

CSEE 5590 0001
Special Topics SPRING 2018

DEEP LEARNING
LAB ASSIGNMENT - 3

Submitted On 4/6/2018

Name: Satya Sai Deepthi Katta

Class ID: 21



Department of Computer Science and Electrical Engineering

DOCUMENT CONTENTS

1. Introduction
2. Objective
3. Approaches/Methods
4. Workflow
5. Datasets
6. Parameters
7. Evaluation & Discussion
8. Conclusion

Introduction

Logistic Regression is the predictive analysis of the data to explain the relationship between one dependent binary variable and one or more nominal, ordinal independent variables. The assignment is to display the Logistic Regression results and graphs to be plotted in Tensor Board.

Objective

The main aim of the lab work is to create an exposure to Deep Learning concepts.

- Implement Logistic using new dataset
- Show the workflow graph in tensor board
- Change the hyperparameter and comparing the results.

Approach

The approach for the assignment can be defined as simple steps given below:

- Importing Data from Dataset
- Assigning X and Y Placeholders
- Variable Weights and Bias Collection
- Construction of prediction model
- Optimize model for less errors
- Train model for the training data
- Compare the prediction and actual model variables
- Compute the accuracy of the model

Parameters

Parameters set in the assignment are listed below:

Learning Rate = 0.01/0.1/1

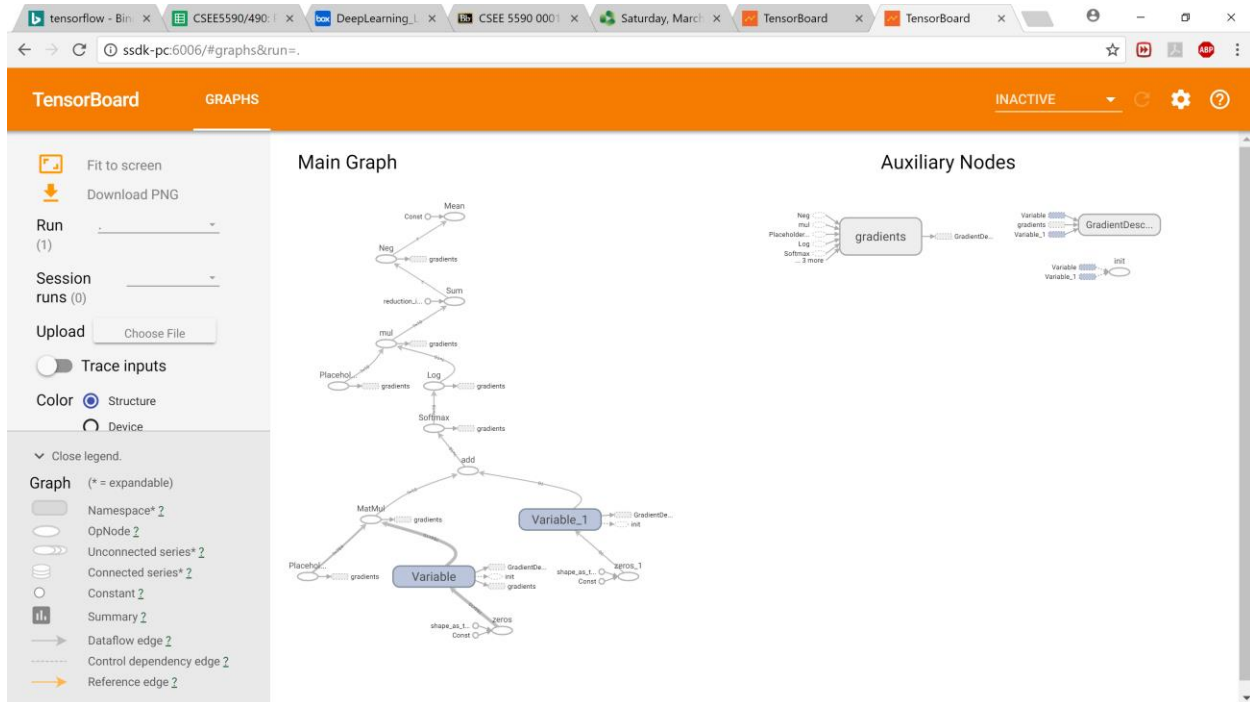
Number of Training Epochs used = 150/100/50

Size of the batch = 10000/1000/100

Displaying step = 10

Workflow

The workflow of Logistic Regression on a Model performed on MNIST Dataset is shown in Tensor Board.

