

multiclass_logistic_regression

November 5, 2020

NAME: DEEPESH BHATTA

UTA_ID: 1001837305

0.1 Problem 2

(Multi-class classification, 30pts) Implement multi-class logistic regression using a soft-max function and cross entropy. Use MNIST data for your training and testing data. The dataset has 60000 training and 10000 test samples that contain images of hand-written numbers in 28x28 pixels. There are various sources you can find online to fetch the data. Please use the images within class 0 to 4 (i.e., 5 classes) only, you can do this by checking the labels given to each image. Set the maximum number of iterations to 10000. Print out accuracy, precision and recall at the end to evaluate your implementation, and write up a brief summary in your report.

```
[1]: from multiclass_logistic_regression import load_data, convert_to_one_hot, \
      ↪ logistic_regression, calc_metrics

[2]: # 10000 iterations takes too long to run
steps = 10000
learning_rate = 0.001
train_data, train_labels, test_data, test_labels = load_data()
train_labels = convert_to_one_hot(train_labels)
weights = logistic_regression(train_data, train_labels, steps, learning_rate)
acc, predictions, confusion_matrix, precision, recall = calc_metrics(test_data, \
      ↪ test_labels, weights)
print("Confusion Matrix: \n", confusion_matrix)
print("Precision: " + repr(round(precision, 4)))
print("Recall: " + repr(round(recall, 4)))
print("Accuracy: " + repr(round(acc/len(test_data), 4) * 100) + " %")
```

Confusion Matrix:

```
[[972.  0. 134.  71. 110.]
 [ 0. 981.   1.   0.   8.]
 [ 2.  91. 784.  26.  47.]
 [ 6.  63. 104. 910.  71.]
 [ 0.   0.   9.   3. 746.]]
```

Precision: 0.8553

Recall: 0.8688

Accuracy: 85.48 %

[]: