UNIVERSITY OF MISSOURI-KANSAS CITY

Python and Deep Learning

 # Lab 3 Assignment

Team Members:

Pragathi Thammaneni

Sridevi Mallipudi

Introduction:

The key objective of this assignment is to focus on tensor flow deep learning library**.**

Objectives:

To code for the 2 questions by implementing the below concepts**.**

- Installed required packages
- Fitting into data frame
- Labels extraction
- Divided into training and testing
- One hot coding
- Implemented logistic regression
- Optimizer to minimize the loss
- Calculated accuracy
- Implemented the word embeddings

Approaches /Methods:

Using Python 3.6, PyCharm (Community edition)

Workflow &Datasets/Parameters and Evaluation:

The below each question will follow different approaches to solve. Coding is done to perform the evaluation of each individual snippet to execute the datasets which are provided as the input parameters.

Question 1:

1. Implement the Logistic Regression with new data set which is not used in class
Show the graph in TensorBoard
Change the hyperparameter and compare the result

Solution:

This snippet code provides the above implementation which for the logistic regression, have considered iris data and used pandas to set into the data frame and installed sklearn package to load the iris data. After the data framing I have implemented logistic regression and calculated loss function to analyze the accuracy. In this data is divided into training and testing set such as 80% of training data and 20% of testing data. At last tensor flow session is created for the logistic regression model and calculated the accuracy

Data Set: Iris Data set

Iris data set is used to implement the logistic regression model. This dataset has three kinds of colors for petal and sepal length to store in n dimensional array.

Code Snippet:

The below is the code for executing the above workflow

103 lines (96 sloc) | 4.02 KB — Raw | Blame | History

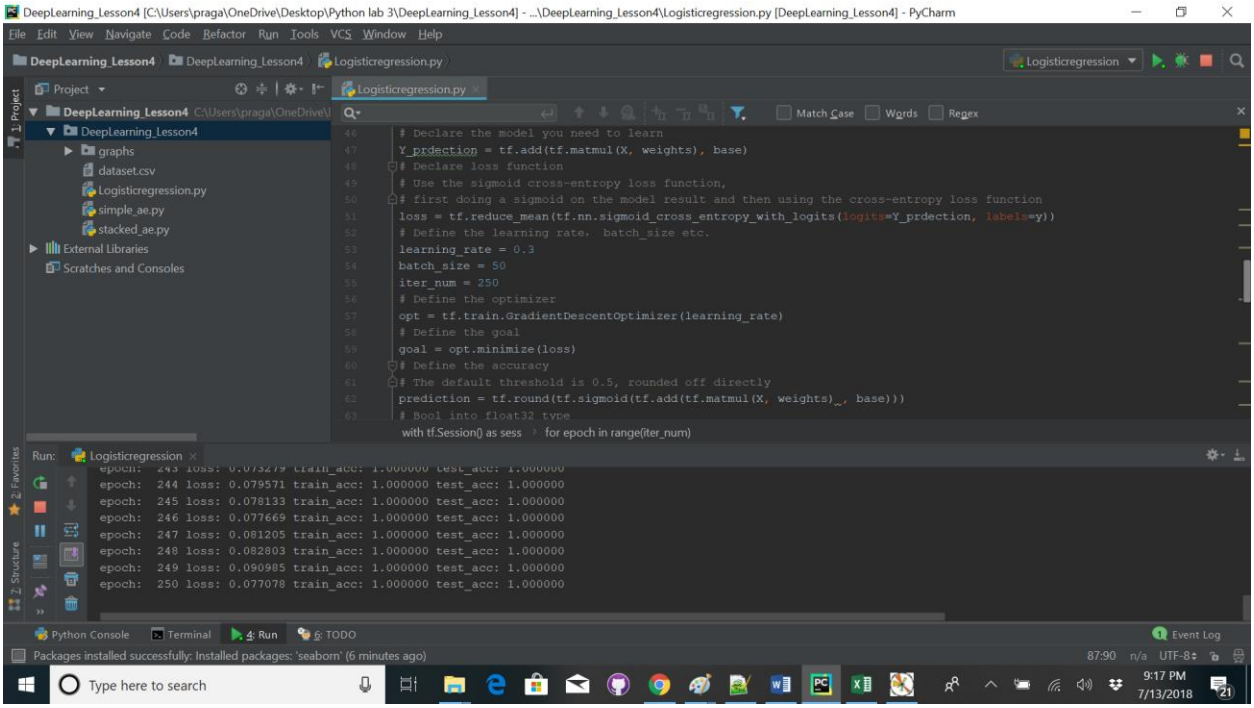
```
import numpy as np
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf

iris = pd.read_csv('dataset.csv')
iris.Species = iris.Species.replace(to_replace=['Iris-setosa', 'Iris-versicolor'], value=[0, 1])

X = iris.drop(labels=['Id', 'Species'], axis=1).values
y = iris.Species.values

# set seed for numpy and tensorflow
# set for reproducible results
seed = 5
np.random.seed(seed)
tf.set_random_seed(seed)
# dataset segmentation
# splitting the dataset as train data and test data
train_index = np.random.choice(len(X), round(len(X) * 0.8), replace=False)
test_index = np.array(list(set(range(len(X))) - set(train_index)))
train_X = X[train_index]
train_y = y[train_index]
test_X = X[test_index]
test_y = y[test_index]
# Define the normalized function
def min_max_normalized(data):
    col_max = np.max(data, axis=0)
    col_min = np.min(data, axis=0)
    return np.divide(data - col_min, col_max - col_min)
# Normalized processing, must be placed after the data set segmentation,
# otherwise the test set will be affected by the training set
train_X = min_max_normalized(train_X)
test_X = min_max_normalized(test_X)

# Declare the variables that need to be learned and initialization
```

