6. Changing Labels

We now wish to classify the digits by their (mod 3) value, such that the new label $y^{(i)}$ of sample i is the old $y^{(i)} \pmod 3$. (Reminder: Return the `temp_parameter` to be 1 if you changed it for the last section)

You will be working in the file `part1/main.py` and `part1/softmax.py` in this problem

Using the Current Model - update target

3.0/3.0 points (graded)

Given that we already classified every $x^{(i)}$ as a digit, we could use the model we already trained and just calculate our estimations (mod 3).

Implement `update_y` function, which changes the old digit labels for the training and test set for the new (mod 3) labels.

Available Functions: You have access to the NumPy python library as `np`

```
def update_y(train_y, test_y):
    """
    Changes the old digit labels for the training and test set for the new (mod 3)
    labels.

    Args:
        train_y - (n, ) NumPy array containing the labels (a number between 0-9)
                  for each datapoint in the training set
        test_y - (n, ) NumPy array containing the labels (a number between 0-9)
                 for each datapoint in the test set

    Returns:
        train_y_mod3 - (n, ) NumPy array containing the new labels (a number between 0-2)
                       for each datapoint in the training set
        test_y_mod3 - (n, ) NumPy array containing the new labels (a number between 0-2)
                      for each datapoint in the test set
```

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

Correct

Test results

Hide output

CORRECT

Test: all digits

Testing that all digits are transformed properly

Output:

```
(array([0, 1, 0, 1, 0, 2, 2, 2, 1, 0]), array([1, 0, 0, 0, 2, 1, 2, 1, 0, 2]))
Test completed
```

Hide output

Submit You have used 1 of 20 attempts

Using the Current Model - compute test error

3.0/3.0 points (graded)

Implement `compute_test_error_mod3` function, which takes the test points `X`, their correct labels `Y` (digits (mod 3) from 0-2), `theta`, and the `temp_parameter`, and returns the error.

Example:

	Estimated Y	Estimated Y (mod 3)	Correct Y	Correct Y (mod 3)
x_1	9	0	8	2
x_2	6	0	6	0
x_3	5	2	8	2

The error of the regression with the original labels would be 0.66667

However, the error of the regression when comparing the (mod 3) of the labels would be 0.33333

Available Functions: You have access to the NumPy python library as `np` and to the `get_classification` function from the project release

```
def compute_test_error_mod3(X, Y, theta, temp_parameter):
    """
    Returns the error of these new labels when the classifier predicts the digit. (mod 3)

    Args:
        X - (n, d - 1) NumPy array (n datapoints each with d - 1 features)
        Y - (n, ) NumPy array containing the labels (a number from 0-2) for each
            data point
        theta - (k, d) NumPy array, where row j represents the parameters of our
                model for label j
        temp_parameter - the temperature parameter of softmax function (scalar)

    Returns:
        test_error - the error rate of the classifier (scalar)
    """
    #YOUR CODE HERE
```

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

Correct

Test results

Hide output

CORRECT

Test: 1

Testing random data points

Output:

```
Error: 0.6
```

Test: 2

Testing random data points

Output:

```
Error: 0.4
```

Test: 3

Testing random data points

Output:

```
Error: 1.0
```

Test: all correct

Testing correct

Output:

```
Error: 0.0
```

Test: all incorrect

Testing correct

Output:

```
Error: 1.0
```

Hide output

Submit You have used 2 of 20 attempts

Using the Current Model - test error

1/1 point (graded)

Find the error rate of the new labels (call these two functions at the end of `run_softmax_on_MNIST`). See the functions' documentation for detailed explanations of the inputs and outputs.

Error rate for labels mod 3: 0.07679999999999998 ✔

Submit You have used 1 of 20 attempts

✔ Correct (1/1 point)

Retrain with New Labels

3/3 points (graded)

Now suppose that instead we want to retrain our classifier with the new labels. In other words, rather than training the model to predict the original digits and then taking those predictions modulo 3, we explicitly train the model to predict the digits modulo 3 from the original image.

How do you expect the performance to change using the new labels?

- Increase
- Decrease ✔
- Stay the same

Implement `run_softmax_on_MNIST_mod3` in **main.py** to perform this new training; report the new error rate.

Error rate when trained on labels mod 3: 0.18720000000000003 ✔

Submit You have used 2 of 2 attempts

✔ Correct (3/3 points)

Discussion

Hide Discussion

Topic: Unit 2 Nonlinear Classification, Linear regression, Collaborative Filtering (2 weeks):Project 2: Digit recognition (Part 1) / 6. Changing Labels

Add a Post

Show all posts ▼ by recent activity ▼

?	Using the Current Model - compute test error The function `get_classification` returns an np array of numbers between 0 and 9. We can't call update_y in the function compute_test_error_mod3. That means we should w...	15
?	Using the Current Model - update target We're supposed to relabel the original data set (numbers between 0 and 9) to a new data set (numbers between 0 and 2) according to what rule ??	7
💬	(mod 3) value Can you clarify what does it mean the mod 3 value?. Not sure this was explained before?	3
💬	Regarding recommendation algorithms. More classes in terms of information theory, less entropy ? Does anyone understand in terms of information theory, what does it mean to have more or less classes ? Do more classes always represent more knowledge about the class...	2

Learn About Verified Certificates