# CmpE 462 Spring - 2020 Project1 Report

**Alperen Bağ, Cihat Kapusuz, İbrahim Özgürcan Öztaş**

April 27, 2020

# Contents

# Chapter 1

# Introduction

This report is constructed for the first project given in course CmpE 462 Machine Learning. It is given as a group project which can be done by two or three people. Our group consists of three people, namely Cihat Kapusuz, Alperen Bağ and İbrahim Özgürcan Öztaş.

The given task is creating an identifier that distinguishes digit 1 from digit 5 based on logistic regression approach. To perform such a task, we've created a three step plan to compartmentalize the project into several steps that will result in a comprehensive report.

First step is the feature extraction, which helps us to find distinguishable, unique traits between digit 1 and digit 5 that aids us classifying the digits. With the extracted features, we can label each image as digit 1 or digit 5 as successful as possible.

Second step is the implementing the logistic regression algorithm, based on 2-D images provided by the data set MNIST. In this step, we prepare the data for our extracted features, and after that, we've implemented several beneficial functions to aid us calculating the logistic regression for the given training data.

Third step is simply evaluating the test data according to the model trained in the second step. In first step, we're required to extract 2 set of features, one is given by our instructor and the other one is left into our judgment. The one that is required by our instructor consists of 2 features, average intensity and symmetry. The one that is left into our judgment consists of variance of x coordinates of white pixels, and the distance on x axis between maximum valued white pixel and minimum valued white pixel. With these sets of features, we've trained our model to identify the given images.

In conclusion, this report has been partitioned into several chapters and each chapter express substantial information regarding to the relevant code cell in Jupyter Notebook, provided alongside with this report.

# Chapter 2

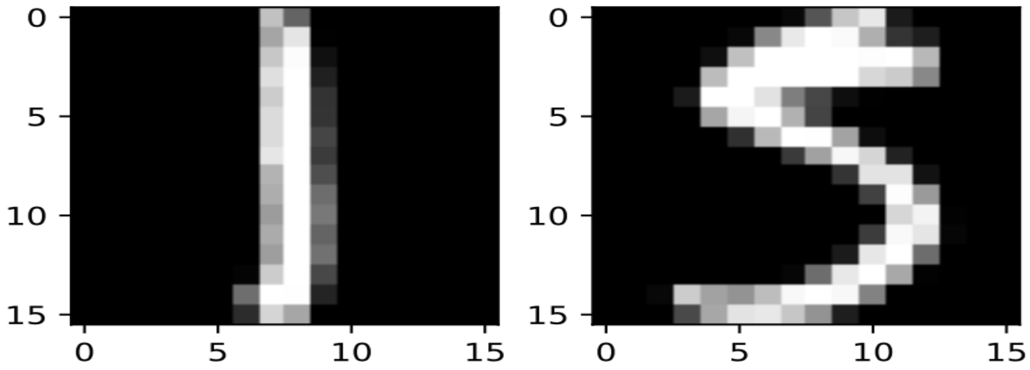
# Feature Extraction

## 2.1 Importing libraries and data[1][2]

In this chapter, first of all we've imported the libraries numpy and matplotlib. After that, we've loaded the partition of data set MNIST given by our instructor in two partitions, one as training data and one is test data. First and second cells are relevant to this process.

## 2.2 Displaying example digits[3]

After importing the necessary libraries and data, we've required to display one image per digit to show the content of our data set.