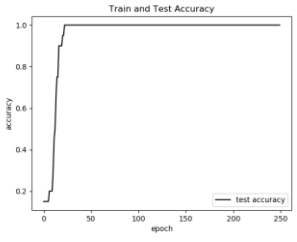
Different optimizer has been implemented

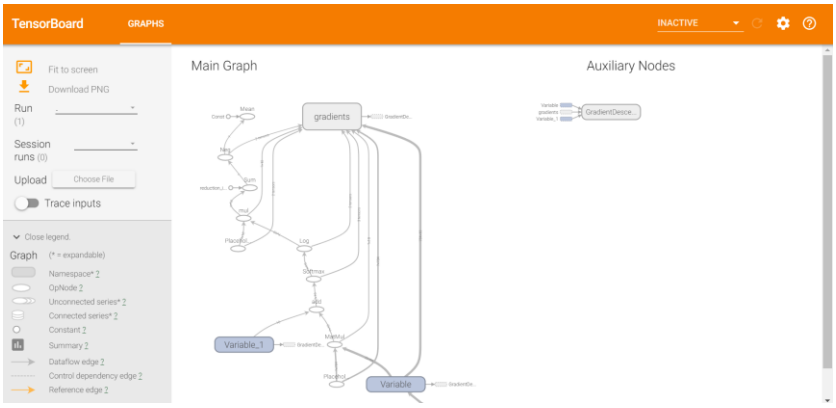
GradientDescentOptimizer, RMSPropOptimizer, AdamOptimizer, AdagradOptimizer with different learning rates. And the loss rate, accuracy and tensor flow is generated for the shown below outputs.

Output Screenshot:

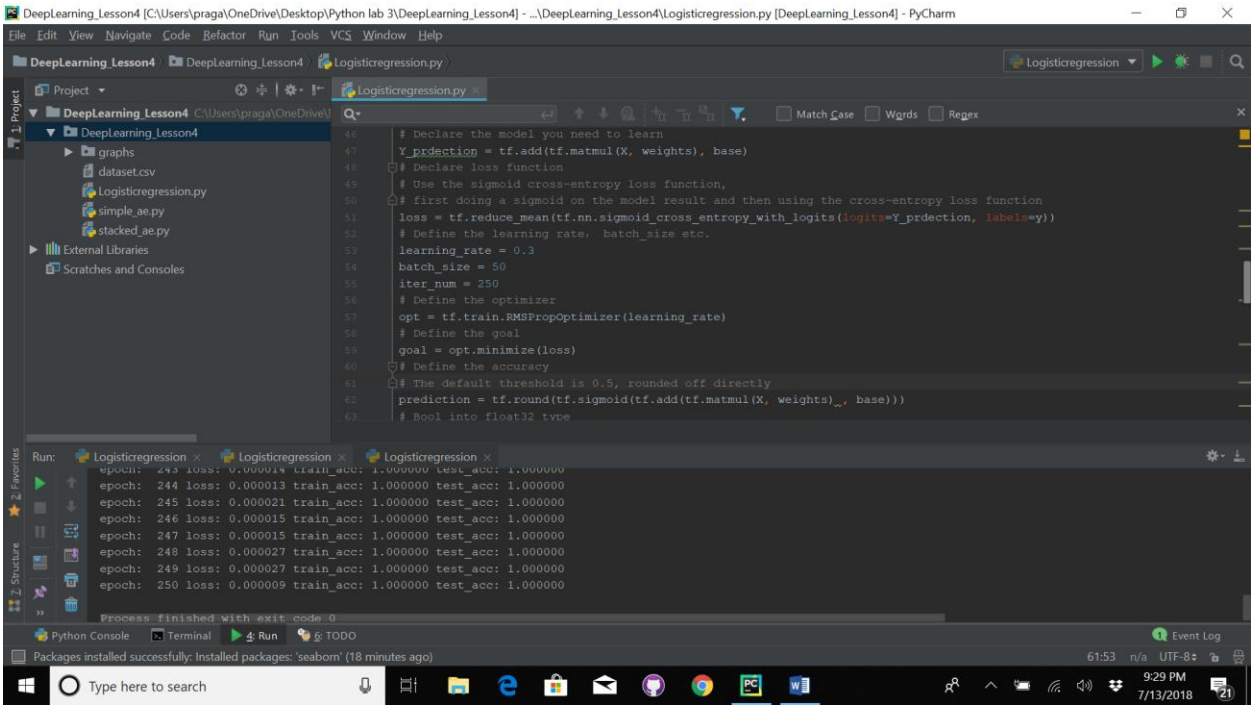
GradientDescentOptimizer ,Tensor flow:

