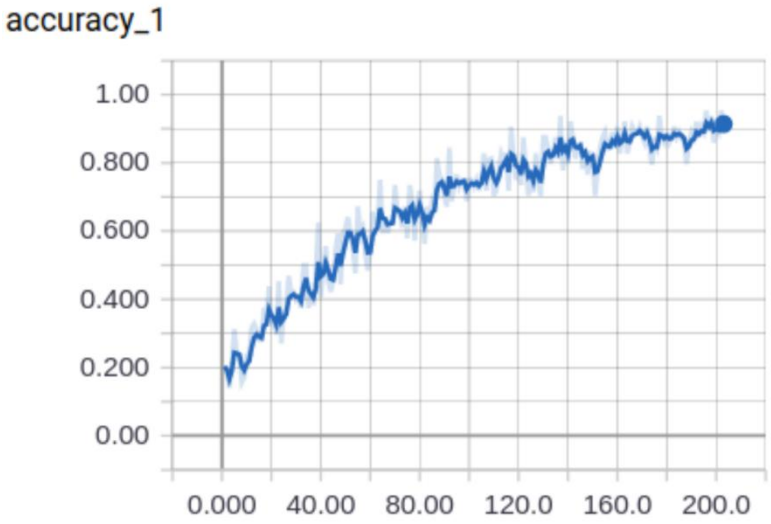
Output:

Here is the output graph from the Tensor Flow.

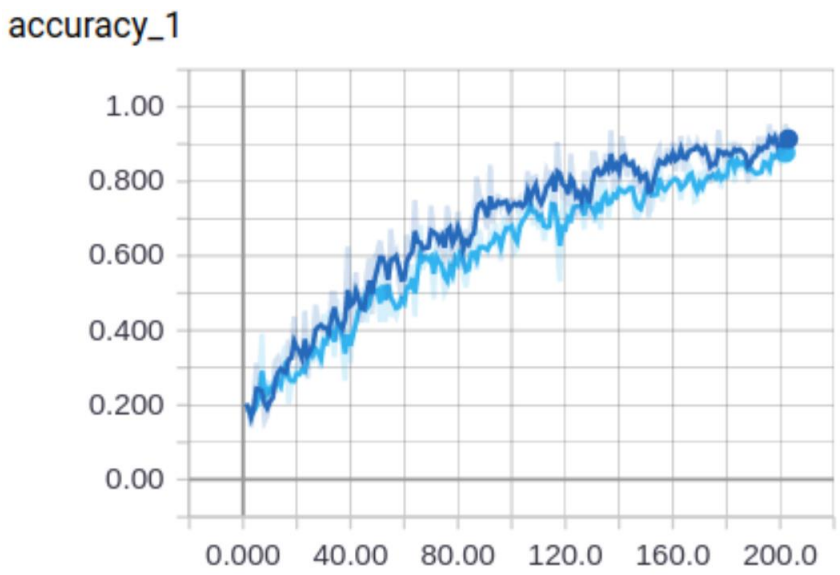


Accuracy Graph:

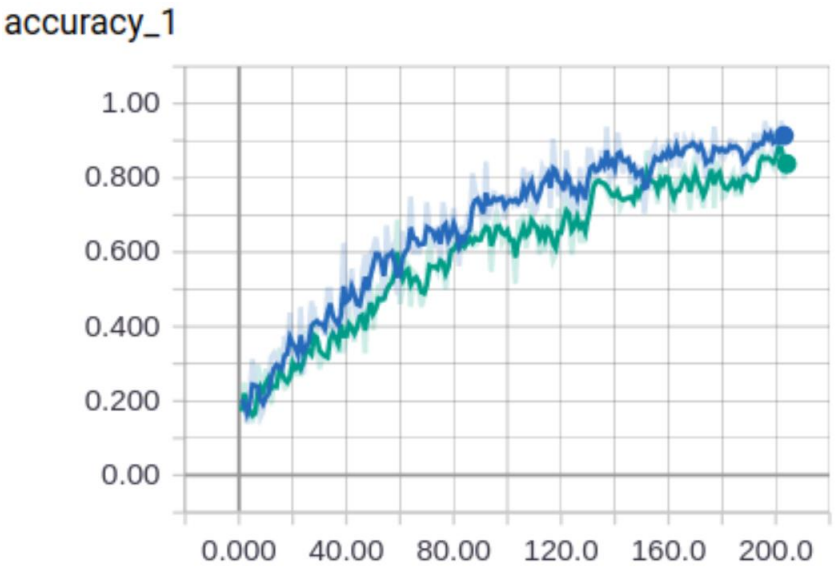
The accuracy for the default values that are provided.