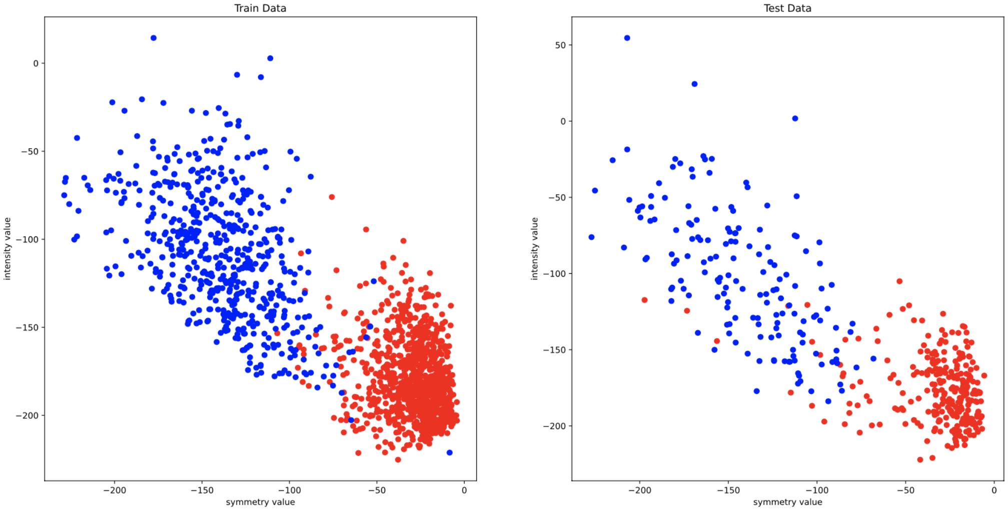


This process is performed in cell 3.

## 2.3 Implementing first representation[4][5]

Now, we can extract the average intensity and symmetry values for each image provided in data set. To do so, we've taken summation of test and training data separately and divided into total number of test and training data, respectively.

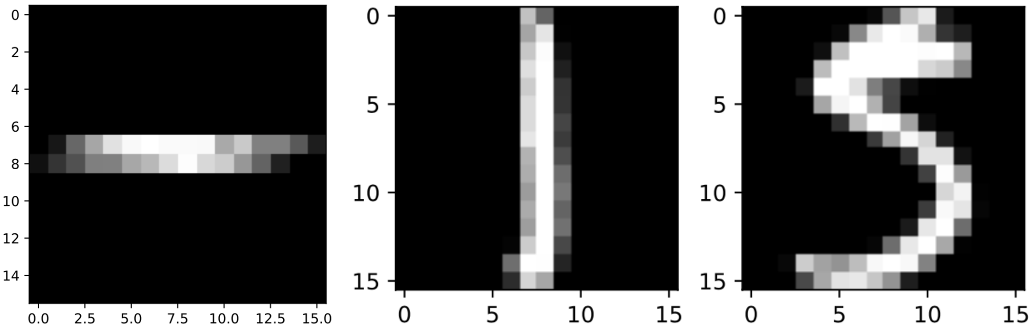
After that, we want to see the results of first representation of feature extraction and we've implemented a plot that illustrates the features on both axis and the label for the image as blue and red. By doing so, we've realized that by using two features, we could actually distinguish digit 1 from digit 5 with a very promising accuracy.



The calculation is performed in cell 4, and the plotting is performed in cell 5.

## 2.4 Bonus: Displaying one of the certain outlier data[6]

We want to show one of the outlier data that is labeled as digit 5 is neither a digit 5 nor digit 1. Hence, the data set has several flaws that reduces the accuracy of the model.



It is obvious to see that the leftmost image is certainly not a digit 5 and certainly not a digit 1. Yet, we need to think the fact that the data set we use or another data set that can be used in the project may include incorrectly labeled data and it is a fact of life to have imperfections.