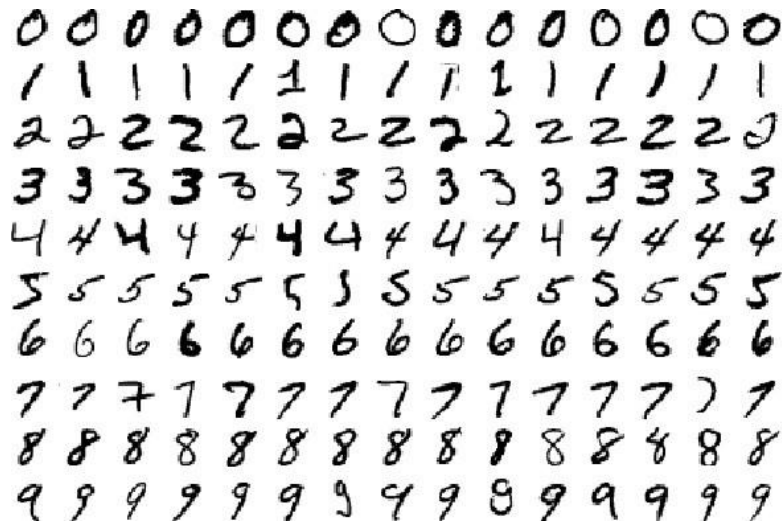


Dataset

The dataset used is *MNIST Dataset*.

MNIST: Mixed National Institute of Standards and Technology Database.

- This dataset consists hand written digits ranging from 0 to 9.
- Every image is of 28 x 28 pixel flattened with one dimensional tensor size of 784 where label is provided for each.
- The datasets for certain values is given below, where images in first row are labeled 0, in second row labeled 1 and goes on.



Configuration

For executing the tasks given in the lab assignment, advanced version of **Python 3.6.4** is used and the code is built in **PYCHARM** Software.

Evaluation & Discussion

The code snippets are provided evaluating the logistic regression performed on MNIST dataset.

Firstly, tensor flow needs to be imported into the work space. The other import statements suppress the warnings raised while running tensor flow in unfavorable OS systems.

```
import os
os.environ['TF_CPP_MIN_LOG_LEVEL']='2'
import tensorflow as t_f
```

To read the data from datasets 'input_data' needs to be imported from tensor flow tutorials.

```
# Import new dataset not used in class : MNIST Dataset
from tensorflow.examples.tutorials.mnist import input_data
#saving the data
m = input_data.read_data_sets("", one_hot=True)
```

Now parameters are defined to define the model required for calculating the accuracy of the dataset. Learning rate, number of epochs, training batch size, displaying epoch size are the parameters set for the model.

```
# Parameters defined to train the data
l_r = 0.1 # learning rate for the dataset
```

```
t_e = 25    # number of training epochs used
b_s = 90    # batch size for training data
d_s = 1     # step size for displaying epochs
```

To display the output graph in Tensor Board, x and y variables are set with place holders for up to 784 data points and classifying 10 classes (0-9 digits).

```
# Tensor Flow Graph Input
# mnist data image of shape 28*28=784 in the database
x_var = t_f.placeholder(t_f.float32, [None, 784])
# Variable needs to recognize 0-9 digits i.e, 10 classes
y_var = t_f.placeholder(t_f.float32, [None, 10])
```

In the model, ‘Weights’ and ‘Bias’ are set as variable values which changes according to the data points taken from the set. The range is within the placeholder limits.

```
# Setting Model Weights for the Dataset
Wei = t_f.Variable(t_f.zeros([784, 10]))
bi = t_f.Variable(t_f.zeros([10]))
```

A prediction model is constructed using softmax. Cross entropy is used in reducing the mean error. An optimizer for Gradient Descent is used to reduce the error as much as possible which increases the accuracy of the model.

```
# Prediction Model Construction
prdt = t_f.nn.softmax(t_f.matmul(x_var, Wei) + bi)

# Cross entropy is used in reducing the error
cst = t_f.reduce_mean(-t_f.reduce_sum(y_var*t_f.log(prdt), reduction_indices=1))

# Gradient Descent
optm = t_f.train.GradientDescentOptimizer(1_r).minimize(cst)

# Initialize the variables (i.e. assign their default value)
st = t_f.global_variables_initializer()
```

The dataset is trained iteratively within the session time for the number of epochs given specified during the parameters initialization.

```
# Training session begins here
with t_f.Session() as s:

    # Run the initializer
    s.run(st)
    writer = t_f.summary.FileWriter('./graphs/logistic_reg', s.graph)

    # Training cycle for the specified range of epochs
    for ep in range(t_e):
        ag_ct = 0.
        t_b = int(m.train.num_examples/b_s) #total batch taken

        # Over all looping of the batch
        for i in range(t_b):
            batch_xs, batch_ys = m.train.next_batch(b_s)

            #Running optimization
            _, c = s.run([optm, cst], feed_dict={x: batch_xs, y: batch_ys})
```

```

        # Average cost calculation
        ag_ct += c / t_b

    # Display logs per epoch step
    if (ep+1) % d_s == 0:

        print("Epoch:", '%04d' % (ep+1), "cost ", "{:.9f}".format(ag_ct))

    print("Complete Optimization Done!")