The accuracy when the embedding_dim is reduced to half i.e., 64

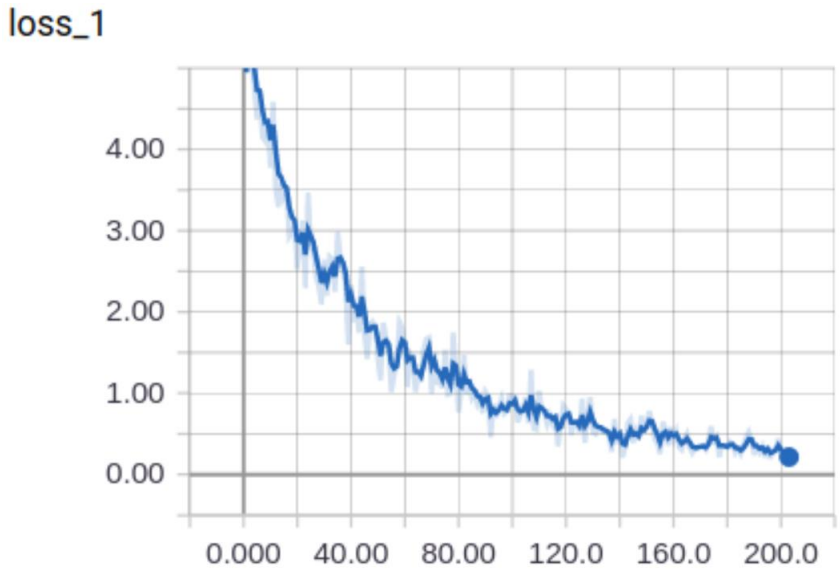


The accuracy when the num_filters is reduced to half of the default value.

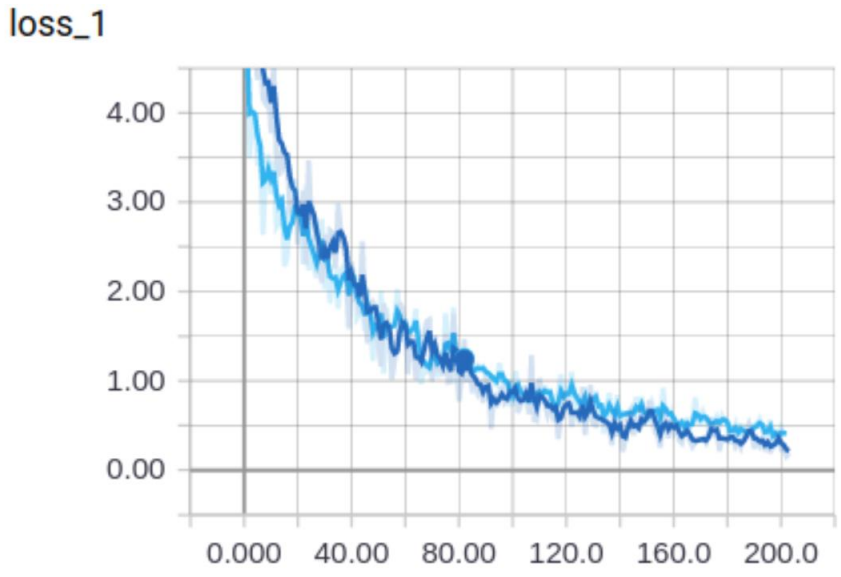


Loss Graph:

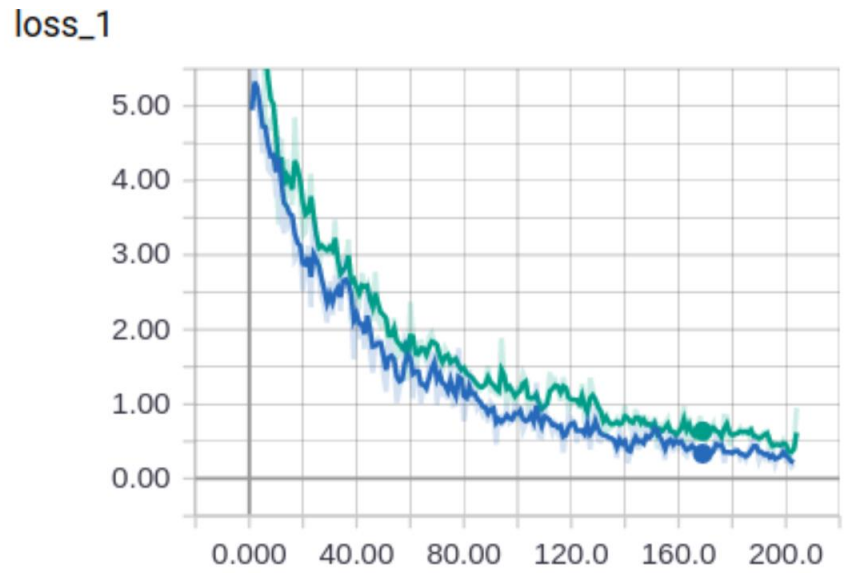
The loss graph with the default values provided for the hyper-parameters.



The loss when embedding_dim is reduced to half.



The loss when num_filters is reduced to half when compared to the default value.



Conclusion:

The loss reduces and the accuracy increases as we increase the number of iterations. By more iterations the system or the model will be more accurate.