## 2.5 Implementing second representation[7]

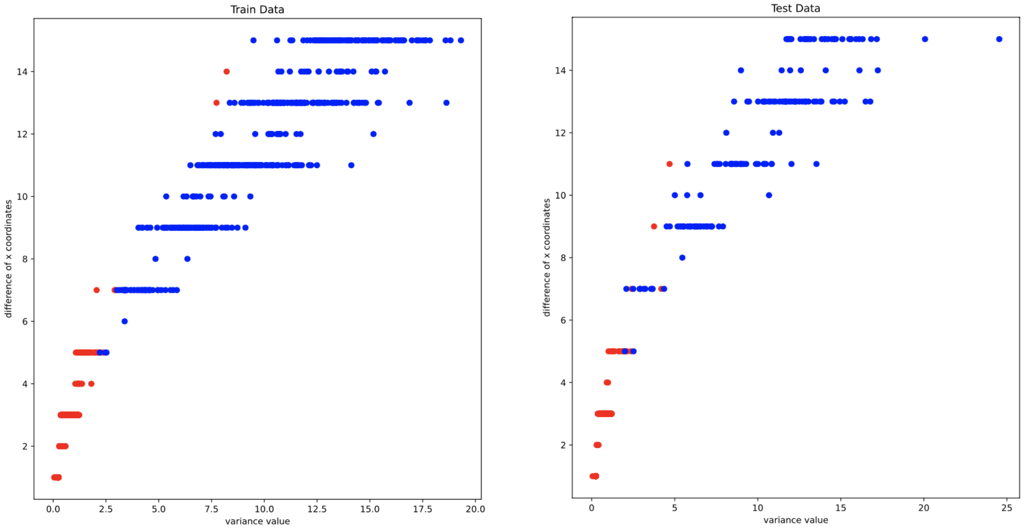
Implementing first representation is quite a success, yet it is not enough to find the most suitable model to compose a very successful digit recognition tool. Thus, we've acquired a new set of features to increase the distinguishing accuracy.

Our first feature is calculating the variance among the white valued pixels, which have intensity greater than 0, on the x axis. By doing so, we've hoped to find out that the images of digit 1 should have a small variance, since digit 1 is generally illustrated as a one narrow long line and a small bump with an angle approximately 45° joined from upmost part of the narrow line.

On the other hand, digit 5 has more variance than 1, since it is mostly illustrated as a one broad line at the top, joined with a short narrow line from the leftmost point of the broad line, and a crescent joined with the bottom most point of the short narrow line. Hence, variance among white valued pixels at the x axis is a very instrumental feature to consider.

Our second feature is the absolute difference, which is the difference between maximum and minimum values, among the white valued pixels on x axis. By doing so, we've hoped to observe that since illustrating digit 5 generally requires a more broad approach than illustrating digit 1.

After deducting these two new features, we want to see the newly distinguished version of our test and training data to perceive the behaviours of our data sets.



It is obvious to understand that digit 1 and digit 5 can be separated with a very high accuracy with the consideration of newly composed set of features.

It is performed by cell 7.

# Chapter 3

# Logistic Regression

In this chapter, we've initiated to train our identifier model. Since the images in the data set is constructed as 2-D images, our model consists of 3 coefficient, namely $w_0$, $w_1$, and $w_2$. In our data set, we have $x_1$ and $x_2$ values, hence it is a necessity to append a 1 into the extracted set of values for each image in our data set, because the constant value doesn't depend on the features that are taken into consideration.

After preparation, we've constructed several helper functions to aid the training process. And then, we've implemented our logistic regression model via taking the new value based on the gradient descent on the previous point at each step, until the model converges to the global minimum or the epoch value is reached.

Then, it is requested to add a regularization into our model to increase the accuracy of the model. To do so, it is again requested to use a 5-fold cross validation to pick the lambda which gives the highest accuracy. In this chapter, we've done 5-fold cross validation for both sets of features extracted in chapter 2.

## 3.1 Data Preparation[8]

To be able to use the data set in correct manner, we've appended a list of 1's into the train and test data sets, each. It synchronizes the data with the desired coefficients, since we know that for each feature, we have a coefficient, and there is a constant value that doesn't depend on any feature. Yet it has an impact on the overall model, so we've added a 1 into the $w$ for the constant value $x_0$ to perform a successful training.

Preparation is done in cell 8.