```

A prediction correction is obtained comparing the predicted value and the y variable taken from the dataset. The accuracy is calculated by taking reduced mean between them.

```

# Designed Test Model

c_p = t_f.equal(t_f.argmax(prdt, 1), t_f.argmax(y, 1))
#Accuracy Calculation for the developed epochs
acc = t_f.reduce_mean(t_f.cast(c_p, t_f.float32))
print("Accuracy of the MNIST Dataset:", acc.eval({x: m.test.images, y:
m.test.labels}))

```

Results

Case 1:

Changing Learning Rate

Logistic Regression of the MNIST Dataset considering parameters below

Training Epochs = 150; Batch size = 10000; constant display size

Learning Rate= 1

```

DL_LAB1 [C:\Users\satyasaideepthi\PycharmProjects\DL_LAB1] - task1.py [DL_LAB1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help

DL_LAB1 task1.py
10
11 # Parameters Defined
12 l_r = 1 # learning rate for the dataset
13 t_e = 150 # number of training epochs used
14 b_s = 10000 # batch size for training data
15 d_s = 10 # step size for displaying epochs
16
17 # Tensor Flow Graph Input

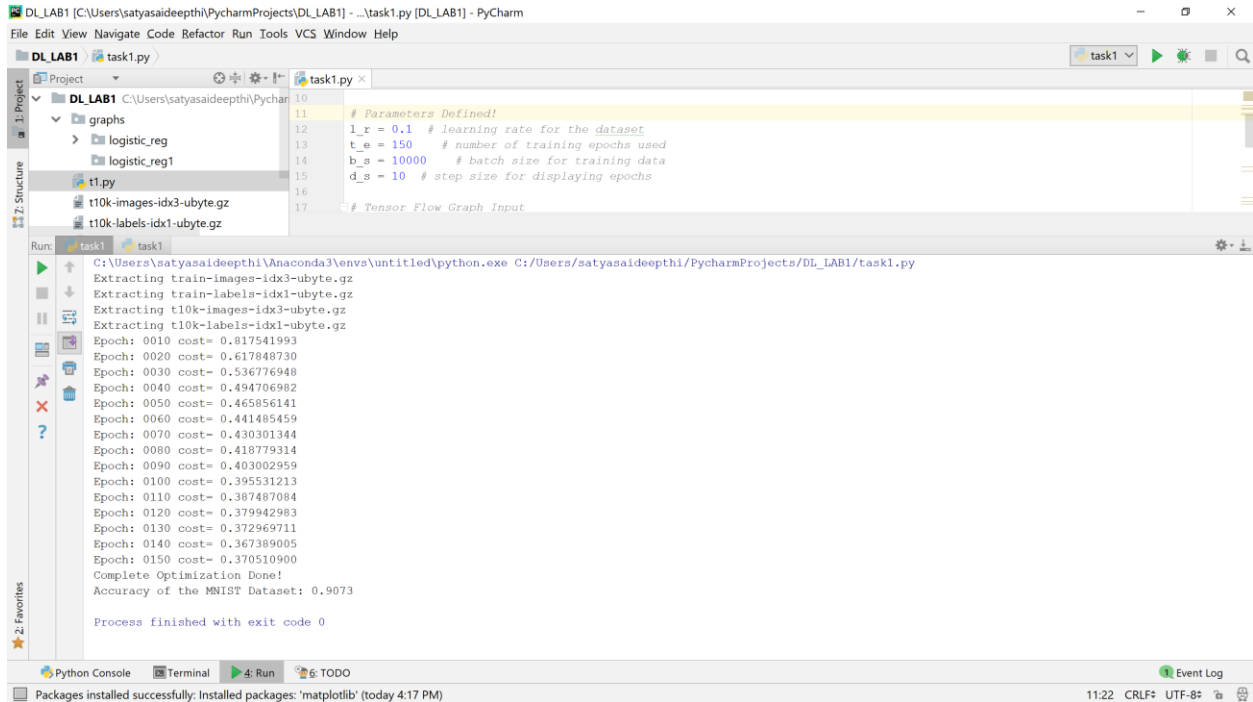
Run: task1
C:\Users\satyasaideepthi\Anaconda3\envs\untitled\python.exe C:/Users/satyasaideepthi/PycharmProjects/DL_LAB1/task1.py
Extracting train-images-idx3-ubyte.gz
Extracting train-labels-idx1-ubyte.gz
Extracting t10k-images-idx3-ubyte.gz
Extracting t10k-labels-idx1-ubyte.gz
Epoch: 0010 cost= 0.396199197
Epoch: 0020 cost= 0.346530694
Epoch: 0030 cost= 0.330156726
Epoch: 0040 cost= 0.319048935
Epoch: 0050 cost= 0.307238138
Epoch: 0060 cost= 0.300852036
Epoch: 0070 cost= 0.297428578
Epoch: 0080 cost= 0.293768561
Epoch: 0090 cost= 0.291635936
Epoch: 0100 cost= 0.284100199
Epoch: 0110 cost= 0.281596446
Epoch: 0120 cost= 0.281817585
Epoch: 0130 cost= 0.280054754
Epoch: 0140 cost= 0.276479262
Epoch: 0150 cost= 0.275187474
Complete Optimization Done!
Accuracy of the MNIST Dataset: 0.9222

Process finished with exit code 0

Python Console Terminal Run TODO
Packages installed successfully: Installed packages: 'matplotlib' (today 4:17 PM)
11:22 CRLF UTF-8