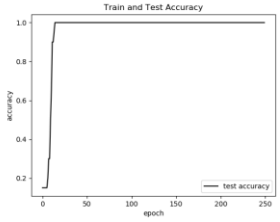


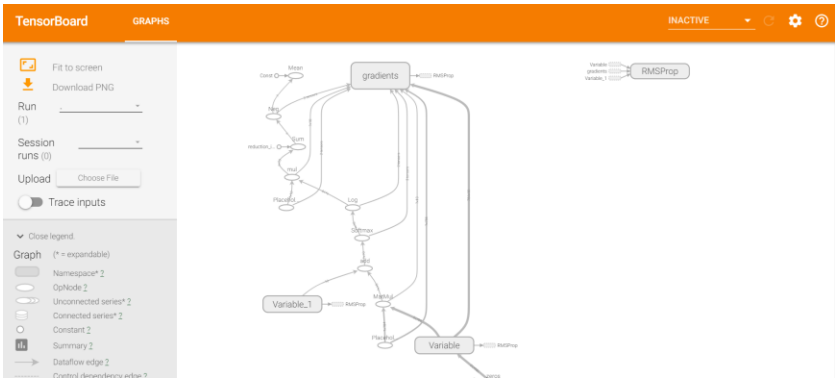




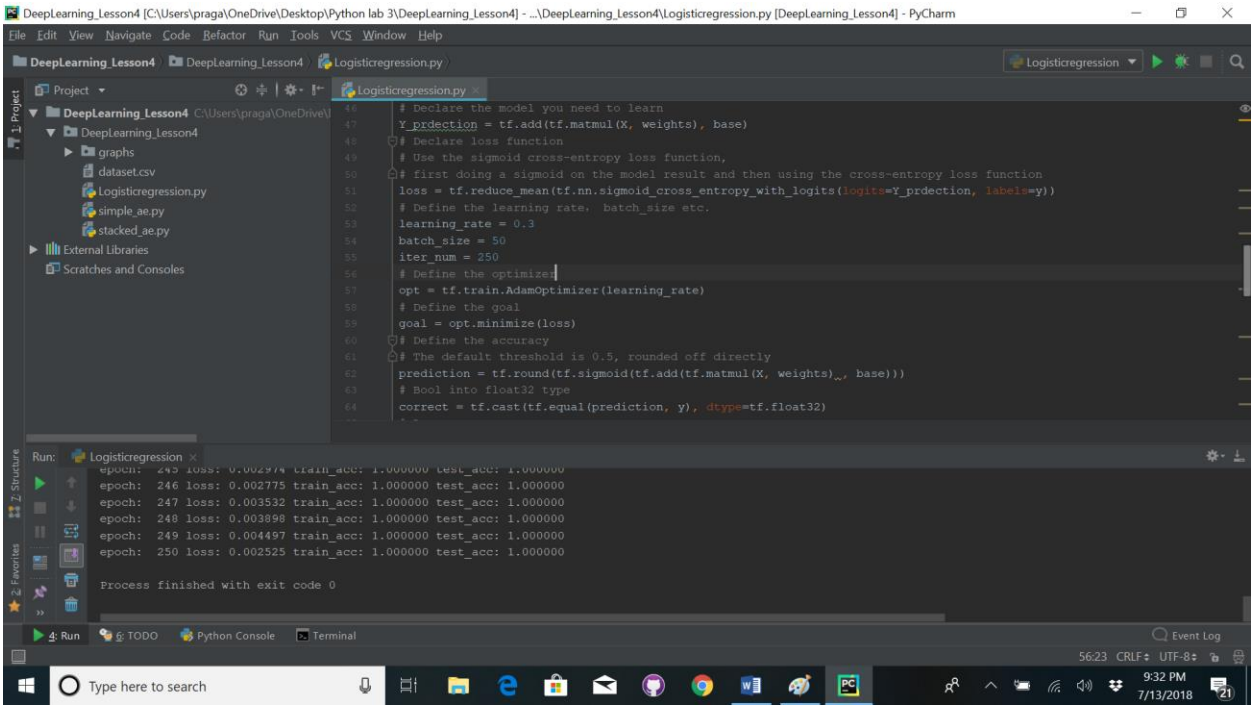
RMSPropOptimizer,Tensor flow:

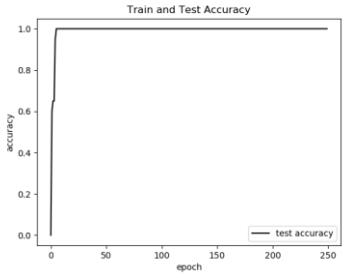


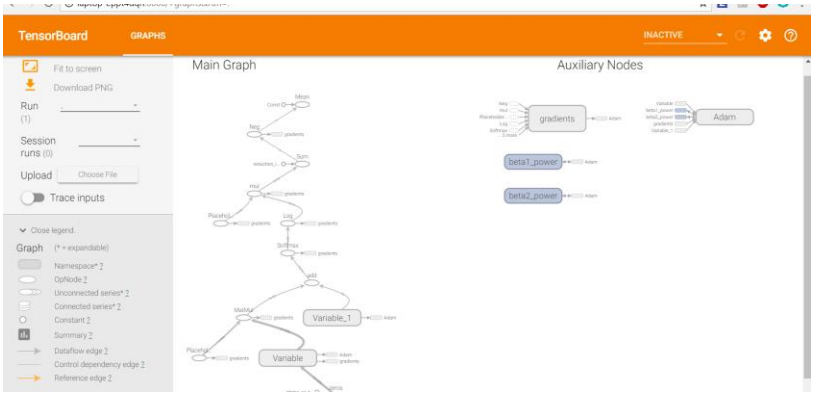




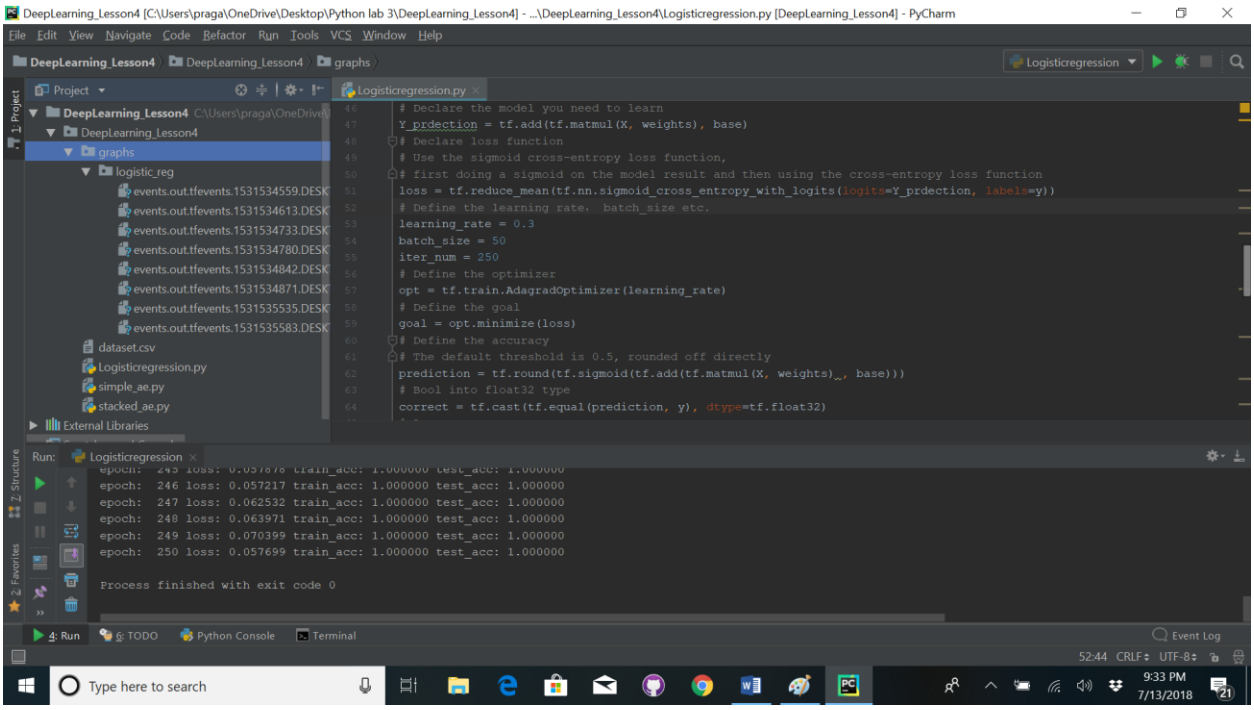
AdamOptimizer,Tensor flow:

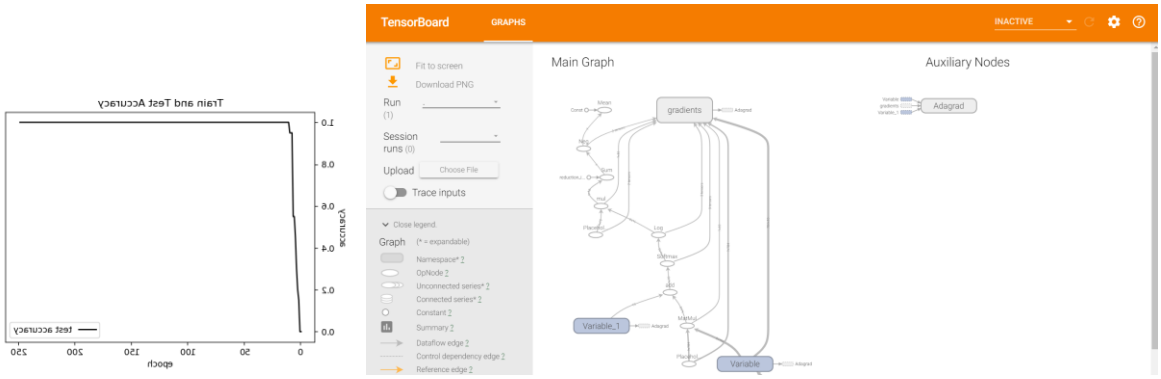






AdagradOptimizer,Tensor flow:





Question 2:

2. Implement the Word Embeddings with new data set which is not used in class
Show the results in TensorBoard
Change the hyperparameter and compare the result

Solution:

This snippet code provides the above implementation For this task we have chosen the ewik8 dataset in 'http://mattmahoney.net/dc/enwik8.zip' URL.We have changed the different hyper parameters like optimizers,learning rate,number of steps,step size and observed the loss.

- **Data Set : enwik8.zip**

Code Snippet:

The below is the code for executing the above workflow

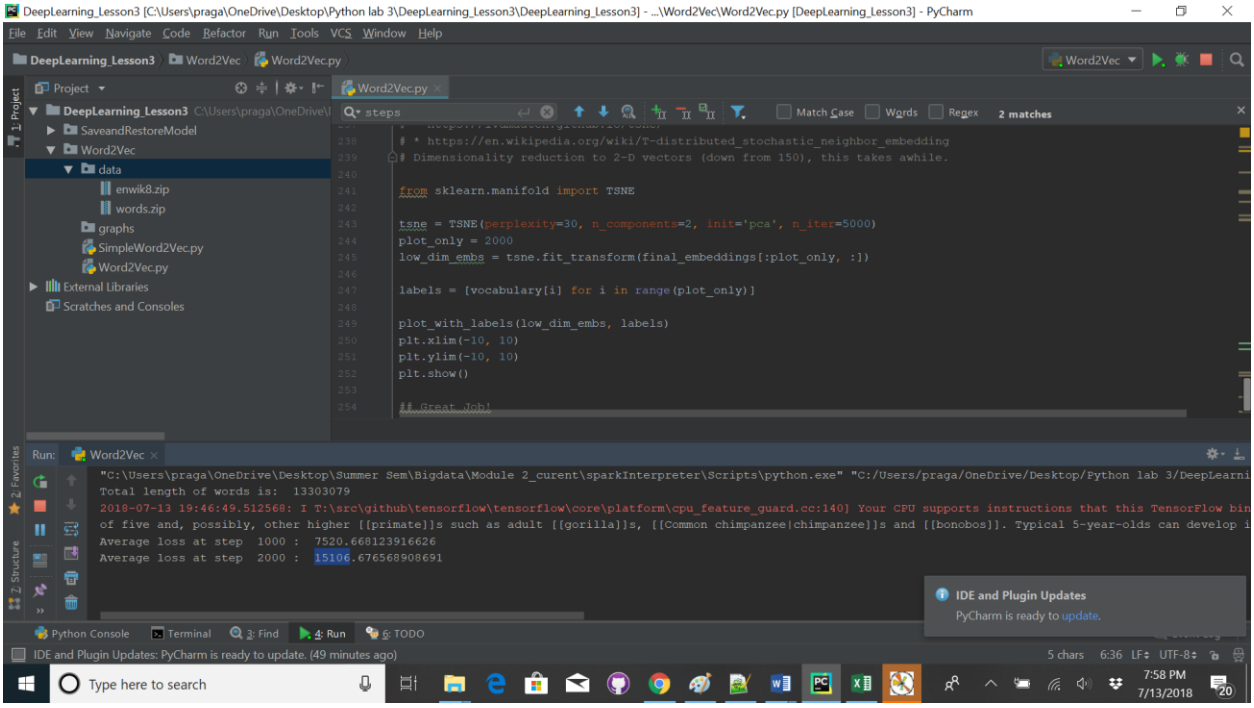
255 lines (187 sloc) | 7.6 KB — Raw | Blame | History

