2. Linear Regression with Closed Form Solution

After seeing the problem, your classmate Alice immediately argues that we can apply a linear regression model, as the labels are numbers from 0-9, very similar to the example we learned from Unit 1. Though being a little doubtful, you decide to have a try and start simple by using the raw pixel values of each image as features.

Alice wrote a skeleton code `run_linear_regression_on_MNIST` in main.py, but she needs your help to complete the code and make the model work.

Closed Form Solution of Linear Regression

5.0/5.0 points (graded)

To solve the linear regression problem, you recall the linear regression has a closed form solution:

$$\theta = (X^T X + \lambda I)^{-1} X^T Y$$

where I is the identity matrix.

Write a function `closed_form` that computes this closed form solution given the features X, labels Y and the regularization parameter λ.

Available Functions: You have access to the NumPy python library as `np`; No need to import anything.

```
def closed_form(X, Y, lambda_factor):
    """
    Computes the closed form solution of linear regression with L2 regu

    Args:
        X - (n, d + 1) NumPy array (n datapoints each with d features p
        Y - (n, ) NumPy array containing the labels (a number from 0-9)
            data point
        lambda_factor - the regularization constant (scalar)
    Returns:
        theta - (d + 1, ) NumPy array containing the weights of linear
        represents the y-axis intercept of the model and therefore X[0]
    """
    # YOUR CODE HERE
    from numpy.linalg import inv
```

Press ESC then TAB or click outside of the code editor to exit

Correct

```
def closed_form(X, Y, lambda_factor):
    """
    Computes the closed form solution of linear regression with L2 regularizati

    Args:
        X - (n, d + 1) NumPy array (n datapoints each with d features plus the 
        Y - (n, ) NumPy array containing the labels (a number from 0-9) for eac
            data point
        lambda_factor - the regularization constant (scalar)
    Returns:
        theta - (d + 1, ) NumPy array containing the weights of linear regressi
        represents the y-axis intercept of the model and therefore X[0] = 1
    """
    I = np.identity(X.shape[1])
    theta = np.linalg.inv(X.T @ X + lambda_factor * I) @ X.T @ Y
    return theta
```

Test results

CORRECT

See full output

See full output

Submit

You have used 1 of 25 attempts

Answers are displayed within the problem

Test Error on Linear Regression

1.0/1.0 point (graded)

Apply the linear regression model on the test set. For classification purpose, you decide to round the predicted label into numbers 0-9.

Note: For this project we will be looking at the error rate defined as the fraction of labels that don't match the target labels, also known as the "gold labels" or ground truth. (In other context, you might want to consider other performance measures such as precision and recall, which we have not discussed in this course).

Please enter the **test error** of your linear regression algorithm for different λ (copy the output from the `main.py` run).

$\text{Error}|_{\lambda=1} =$ 0.7697 **Answer:** 0.7697

$\text{Error}|_{\lambda=0.1} =$ 0.7698 **Answer:** 0.7698

$\text{Error}|_{\lambda=0.01} =$ 0.7702 **Answer:** 0.7702

Submit You have used 1 of 20 attempts

Answers are displayed within the problem

What went Wrong?

1.0/1.0 point (graded)

Alice and you find that no matter what λ factor you try, the test error is large. With some thinking, you realize that something is wrong with this approach.

- [ ] Gradient descent should be used instead of the closed form solution.
- [x] The loss function related to the closed-form solution is inadequate for this problem.
- [ ] Regularization should not be used here.

Solution:

The closed form solution of linear regression is the solution of optimizing the mean squared error loss. This is not an appropriate loss function for a classification problem.

Submit You have used 2 of 2 attempts

Answers are displayed within the problem

Discussion

Hide Discussion

Topic: Unit 2 Nonlinear Classification, Linear regression, Collaborative Filtering (2 weeks):Project 2: Digit recognition (Part 1) / 2. Linear Regression with Closed Form Solution

Add a Post

Show all posts by recent activity

- [STAFF] Not able to read the output from main.py — 8
- [Staff] Error submitting answer — 3
  I didn't realize that on the *What went wrong* question it was possible to submit multiple an...
- what went wrong — 7
  Can you give me some hints on the third discussion problem? My intuition is that it can be no...
- closed form solution: When to dot() and when to multiply? — 7
  I'm trying to translate the function into a Python script. However, I've realized I don't really un...
- Keep getting numpy.linalg.LinAlgError: Singular matrix — 9
  I am very frustrated because I cannot pass the closed form test in $test.py$ since last 2 hours...
- [Staff] Memory error in matrix multiplication — 2
  File "C:\Users\firda\PycharmProjects\DigitRecognition\part1\linear_regression.py", line 23, in ...
- Memory error — 20
  HI i am getting memory error while running this code. any idea how to resolve this
- What is the purpose of stacking the training set with a column of ones? — 2
  Hey guys, I am trying to understand the reason for adding a column of ones > train_x_bias = ...
- Value Error — 5
  I don't understand why I get this error ? ValueError: operands could not be broadcast togeth...
- Solution penalises theta_0 — 1
  If I am not mistaken, when a column of ones is added to the left of X and the first entry of the...
- Is np.transpose the correct way to transpose for this problem? — 4
  Is np.transpose the correct way to transpose for this problem?
- Running into numpy.linalg.LinAlgError: Singular matrix — 7
  I just try to do the inner term $X.transpose().\dot{X}$ and I keep running into the exception ...
- FileNotFoundError: [Errno 2] No such file or directory: '../Datasets/mnist.pkl.gz' — 3
  I have redownloaded the package and made the changes as per : https://courses.edx.org/cou...

Learn About Verified Certificates