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 $y^{(i)} \pmod 3$. An example is provided in the next section. (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
    labels.

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

    Returns:
        train_y_mod3 - (n, ) NumPy array containing the new labels (a n
                       for each datapoint in the training set
        test_y_mod3 - (n, ) NumPy array containing the new labels (a nu
```

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

Correct

```
def update_y(train_y, test_y):
    """
    Changes the old digit labels for the training and test set for the new (mod
    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 be
                       for each datapoint in the training set
        test_y_mod3 - (n, ) NumPy array containing the new labels (a number betw
                      for each datapoint in the test set
    """
    return np.remainder(train_y, 3), np.remainder(test_y, 3)
```

Test results

See full output

CORRECT

See full output

Submit You have used 1 of 25 attempts

Answers are displayed within the problem

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.

In comparison, the error of the regression with the labels after the “mod 3" operation would be 0.33333.

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

```
def compute_test_error_mod3(X, Y, theta, temp_parameter):
    """
```

```
    Returns the error of these new labels when the classifier predicts

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

    Returns:
        test_error - the error rate of the classifier (scalar)
    """
```

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

Correct

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

    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 eac
            data point
        theta - (k, d) NumPy array, where row j represents the parameters of ou
                model for label j
        temp_parameter - the temperature parameter of softmax function (scalar)

    Returns:
        test_error - the error rate of the classifier (scalar)
    """
    assigned_labels = get_classification(X, theta, temp_parameter)
    return 1 - np.mean(np.remainder(assigned_labels,3) == Y)
```

Test results

See full output

CORRECT

See full output

Submit You have used 2 of 25 attempts

Answers are displayed within the problem

Using the Current Model - test error

1.0/1.0 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: 0.0768 ✔ **Answer:** 0.0768

Submit You have used 1 of 20 attempts

Answers are displayed within the problem

Retrain with New Labels

3.0/3.0 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: 0.1872

Answer: 0.1881

Solution:

We are trying to find common features of all numbers that have the same mod 3 value, however a lot of them look widely different, so it is harder to separate the data set into 3 groups since, for example, 2 does not share many features with 5 or 8. Therefore one would expect the performance to decrease, and this is what happens.

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) / 6. Changing Labels

Add a Post

Show all posts by recent activity

- Why take the modulus? 6
  I managed this section, but am confused at the motivation behind the transformation. I gues...
- [STAFF] Using the Current Model - compute test error 4
  Is nt it that we just use compute test error mod 3 instead of compute test error function? I ge...

☑	Using the Current Model - test error I did everything else, except this one. I am using run_softmax_on_MNIST(temp_parameter=1),...	9
☑	Feature Vector Size	3
💬	How can the answer to "How do you expect the performance to change using the new labels?" ever be wrong? You are asking for an opinion.	3
☑	Using the Current Model - test error Guys, I am lost here. Tried pretty much everything. I have adjusted the k value to 3 in my soft...	3
☑	[Staff] Updating Target - Current Model Appreciate any support for how to use only the information in this function (Y train and Y test...	7
💬	Expected output of compute_test_error_mod3	1
💬	What is the point of retraining with new labels? I do not understand the motive behind retraining the model with the changed labels?	1
?	Correct answer marked incorrect. Hello everybody, Can someone tell me why am I getting my answer marked as incorrect? the ...	4
?	When running the mod3 version of the model, should we update the k parameter too? If we are training on mod3 data, we only need k = 3, correct?	2
💬	meaning of "increased performance" I'm assuming that increased performance means it runs in less time. True?	2

Learn About Verified Certificates