## 3.2 Implementing helper functions[9]

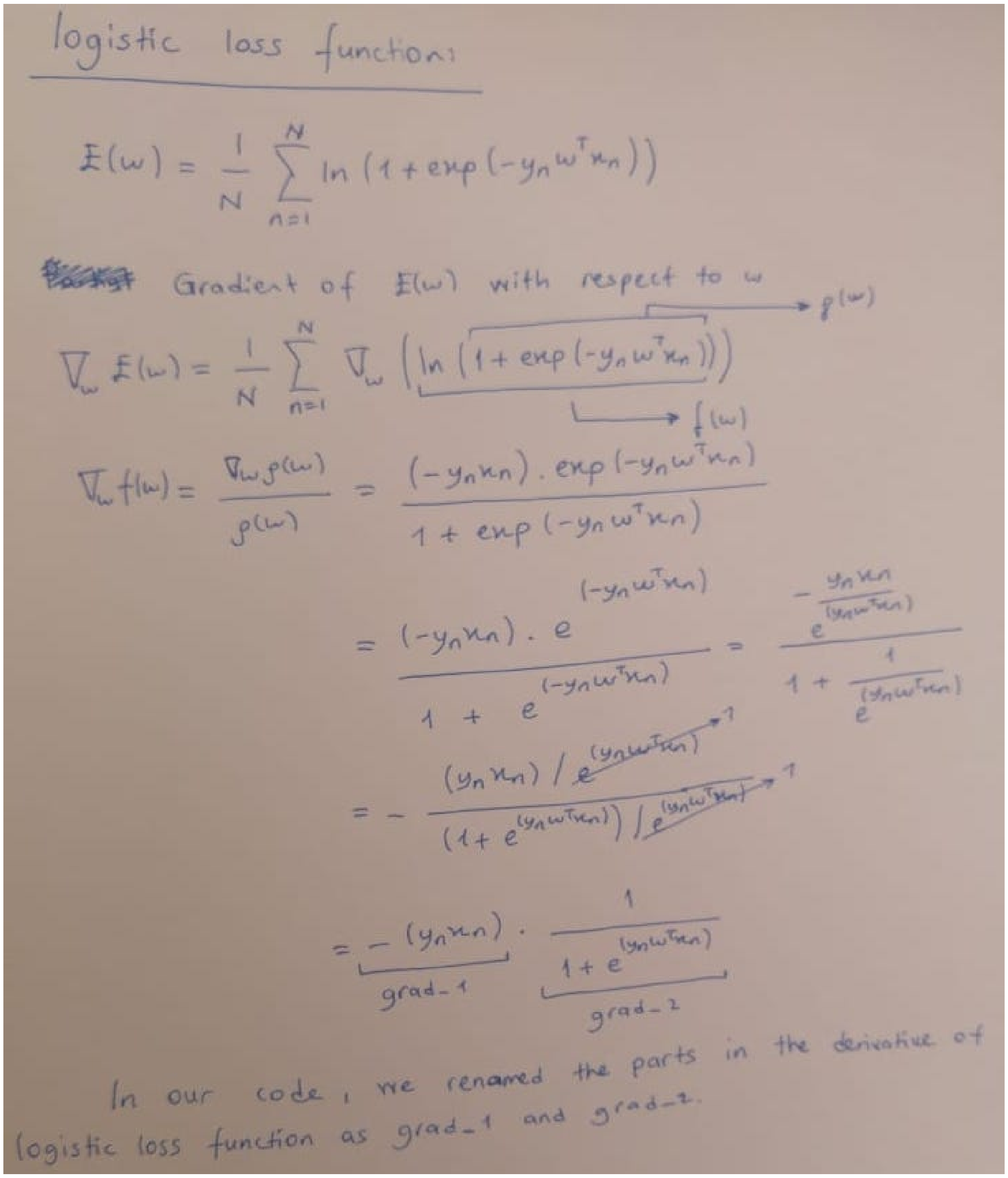
Training the logistic regression model requires logistic loss calculation and the sigmoid function to validate after training. In this section, these functions are implemented and used in further sections.

In $sigmoid(x)$ function, we've returned the corresponding value in sigmoid function for given value x.

In $logistic\_loss(y\_pred, y\_true, reg\_lambda, w)$ function, we've calculated the overall loss for the given data and returned the result.

In $fit\_logistic\_model(X, y, epoch, learning\_rate, reg\_lambda)$ function, we've trained our model at *epoch* steps to come to a convergence point for our model. If there's a regularization, *reg_lambda* is taken into consideration. Else, it is accepted as zero.

