CS584 – Assignment 3

Discriminative learning

A20364639

Yung Chi Liu

* Problem statement

In this assignment, the main topic is to implement Discriminative learning algorithms. First topic in this assignment is Logistic regression. I will implement the logistic regression algorithm for 2 class discrimination, k class discrimination and non-linear combinations of input. And test with 2 data set, one is called balance data set a numeral data with 4 features and 3 class, the other is MNIST data ( graphic data). Second topic is Multilayer perception. In this part I’ll derive the update equations for a tow layer feed forward MLP with a single output.

1. Logistic regression:

* Design and Parameters :

(a). Logistic Regression for 2 class:

First, define the parameters I need to use

Sigmoid Function:

and guess first with random float which close to 0. Which n is the number of features with x.

then define the update equation with Gradient Descent which is learning rate

then start to train, until update 1000 times or the training error rate is lower than 1/10.

Compute the classification for test data with Sigmoid Function

(b). Use non-linear combinations of inputs:

Take 2 features x for example:

with x1 combination will generate Z

Then use the same strategy as Logistic regression for 2 class

(c). Logistic Regression algorithm for K-class discrimination:

In this part, replace sigmoid function with soft max function and use one against all others strategy.

Soft Max function:

And classification is :

Update function with training :

After all these done, do the classification for testing data. And compute the accuracy for this algorithm.

* Data Sets:

Data sets:

I used two data sets in this assignment. First one in balance data set with 4 features in each example.

Attribute Information:

1. Class Name: 3 (L, B, R)

2. Left-Weight: 5 (1, 2, 3, 4, 5)

3. Left-Distance: 5 (1, 2, 3, 4, 5)

4. Right-Weight: 5 (1, 2, 3, 4, 5)

5. Right-Distance: 5 (1, 2, 3, 4, 5)

Class Distribution:

1. 46.08 percent are L

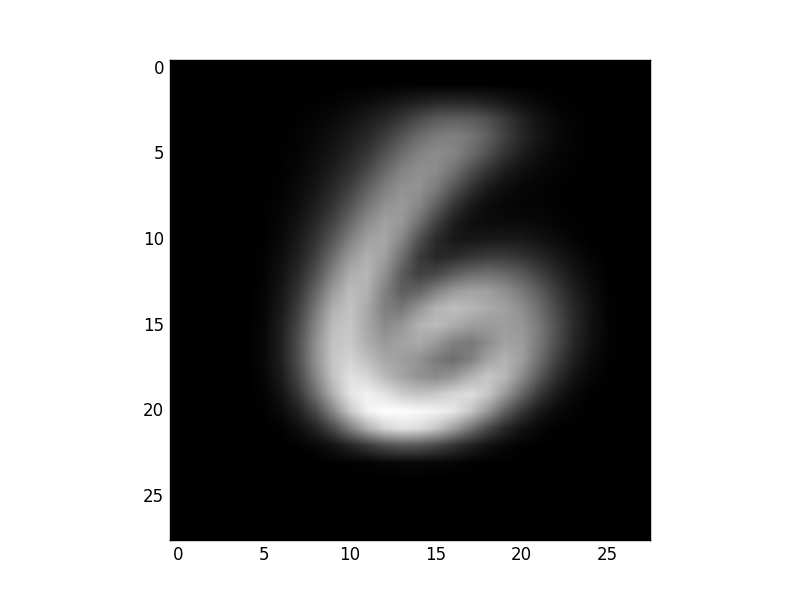
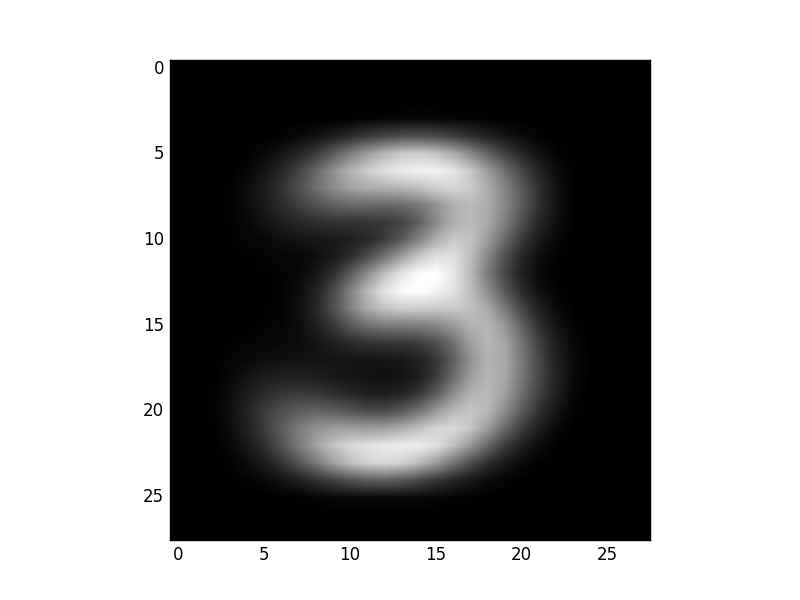
2. 07.84 percent are B