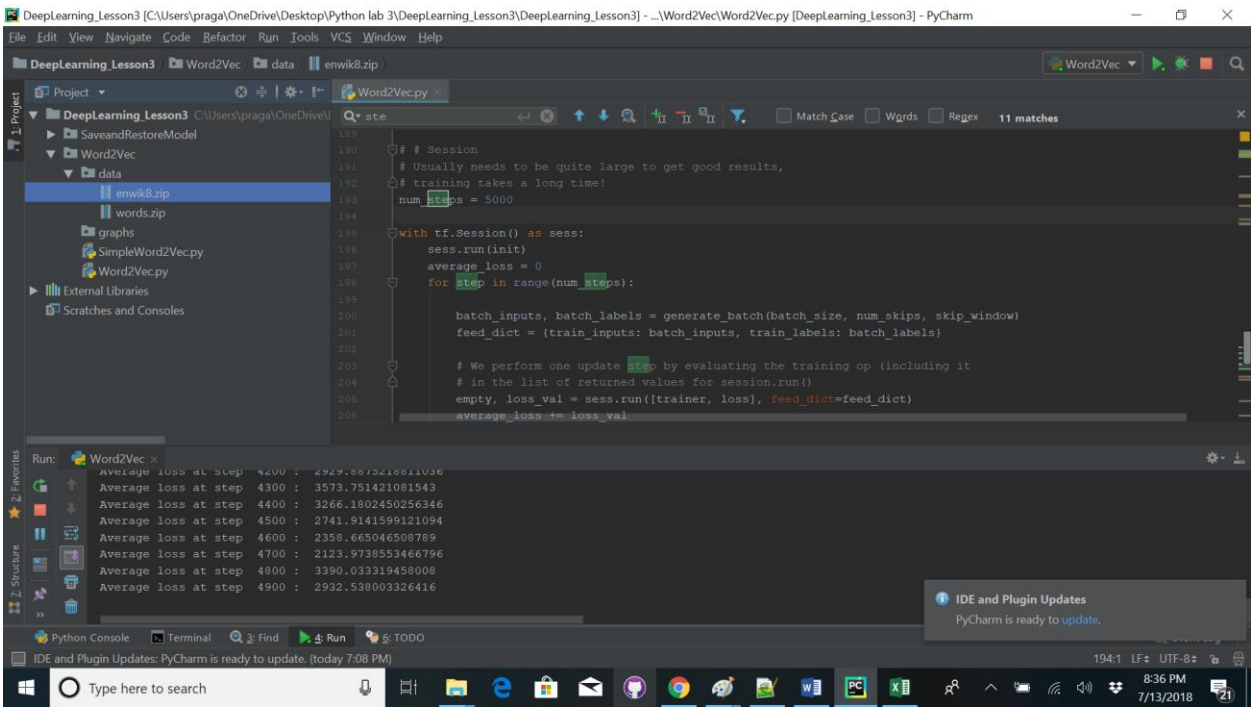
```
# # Word2Vec

# The code for this lecture is based off the great tutorial example from tensorflow!
# Walk through:
# https://www.tensorflow.org/tutorials/word2vec
# Raw Code: https://github.com/tensorflow/tensorflow/blob/master/tensorflow/examples/tutorials/word2vec/word2vec_basic.py

# # Step 0: Imports
import collections
import math
import os
import errno
import random
import zipfile

import numpy as np
from six.moves import urllib
from six.moves import xrange
from collections import Counter
import tensorflow as tf

# # Step 1: The data.
data_dir = "data"
data_url = 'http://mattmahoney.net/dc/enwik8.zip'


def fetch_words_data(url=data_url, words_data=data_dir):
    # Make the Dir if it does not exist
    os.makedirs(words_data, exist_ok=True)

    # Path to zip file
    zip_path = os.path.join(words_data, "enwik8.zip")
```