In $fit\_logistic\_model$ function, we've used the gradient descent which requires the derivative of the logistic loss function at each step. We've deducted the derivative by calculations made on paper and added the proof below.



logistic loss function:

$$E(w) = \frac{1}{N} \sum_{n=1}^{N} \ln \left(1 + \exp(-y_n w^T x_n)\right)$$

Gradient of $E(w)$ with respect to $w$

$$\nabla_w E(w) = \frac{1}{N} \sum_{n=1}^{N} \nabla_w \Big( \underbrace{\ln \big( \overbrace{1 + \exp(-y_n w^T x_n)}^{g(w)} \big)}_{f(w)} \Big)$$

$$\nabla_w f(w) = \frac{\nabla_w g(w)}{g(w)} = \frac{(-y_n x_n) . \exp(-y_n w^T x_n)}{1 + \exp(-y_n w^T x_n)}$$

$$= \frac{(-y_n x_n) . e^{(-y_n w^T x_n)}}{1 + e^{(-y_n w^T x_n)}} = \frac{-\frac{y_n x_n}{e^{(y_n w^T x_n)}}}{1 + \frac{1}{e^{(y_n w^T x_n)}}}$$

$$= - \frac{(y_n x_n) / \cancelto{1}{e^{(y_n w^T x_n)}}}{(1 + e^{(y_n w^T x_n)}) / \cancelto{1}{e^{(y_n w^T x_n)}}}$$

$$= \underbrace{-(y_n x_n)}_{grad\_1} . \underbrace{\frac{1}{1 + e^{(y_n w^T x_n)}}}_{grad\_2}$$

In our code, we renamed the parts in the derivative of logistic loss function as grad_1 and grad_2.

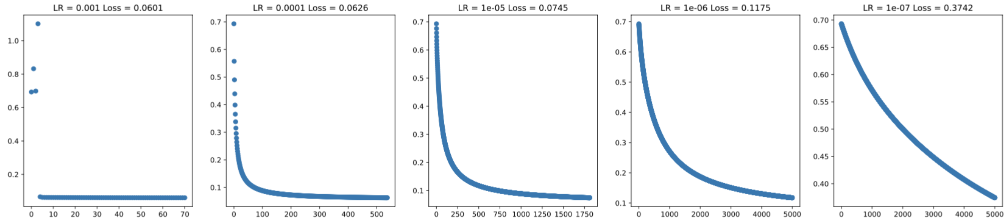
In our code, we renamed the parts in the derivative of logistic loss function as grad_1 and grad_2.

This is performed in cell 9.

## 3.3 Model Training on Representation 1[10]

After several preparations and implementations of assisting functions, it is time to train our model for feature set 1, that depends on the features required from out instructor.

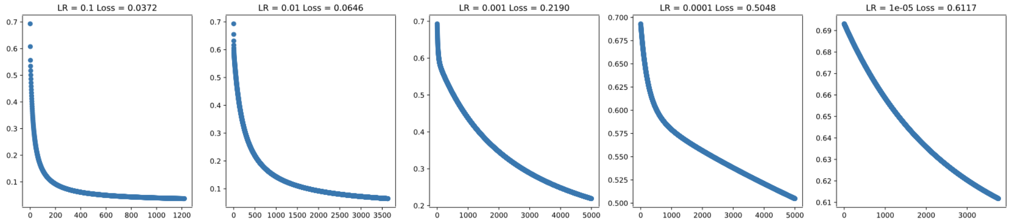
For different learning rates, we've trained our model and plotted our results for each learning rate. We've picked 5 different learning rate and collected 5 graphs.



This is performed in cell 10.

## 3.4 Model Training on Representation 2[11]

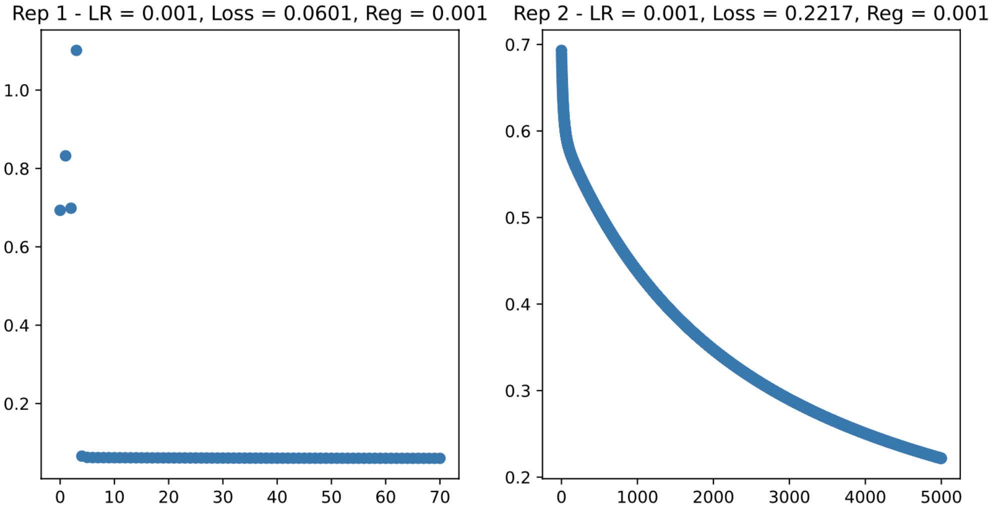
For different learning rates, we've trained our model for representation 2 and plotted our results for each learning rate. We've picked 5 different learning rate and collected 5 graphs.



This is performed in cell 11.

## 3.5 Model Training on Representation 1 and Representation 2 with $l_2$ regularization[12]

For one learning rate, we've retrained our model but this time, regularization is taken into consideration. After training, we've plotted the results to have a comprehension of the effects that regularization has brought into the model.



This is performed in cell 12.

## 3.6 Cross Validation Helper Functions[13]

In this section, we've implemented several functions to utilize the separation of data and result validation.

The $accuracy(X, y, w)$ function returns the accuracy of the given data by multiplying each value in $w$ with $X$ and compares with $y$. If the calculation matches with the given $y$ value, it counts as success and else it counts as failure.

The $merger(input_list, index)$ function returns a list consists of the all sub lists except the one that is indexed with $index$. This allows us to create input variations according to the cross validation process.

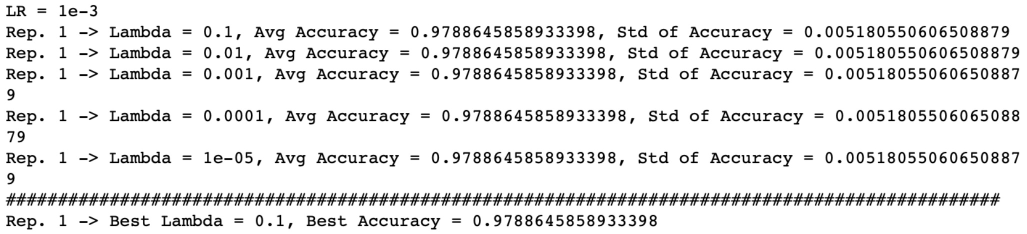
This is performed in cell 13.

## 3.7 Cross Validation for Representation 1[14][15]

First, we've used our helper function $merger()$ to organize the representation 1 training data set into 5 different sets which doesn't consist the corresponding sub list according to their index value. We've also organized the corresponding truth values to match the new sets.

This is performed in cell 14.