```

Learning Rate = 0.1



The screenshot shows the PyCharm IDE with a project named 'DL_LAB1'. The file 'task1.py' is open, showing parameters defined for a training process. The learning rate is set to 0.1. The output window shows the execution of the script, which includes extracting training data and displaying the cost for each epoch from 0010 to 0150. The final accuracy of the MNIST Dataset is 0.9073.

```
10 # Parameters Defined!
11
12 l_r = 0.1 # learning rate for the dataset
13 t_e = 150 # number of training epochs used
14 b_s = 10000 # batch size for training data
15 d_s = 10 # step size for displaying epochs
16
17 # Tensor Flow Graph Input
```

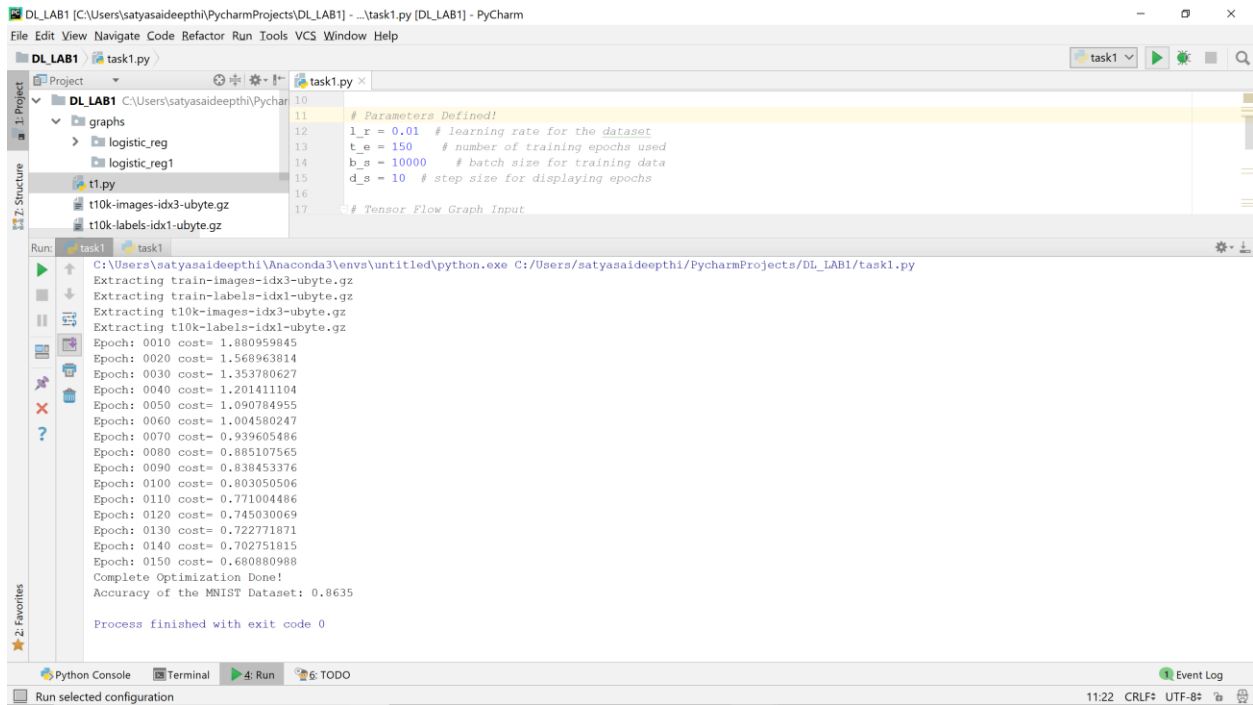
Run: C:\Users\satyasaideepthi\Anaconda3\envs\untitled\python.exe C:/Users/satyasaideepthi/PycharmProjects/DL_LAB1/task1.py

Extracting train-images-idx3-ubyte.gz
Extracting train-labels-idx1-ubyte.gz
Extracting t10k-images-idx3-ubyte.gz
Extracting t10k-labels-idx1-ubyte.gz

Epoch: 0010 cost= 0.817541993
Epoch: 0020 cost= 0.617848730
Epoch: 0030 cost= 0.536776948
Epoch: 0040 cost= 0.494706982
Epoch: 0050 cost= 0.465856141
Epoch: 0060 cost= 0.441485459
Epoch: 0070 cost= 0.430301344
Epoch: 0080 cost= 0.418779314
Epoch: 0090 cost= 0.403002959
Epoch: 0100 cost= 0.395531213
Epoch: 0110 cost= 0.387487084
Epoch: 0120 cost= 0.379942983
Epoch: 0130 cost= 0.372969711
Epoch: 0140 cost= 0.367389005
Epoch: 0150 cost= 0.370510900
Complete Optimization Done!
Accuracy of the MNIST Dataset: 0.9073

