Kennesaw State University

CS 7357 Neutral Nets and Deep Learning

Project 2

Description

Task 1. Classifying lr data

Download the lr data in the course web page. There are two files: lr_training.csv and lr_test.csv. The dataset includes 95 and 50 samples respectively for label of 0 and 1. So, we will solve a binary classification problem.

You will train a linear regression model using the training data (lr_training.csv) and will compute accuracy with the test data (lr_test.csv).

For Task 1, please follow the procedure:

1. Train a linear regression model
   - Find the optimal coefficients (b_opt) by the equation: b_opt = (X'X)^(-1)X'y
   - Please use "numpy.linalg.pinv" for matrix inverse. The function computes pseudo-inverse. If you use "numpy.linalg.inv", it will cause an error.
2. Display the optimal coefficients (denoted by b_opt)
3. Classify test data (lr_test.csv) with a threshold of 0.5.
   - y_pred = X_test * b_opt
   - if y_pred > 0.5, class 1, otherwise 0
4. Display the accuracy

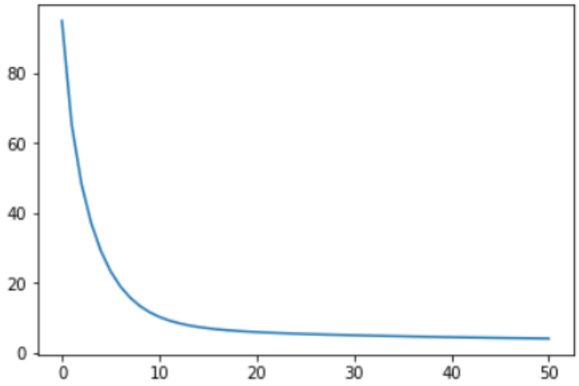
Task 2. Implementation of Gradient Descent with lr data

For Task 2, we will use the same data as Task 1. However, we will find the optimal coefficients by using "Gradient Descent" algorithm. Then, we will compare with the solution that we found in Task 1.

The procedure of Task 2 is the almost same as Task 1, but need to implement "Gradient Descent" algorithm, instead of a single line equation ((X'X)^(-1)X'y).

For the Gradient Descent algorithm, please follow the procedure:

1. Set the initial coefficient to zeros (can be any random values though)
   - Think of what the dimension of the coefficient vector is
2. Determine hyper-parameters such as learning rate and iteration numbers
3. Run "gradient descent" algorithm with the hyper-parameters and check "Learning Curve"



   * Learning curve shows whether it converges or not. X-axis shows iteration, while y-axis shows cost
   * Learning curve has to be shown as "converged", otherwise the solution may be not good.
4. Display the optimal coefficients (denoted by b_est)
5. Classify test data (MNIST_test.csv) with a threshold of 0.5.
   - y_pred = X_test * b_est
   - if y_pred > 0.5, class 1, otherwise 0
6. Display the accuracy
7. Display the total differences between b_opt and b_est
   - sum(abs(b_opt – b_est))

Submission

You have to submit the followings to D2L:

1. MS word file
   - Describe what you have done for the homework assignment.
2. Python source code file(s)
   - Must be well organized (comments, indentation, …)
   - You need to upload the “original python file (*.py)” and also its “PDF” version.
     - o For the PDF file, you can just convert the source file to PDF. One way is to print the source file and save to “PDF”.

You have to submit the files SEPERATELY. DO NOT compress into a ZIP file.