And then, we've initiated our cross validation process, which consists of model training and extracting accuracy for each fold. And then, we've printed out the results to see which lambda value fits more perfectly than the rest.



```
LR = 1e-3
Rep. 1 -> Lambda = 0.1, Avg Accuracy = 0.9788645858933398, Std of Accuracy = 0.005180550606508879
Rep. 1 -> Lambda = 0.01, Avg Accuracy = 0.9788645858933398, Std of Accuracy = 0.005180550606508879
Rep. 1 -> Lambda = 0.001, Avg Accuracy = 0.9788645858933398, Std of Accuracy = 0.00518055060650887
9
Rep. 1 -> Lambda = 0.0001, Avg Accuracy = 0.9788645858933398, Std of Accuracy = 0.0051805506065088
79
Rep. 1 -> Lambda = 1e-05, Avg Accuracy = 0.9788645858933398, Std of Accuracy = 0.00518055060650887
9
################################################################################################
Rep. 1 -> Best Lambda = 0.1, Best Accuracy = 0.9788645858933398
```

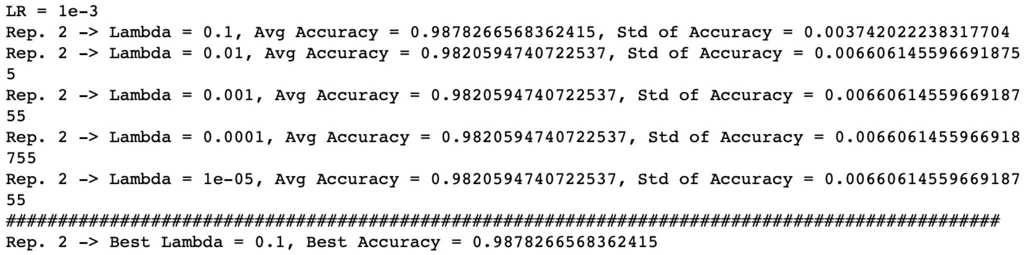
This is performed in cell 15.

## 3.8 Cross Validation for Representation 2[16][17]

Again, we've used our helper function $merger()$ to organize the representation 2 training data set into 5 different sets which doesn't consist the corresponding sub list according to their index value. We've also organized the corresponding truth values to match the new sets.

This is performed in cell 16.

And then, we've re-initiated our cross validation process, which consists of model training and extracting accuracy for each fold. And then, we've printed out the results to see which lambda value fits more perfectly than the rest.



```
LR = 1e-3
Rep. 2 -> Lambda = 0.1, Avg Accuracy = 0.9878266568362415, Std of Accuracy = 0.003742022238317704
Rep. 2 -> Lambda = 0.01, Avg Accuracy = 0.9820594740722537, Std of Accuracy = 0.006606145596691875
5
Rep. 2 -> Lambda = 0.001, Avg Accuracy = 0.9820594740722537, Std of Accuracy = 0.00660614559669187
55
Rep. 2 -> Lambda = 0.0001, Avg Accuracy = 0.9820594740722537, Std of Accuracy = 0.0066061455966918
755
Rep. 2 -> Lambda = 1e-05, Avg Accuracy = 0.9820594740722537, Std of Accuracy = 0.00660614559669187
55
################################################################################################
Rep. 2 -> Best Lambda = 0.1, Best Accuracy = 0.9878266568362415
```

This is performed in cell 17.

We've seen the fact that representation 2 provides more accuracy than the given set of features by our instructor. With that knowledge, this model looks promising to identify digit 1 and digit 5. Yet, it will be known when evaluation chapter is progressed.

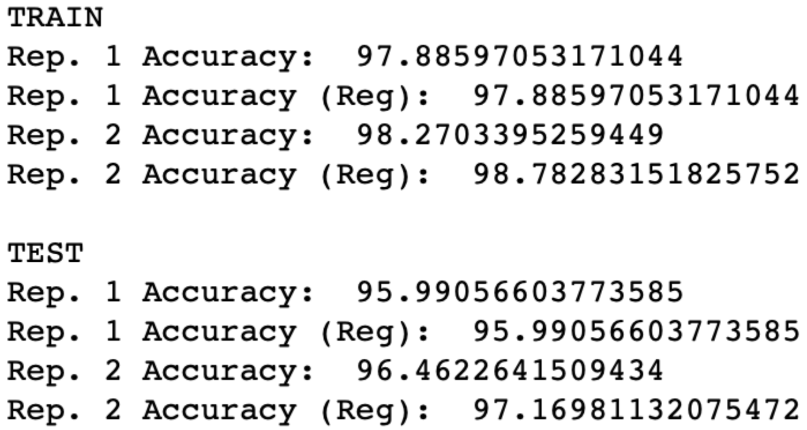
# Chapter 4

# Evaluation

## 4.1 Evaluation for Rep. 1 and Rep. 2 with the best parameters[18]

In this section, we've trained our model with the best coefficients extracted from the training part. And then, we've found out the results which are quite promising.