Process finished with exit code 0

Learning Rate = 0.01



The screenshot shows the PyCharm IDE with the same project 'DL_LAB1'. The file 'task1.py' is open, but the learning rate is now set to 0.01. The output window shows the execution of the script, which includes extracting training data and displaying the cost for each epoch from 0010 to 0150. The final accuracy of the MNIST Dataset is 0.8635.

```
10 # Parameters Defined!
11
12 l_r = 0.01 # learning rate for the dataset
13 t_e = 150 # number of training epochs used
14 b_s = 10000 # batch size for training data
15 d_s = 10 # step size for displaying epochs
16
17 # Tensor Flow Graph Input
```

Run: C:\Users\satyasaideepthi\Anaconda3\envs\untitled\python.exe C:/Users/satyasaideepthi/PycharmProjects/DL_LAB1/task1.py

Extracting train-images-idx3-ubyte.gz
Extracting train-labels-idx1-ubyte.gz
Extracting t10k-images-idx3-ubyte.gz
Extracting t10k-labels-idx1-ubyte.gz

Epoch: 0010 cost= 1.880959845
Epoch: 0020 cost= 1.568963814
Epoch: 0030 cost= 1.353780627
Epoch: 0040 cost= 1.201411104
Epoch: 0050 cost= 1.090784955
Epoch: 0060 cost= 1.004580247
Epoch: 0070 cost= 0.939605486
Epoch: 0080 cost= 0.885107565
Epoch: 0090 cost= 0.838453376
Epoch: 0100 cost= 0.803050506
Epoch: 0110 cost= 0.771004486
Epoch: 0120 cost= 0.745030069
Epoch: 0130 cost= 0.722771871
Epoch: 0140 cost= 0.702751815
Epoch: 0150 cost= 0.680809088
Complete Optimization Done!
Accuracy of the MNIST Dataset: 0.8635

Process finished with exit code 0

Case 2:

Changing Batch Size

Logistic Regression of the MNIST Dataset considering parameters below

Training Epochs = 150; Learning Rate = 0.01; constant display size

Batch Size = 10000

```
DL_LAB1 [C:\Users\satyasaideepthi\PycharmProjects\DL_LAB1] - ...task1.py [DL_LAB1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help

DL_LAB1 task1.py
Project: DL_LAB1 C:\Users\satyasaideepthi\PycharmProjects\DL_LAB1
Structure:
- graphs
  - logistic_reg
  - logistic_reg1
  - t1.py
  - t10k-images-idx3-ubyte.gz
  - t10k-labels-idx1-ubyte.gz
Run: task1 task1
C:\Users\satyasaideepthi\Anaconda3\envs\untitled\python.exe C:\Users\satyasaideepthi\PycharmProjects\DL_LAB1\task1.py
Extracting train-images-idx3-ubyte.gz
Extracting train-labels-idx1-ubyte.gz
Extracting t10k-images-idx3-ubyte.gz
Extracting t10k-labels-idx1-ubyte.gz
Epoch: 0010 cost= 1.880959845
Epoch: 0020 cost= 1.568963814
Epoch: 0030 cost= 1.353780627
Epoch: 0040 cost= 1.201411104
Epoch: 0050 cost= 1.090784955
Epoch: 0060 cost= 1.004580247
Epoch: 0070 cost= 0.939605486
Epoch: 0080 cost= 0.885107565
Epoch: 0090 cost= 0.838453376
Epoch: 0100 cost= 0.803050506
Epoch: 0110 cost= 0.771004486
Epoch: 0120 cost= 0.745030069
Epoch: 0130 cost= 0.722771871
Epoch: 0140 cost= 0.702751815
Epoch: 0150 cost= 0.680880988
Complete Optimization Done!
Accuracy of the MNIST Dataset: 0.8635
Process finished with exit code 0
```