3. 46.08 percent are R

In my code, I change class name( L,B,R) to (1,2,0) and remove class 2 as 2 class classification.

And Second data set is MNIST data, which is graphic data set. Graphic is label with handwriting number from 0 to 9. And I choose label 0 and 1 for 2 class case, label 0, 1 and 2 for k class case.

Data look like:



The image data is a 28X28 size matrix data. In this code which is 1x784 data.

In my implementation, I extract 900 examples from training data 100 examples from testing data for each class.

* Result and estimate:

(a). 2 class Logistic Regression

Test with different learning rate

10 fold validation

Result with MNIST data.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Avg. Testing Accuracy | 0.998 | 0.903 | 0.8865 |
| Avg. Gradient descent time | 2 | 14.8 | 139.6 |

As the result when use smaller learning rate the accuracy of the testing decreased, which is not as expected. One reason could be my Update function, stop to update when testing error rate is lower than 0.1. As the learning rate smaller the error rate just lower than 0.1 is much more close to 0.1 compared to larger learning rate. For example, with = 0.0001 the error rate just pass 0.1 is 0.965 and = 0.00001 is 0.9985. Which small learning rate get lower error rate in this case.

On way to solve this problem could use to control the update function. Like when is small than 0.001 stop update the

10 fold validation

Result with balance data.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Avg. Testing Accuracy | 0.791197822142 | 0.90647307925 | 0.727374470659 |
| Avg. Gradient descent time | 6.7 | 56.3 | 779.8 |

As this case when is not taken, because some of the test case not converge when updated 1000 times. The previews two case is good as learning rate smaller get better testing accuracy.

(b). Use non-linear combinations of inputs:

As this case I only get result from balance data. The MNIST data is overflow when combination 1 time with 784 elements.

10 fold validation

Result with balance data.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Avg. Testing Accuracy | 0.85148215366 | 0.840804597701 | 0.913067150635 |
| Avg. Gradient descent time | 7.4 | 6.7 | 48.1 |

As the result when learning rate smaller the accuracy is better.

Compare with the test without use non-linear combinations inputs, the accuracy increased. It shows with more complex input, made data to high dimension get more accurate result.

(C). Logistic Regression algorithm for K-class discrimination:

(a). 3 class Logistic Regression

Test with different learning rate

10 fold validation

Result with MNIST data.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Avg. Testing Accuracy | 0.915666666667 | 0.784 | 0.752 |
| Avg. Gradient descent time | 1.5 | 8.7 | 81.4 |

The result of 3 classes logistic regression. The Best performance is occurred when learning rate is 0.0001.

10 fold validation

Result with balance data.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Avg. Testing Accuracy | 0.883363581367 | 0.862467997952 |
| Avg. Gradient descent time | 4.8 | 1000.0 |

The Result of 3 classes regression with balance data set. The accuracy is not perfect; one reason is the data set. With 46.08% as class 0, 46.08% as class 1, there only 7.84% as class 2. That makes the harder to converge when use learning rate as 0.00001.

* Multilayer Regression:

(a). Derive the backpropagation update equations for the weight of the output by minimizing the error function.

Error function:

with output layer and hidden layer both sigmoid function:

Use log likelihood, use max log likelihood estimate.

With x-1, objective need to get min. log likelihood.

Compare to likelihood in class which is same at classification equation. But with a little difference in output layer’s update equations.

* Reference:

1. data set UCI website. http://archive.ics.uci.e du/ml/

2. Gady Agam. Lecture note. Discriminative learning

3. Index of Logistic Regression Datasets. https://www.umass.edu/statdata/statdata/stat-logistic.html