Aykut Aykut October 24th, 2018
0034039
Comp421 Homework02

In this homework, I was asked to implement a discrimination by regression algorithm for multiclass classification which can classify 20*16 pixel images to 5 distinct classes, A, B, C, D, E. In order to do that , I followed these 17 steps:

1) I read the x_data_set which contains all 320 (20*16) features of 195 images into memory and y_data_set which contains the true class labels of 195 images. In both files, each row corresponds to a image.
2) I changed the class labels with the following mapping in order to obtain one-hot encoding class label matrix using *dummy_cols* function of *fastDummies* library:
    a) A -> vector of (1, 0, 0, 0, 0)
    b) B -> vector of (0, 1, 0, 0, 0)
    c) C -> vector of (0, 0, 1, 0, 0)
    d) D -> vector of (0, 0, 0, 1, 0)
    e) E -> vector of (0, 0, 0, 0, 1)
3) Train-test split: I splitted the x_data_set and y_data_set into 2 groups, training and test. First 25 images of each class in x_data_set and y_data_set are in training set and remaning 14 images of each class in x_data_set and y_data_set are in test set.
4) I merged training sets of each class into x_training_data_set with *rbind* function and reseted their index. After that in order to apply matrix multiplication, I converted it into data matrix. I applied the same procedure for x_test_data_set, y_training labels and y_test_labels.
5) I removed useless variables which I used on the way preparing training and test data matrices, and no longer necessary.
6) I defined the *sigmoid* function using the following formula:
   $sigmoid(z) = \frac{1}{1+exp(-z)}$ $where\ z = x * w + b$
7) I defined *gradient_w* and *gradient_b* functions using the following formulas:
   $gradient_{w_i} = -\sum_{t=1}^{125} (y_{truth_{i,t}} - y_{predicted_{i,t}}) * y_{predicted_{i,t}} * (1 - y_{predicted_{i,t}}) * x_{train_t}$ where i = 1, 2, 3, 4, 5
   $gradient_{b_i} = -\sum_{t=1}^{125} (y_{truth_{i,t}} - y_{predicted_{i,t}}) * y_{predicted_{i,t}} * (1 - y_{predicted_{i,t}})$ where i = 1, 2, 3, 4, 5
8) I set the *eta, epsilon* and *seed* parameters with the same value as they are in the homework description.
   eta <- 0.01
   epsilon <- 1e-3
   set.seed(521)

9) I initialized *w* and *b* matrixes uniformly random between -0.01 and +0.01. Size of *w* is 320*5 and size of *b* is 1*5.
10) I used gradient descent algorithm for training with *sigmoid* function while predicting and the squared error function:

$$Error = 0.5 * \sum_{t=1}^{125} \sum_{i=1}^{5} (y_{truth_{i,t}} - y_{predicted_{i,t}})^2$$

11) I continued training the model until the sum of change between *old_w* and *w,* and between *old_b* and *b* is smaller than *epsilon*. In total, it took 971 iterations and error value of each iteration was recorded for error plot.
12) I plotted error value vs. iteration in a line plot.
13) I defined a *get_label* function which takes a vector and returns column index of the element equals to 1 in input vector.
14) I defined *predict_labels* function which takes a matrix *y_predict*, applies the *get_label* function to each row of *y_predict* and returns the resulting matrix.
15) I found maximum element at each row (image) using *qlcMatrix* library and changed *y_train_predict* matrix to a one-hot encoding matrix. Then I predicted a label for each training image using *predict_labels* function. I calculated the training confusion matrix and it is as follows:

```
> print(train_confusion_matrix)
           y_train
y_train_hat  1  2  3  4  5
          1 25  0  0  0  0
          2  0 25  0  0  0
          3  0  0 25  0  0
          4  0  0  0 25  0
          5  0  0  0  0 25
```

16) I predicted test images labels using *sigmoid* function at first. Then I found maximum element at each row (image) using *qlcMatrix* library and changed *y_test_predict* matrix to a one-hot encoding matrix. Then I predicted a label for each test image using *predict_labels* function. I calculated the test confusion matrix and it is as follows:

```
> print(test_confusion_matrix)
          y_test
y_test_hat  1  2  3  4  5
         1 13  1  0  0  0
         2  1 11  0  0  2
         3  0  0 14  0  0
         4  0  1  0 14  0
         5  0  1  0  0 12
```

17) I obtained the same results, error vs. iteration plot, training confusion matrix and test confusion matrix with the results given in the homework description.