Batch size = 1000

```
DL_LAB1 [C:\Users\satyasaideepthi\PycharmProjects\DL_LAB1] - ...task1.py [DL_LAB1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help

DL_LAB1 task1.py
Project: DL_LAB1 C:\Users\satyasaideepthi\PycharmProjects\DL_LAB1
Structure:
- graphs
  - logistic_reg
  - logistic_reg1
  - t1.py
  - t10k-images-idx3-ubyte.gz
  - t10k-labels-idx1-ubyte.gz
Run: task1 task1
C:\Users\satyasaideepthi\Anaconda3\envs\untitled\python.exe C:\Users\satyasaideepthi\PycharmProjects\DL_LAB1\task1.py
Extracting train-images-idx3-ubyte.gz
Extracting train-labels-idx1-ubyte.gz
Extracting t10k-images-idx3-ubyte.gz
Extracting t10k-labels-idx1-ubyte.gz
Epoch: 0010 cost= 0.787406903
Epoch: 0020 cost= 0.597607996
Epoch: 0030 cost= 0.522822097
Epoch: 0040 cost= 0.480958513
Epoch: 0050 cost= 0.453461552
Epoch: 0060 cost= 0.433674313
Epoch: 0070 cost= 0.418564809
Epoch: 0080 cost= 0.406517038
Epoch: 0090 cost= 0.396633474
Epoch: 0100 cost= 0.388324095
Epoch: 0110 cost= 0.381207262
Epoch: 0120 cost= 0.375017610
Epoch: 0130 cost= 0.369569717
Epoch: 0140 cost= 0.364726973
Epoch: 0150 cost= 0.360372411
Complete Optimization Done!
Accuracy of the MNIST Dataset: 0.9081
Process finished with exit code 0
```

Batch size = 100

DL_LAB1 [C:\Users\satyasaideepthi\PycharmProjects\DL_LAB1] - task1.py [DL_LAB1] - PyCharm

File Edit View Navigate Code Refactor Run Tools VCS Window Help

DL_LAB1 task1.py

Project DL_LAB1 C:\Users\satyasaideepthi\PycharmProjects\DL_LAB1

- graphs
 - logistic_reg
 - logistic_reg1
 - t1.py
- t10k-images-idx3-ubyte.gz
- t10k-labels-idx1-ubyte.gz

task1.py

```
10 # Parameters Defined
11 l_r = 0.01 # learning rate for the dataset
12 t_e = 150 # number of training epochs used
13 b_s = 100 # batch size for training data
14 d_s = 10 # step size for displaying epochs
15
16 # Tensor Flow Graph Input
```

Run: task1

C:\Users\satyasaideepthi\Anaconda3\envs\untitled\python.exe C:\Users\satyasaideepthi\PycharmProjects\DL_LAB1\task1.py

Extracting train-images-idx3-ubyte.gz
Extracting train-labels-idx1-ubyte.gz
Extracting t10k-images-idx3-ubyte.gz
Extracting t10k-labels-idx1-ubyte.gz

Epoch: 0010 cost= 0.392413943
Epoch: 0020 cost= 0.345417452
Epoch: 0030 cost= 0.325033195
Epoch: 0040 cost= 0.312840153
Epoch: 0050 cost= 0.304405243
Epoch: 0060 cost= 0.298268893
Epoch: 0070 cost= 0.293369387
Epoch: 0080 cost= 0.289366194
Epoch: 0090 cost= 0.286089651
Epoch: 0100 cost= 0.283225651
Epoch: 0110 cost= 0.280746252
Epoch: 0120 cost= 0.278599487
Epoch: 0130 cost= 0.276688474
Epoch: 0140 cost= 0.274886769
Epoch: 0150 cost= 0.273341317
Complete Optimization Done!
Accuracy of the MNIST Dataset: 0.9227

Process finished with exit code 0

Python Console Terminal Run TODO

Packages installed successfully: Installed packages: 'matplotlib' (today 4:17 PM)

11:22 CRLF UTF-8

Case 3:

Changing Epoch Size

Logistic Regression of the MNIST Dataset considering parameters below

Learning Rate = 0.01; constant display size; Batch Size = 10000

Epochs = 150

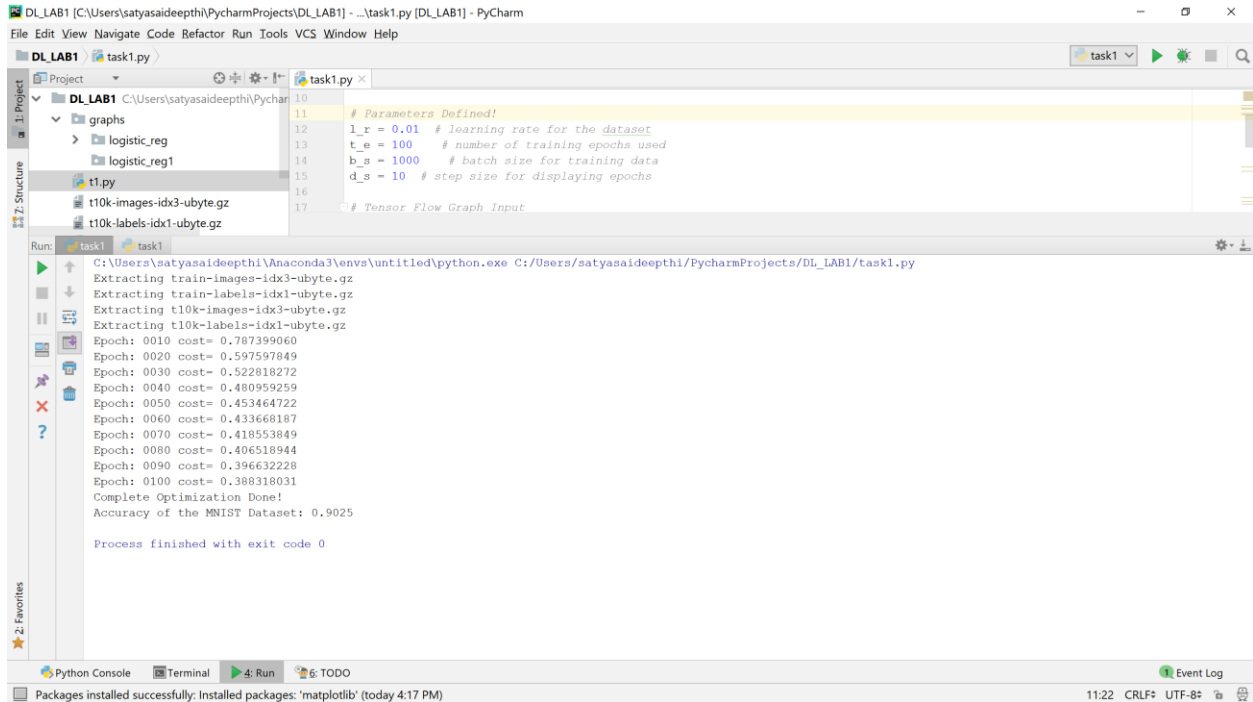
```
DL_LAB1 [C:\Users\satyasaideepthi\PycharmProjects\DL_LAB1] - ...task1.py [DL_LAB1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help

DL_LAB1 task1.py
task1.py
10
11 # Parameters Defined!
12 l_r = 0.01 # learning rate for the dataset
13 t_e = 150 # number of training epochs used
14 b_s = 1000 # batch size for training data
15 d_s = 10 # step size for displaying epochs
16
17 # Tensor Flow Graph Input

Run: task1 task1
C:\Users\satyasaideepthi\Anaconda3\envs\untitled\python.exe C:/Users/satyasaideepthi/PycharmProjects/DL_LAB1/task1.py
Extracting train-images-idx3-ubyte.gz
Extracting train-labels-idx1-ubyte.gz
Extracting t10k-images-idx3-ubyte.gz
Extracting t10k-labels-idx1-ubyte.gz
Epoch: 0010 cost= 0.787406903
Epoch: 0020 cost= 0.597607996
Epoch: 0030 cost= 0.522822097
Epoch: 0040 cost= 0.480958513
Epoch: 0050 cost= 0.453461552
Epoch: 0060 cost= 0.433674313
Epoch: 0070 cost= 0.418564809
Epoch: 0080 cost= 0.406517038
Epoch: 0090 cost= 0.396633474
Epoch: 0100 cost= 0.388324095
Epoch: 0110 cost= 0.381207262
Epoch: 0120 cost= 0.375017610
Epoch: 0130 cost= 0.369569717
Epoch: 0140 cost= 0.364726973
Epoch: 0150 cost= 0.360372411
Complete Optimization Done!
Accuracy of the MNIST Dataset: 0.9081
Process finished with exit code 0

Python Console Terminal Run TODO
Packages installed successfully: Installed packages: 'matplotlib' (today 4:17 PM) 11:22 CRLF UTF-8
```

Epochs = 100



DL_LAB1 [C:\Users\satyasaideepthi\PycharmProjects\DL_LAB1] - ..\task1.py [DL_LAB1] - PyCharm

File Edit View Navigate Code Refactor Run Tools VCS Window Help

DL_LAB1 task1.py

```
10
11 # Parameters Defined!
12 l_r = 0.01 # learning rate for the dataset
13 t_e = 100 # number of training epochs used
14 b_s = 1000 # batch size for training data
15 d_s = 10 # step size for displaying epochs
16
17 # Tensor Flow Graph Input
```

Run: task1

C:\Users\satyasaideepthi\Anaconda3\envs\untitled\python.exe C:/Users/satyasaideepthi/PycharmProjects/DL_LAB1/task1.py

Extracting train-images-idx3-ubyte.gz
Extracting train-labels-idx1-ubyte.gz
Extracting t10k-images-idx3-ubyte.gz
Extracting t10k-labels-idx1-ubyte.gz

Epoch: 0010 cost= 0.787399060
Epoch: 0020 cost= 0.597597849
Epoch: 0030 cost= 0.522818272
Epoch: 0040 cost= 0.480959259
Epoch: 0050 cost= 0.453464722
Epoch: 0060 cost= 0.433668187
Epoch: 0070 cost= 0.418553849
Epoch: 0080 cost= 0.406518944
Epoch: 0090 cost= 0.396632228
Epoch: 0100 cost= 0.388318031
Complete Optimization Done!
Accuracy of the MNIST Dataset: 0.9025