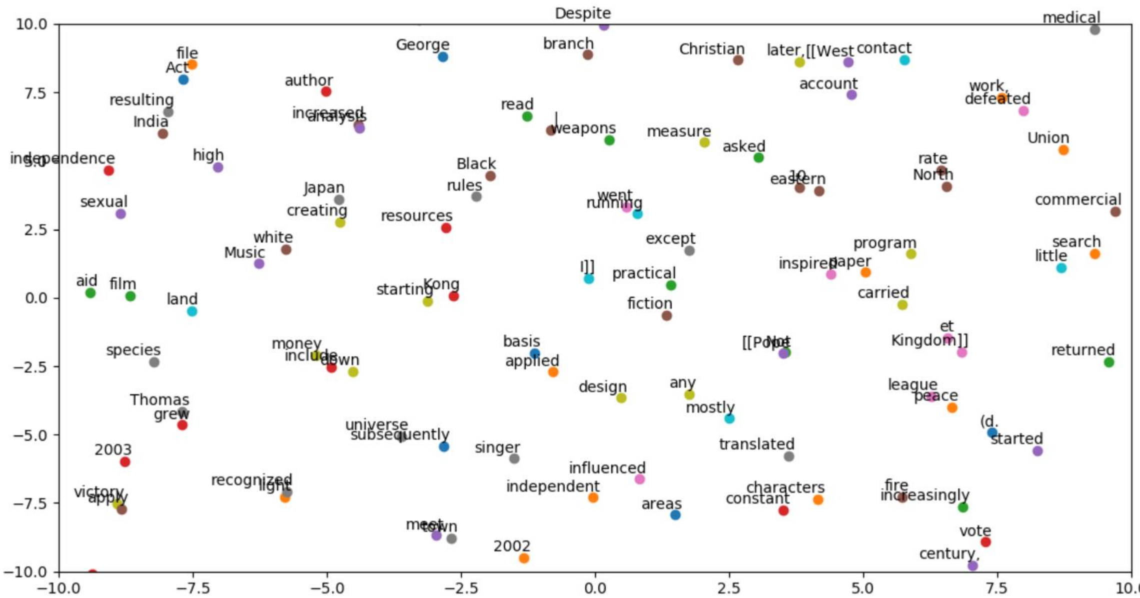
Analysis depending upon hyperparameters:

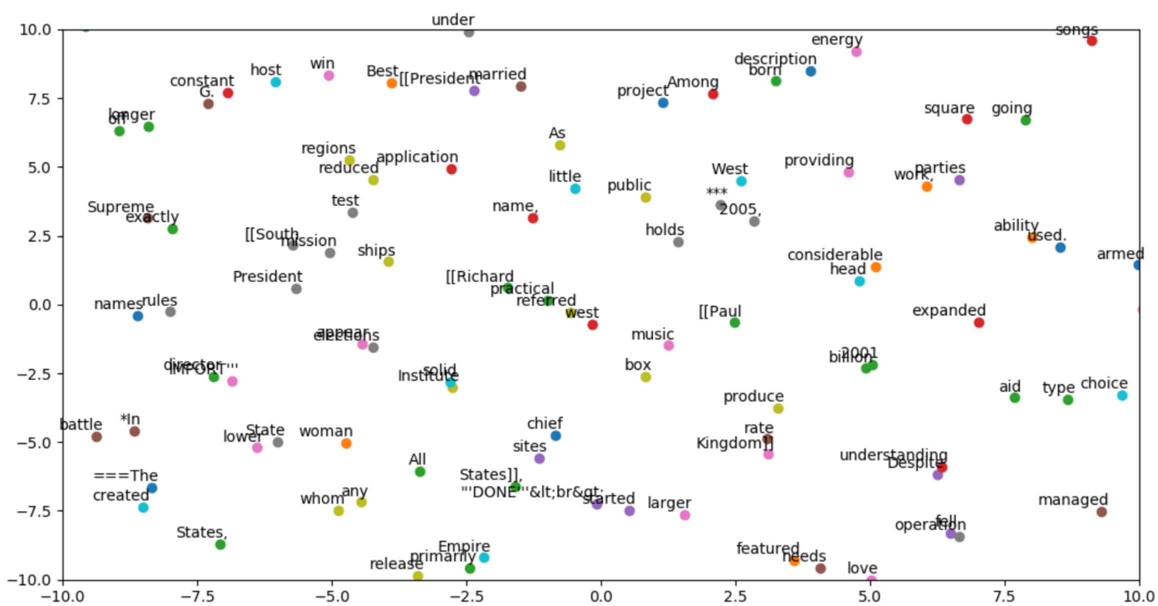
Word Embeddings	Model Learning rate	Optimizer	Window Size	Embedding Size	No.of steps	Loss rate
	0.01	Adam Optimizer	100	150	2001	15106.7
	0.02	Adam Optimizer	120	170	5000	2932.53
	0.01	RMS Optimizer	100	100	7000	2428.24

Output Screenshot:

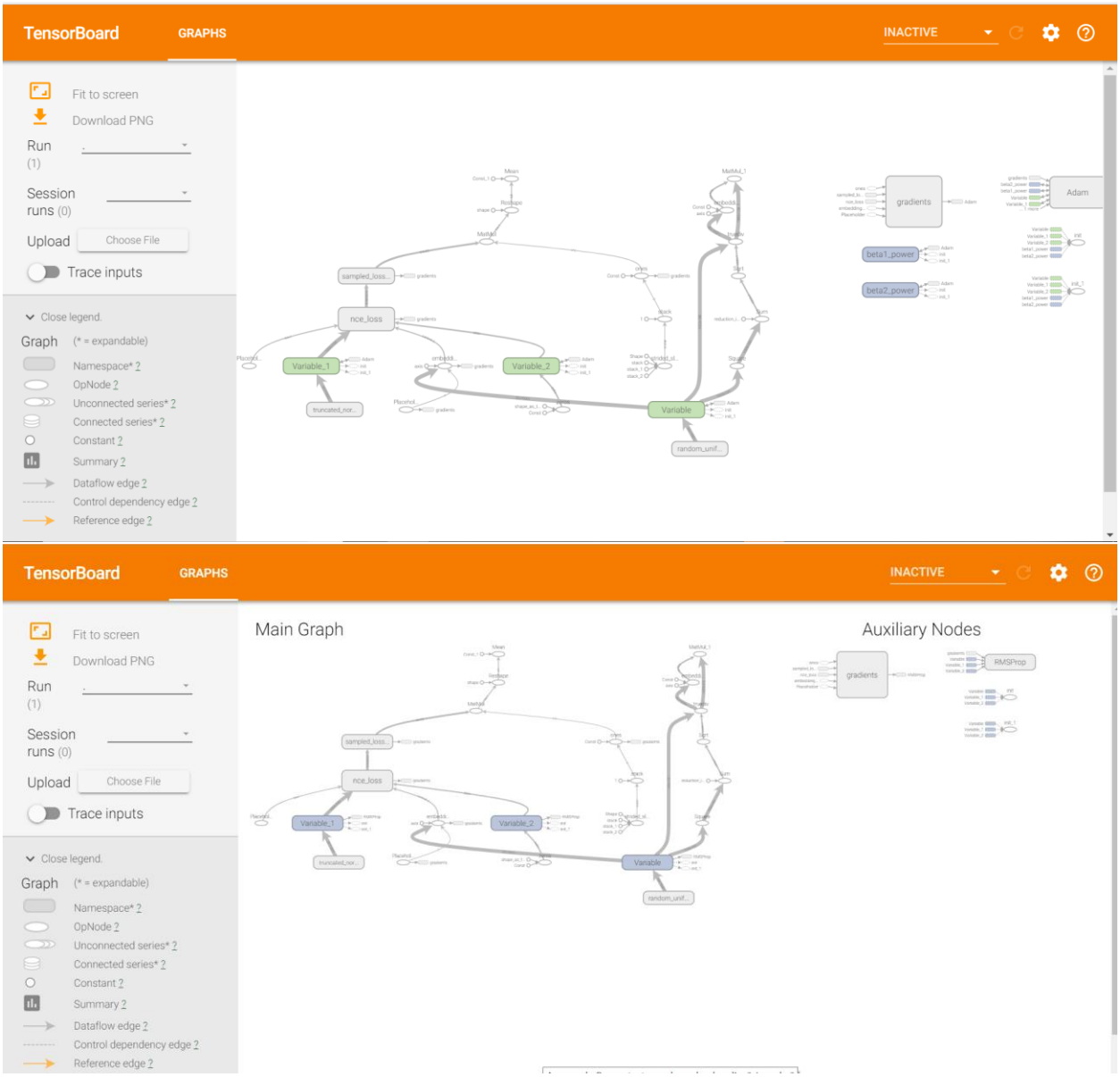





Despite
medical
George
branch
Christian
later,
[[West
contact
file
Act
author
read
account
work,
defeated
resulting
India
weapons
measure
Union
independence
high
asked
rate
North
eastern
Black
Japan
rules
went
running
commercial
sexual
creating
resources
except
white
Music
program
search
little
inspired
paper
aid
film
land
starting
Kong
practical
fiction
carried
et
Kingdom]]
returned
species
money
basis
applied
design
any
league
peace
mostly
Thomas
universe
subsequently
(d.
started
2003
singer
translated
influenced
recognized
independent
characters
constant
fire
increasingly
victory
areas
vote
2002
century,
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


under
energy
songs
description
born
host
win
Best
[[President
married
constant
G.
project
Among
square
going
As
regions
application
providing
reduced
little
public
West
work,
parties
Supreme
exactly
test
name,
***
2005,
holds
ability
used.
[[South
ships
considerable
armed
head
President
[[Richard
practical
referred
west
names
rules
[[Paul
expanded
music
box
2001
billion
Institute
solid
produce
aid
type
choice
battle
*In
State
woman
chief
rate
Kingdom]]
lower
All
sites
understanding
Despite
===The
created
States]],
'''DONE'''
whom
any
started
larger
managed
States,
operation
Empire
featured
needs
primarily
release
love
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

Tensor graph Adam and RMS optimizers:



Final observation:

Depending upon the variation in hyper parameters there is a markable difference in accuracy and loss ,and it also depend on the datasets too.

Conclusion: As stated the above workflow with certain set of parameters is followed in solving the execution by implementing the core and basic concepts of the deep learning programming.

Source code link https://github.com/PragathiThammaneni/Python-and-deep-Learning/tree/master/Labs/Lab%203

Video Link: https://youtu.be/1D264SIytfk

Wiki Link: https://github.com/PragathiThammaneni/Python-and-deep-Learning/wiki/Lab-3-Assignment