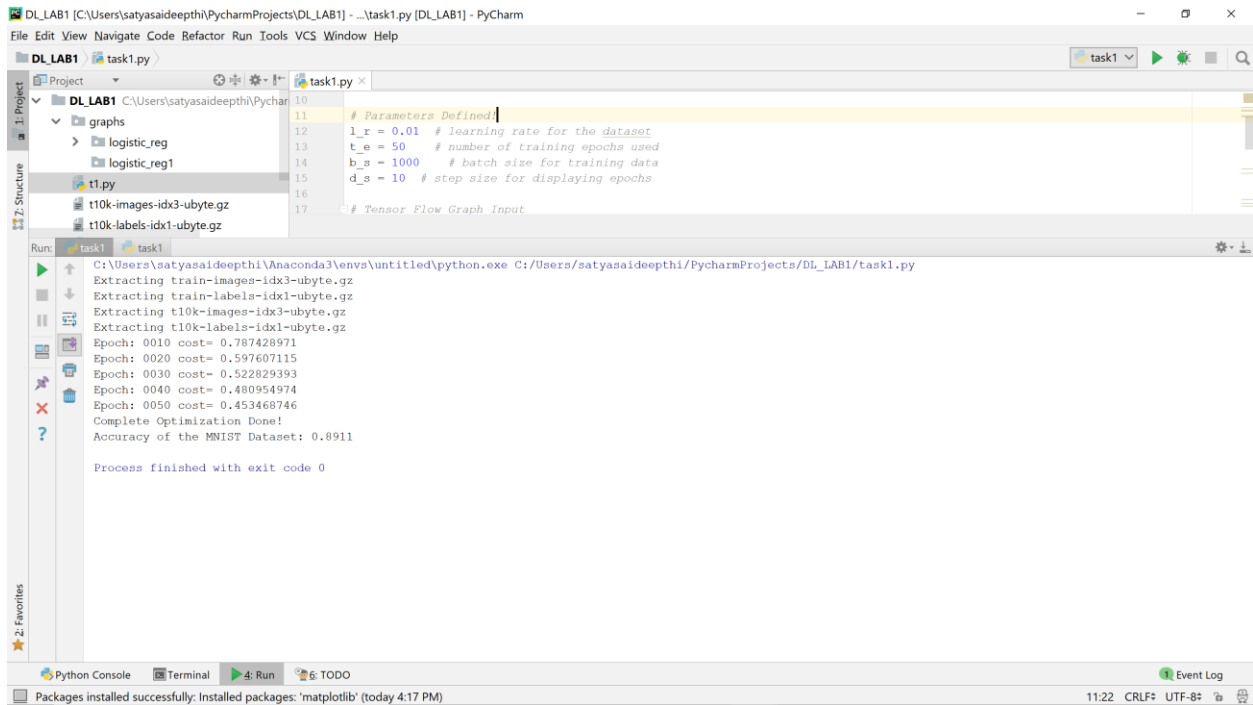
Process finished with exit code 0

Python Console Terminal Run TODO

Packages installed successfully: Installed packages: 'matplotlib' (today 4:17 PM)

11:22 CRLF UTF-8

Epochs = 50



DL_LAB1 [C:\Users\satyasaideepthi\PycharmProjects\DL_LAB1] - ..\task1.py [DL_LAB1] - PyCharm

File Edit View Navigate Code Refactor Run Tools VCS Window Help

DL_LAB1 task1.py

```
10
11 # Parameters Defined!
12 l_r = 0.01 # learning rate for the dataset
13 t_e = 50 # number of training epochs used
14 b_s = 1000 # batch size for training data
15 d_s = 10 # step size for displaying epochs
16
17 # Tensor Flow Graph Input
```

Run: task1

C:\Users\satyasaideepthi\Anaconda3\envs\untitled\python.exe C:/Users/satyasaideepthi/PycharmProjects/DL_LAB1/task1.py

Extracting train-images-idx3-ubyte.gz
Extracting train-labels-idx1-ubyte.gz
Extracting t10k-images-idx3-ubyte.gz
Extracting t10k-labels-idx1-ubyte.gz

Epoch: 0010 cost= 0.787428971
Epoch: 0020 cost= 0.597607115
Epoch: 0030 cost= 0.522829393
Epoch: 0040 cost= 0.480954974
Epoch: 0050 cost= 0.453468746
Complete Optimization Done!
Accuracy of the MNIST Dataset: 0.8911

Process finished with exit code 0

Python Console Terminal Run TODO

Packages installed successfully: Installed packages: 'matplotlib' (today 4:17 PM)

11:22 CRLF UTF-8

Conclusion

After performing Logistic Regression on MNIST Dataset, by changing hyper parameters the following conclusions are drawn from the results:

- With constant batch size and epoch number, as the learning rate increases the accuracy of the model increases.
- With constant learning rate and epoch number, as the batch size increases the accuracy of the model decreases.
- With constant learning rate and batch size, as the epoch number increases accuracy of the model increases.