By our endeavoring efforts, for each representation and implementation, with and without regularization, the results are quite lovely.



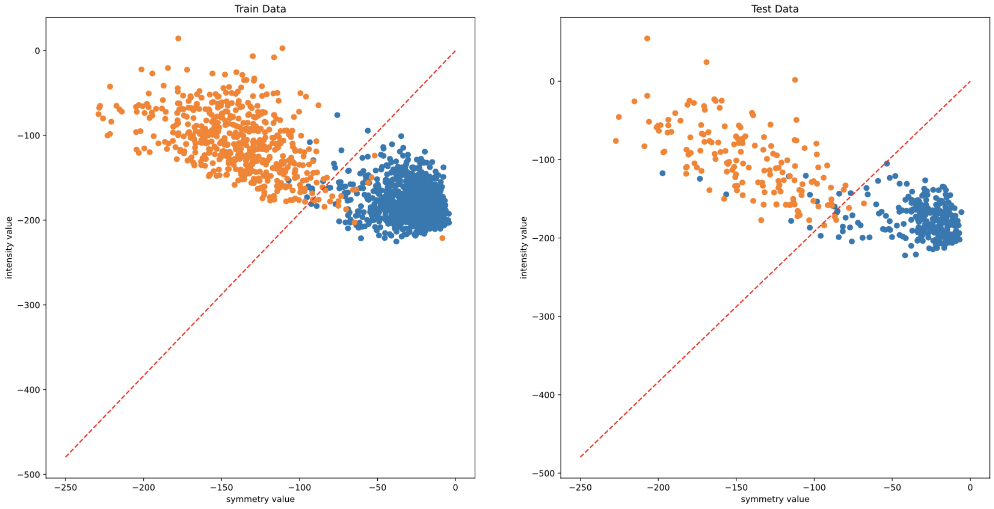
```
TRAIN
Rep. 1 Accuracy:  97.88597053171044
Rep. 1 Accuracy (Reg):  97.88597053171044
Rep. 2 Accuracy:  98.2703395259449
Rep. 2 Accuracy (Reg):  98.78283151825752

TEST
Rep. 1 Accuracy:  95.99056603773585
Rep. 1 Accuracy (Reg):  95.99056603773585
Rep. 2 Accuracy:  96.4622641509434
Rep. 2 Accuracy (Reg):  97.16981132075472
```

This is performed in cell 18.

## 4.2 Evaluation Visualization[19]

After evaluation, we've illustrated the results in graphs to see how our function works as a identifier.



In this visualization, we've plotted the dashed line via our coefficients in our $w$ coefficient array, after our model trained. And it is quite endeavoring to see that the line actually classifies the input data into two distinct groups, which are digit 1 and digit 5.

The dashed line is $l = x \cdot (-w[1]/w[2]) + (-w[0]/w[2])$, as written in our code.

$w[0]$ is constant c, $w[1]$ is constant b, and $w[2]$ is constant a. Hence the line is constructed from line formula $a \cdot x + b \cdot y + c = 0$.

This is performed in cell 19.

## 4.3 Questions / Answers

- **Question 1:** Did regularization improve the generalization performance (did it help reducing the gap between training and test accuracy/errors)? Did you observe any difference between using Representation 1 and 2?
- **Answer 1:** For Representation 1, the regularization does not change any accuracy neither for training nor for test. But Representation 2, the regularization makes a slight improvement that would results in a better output, shown in two previous section.
- **Question 2:** Which feature set did give the best results? Which one is more discriminating?
- **Answer 2:** Representation 2 has given best results, among Representation 1 and representation 2. Hence, Representation 2 is more discriminating.
- **Question 3:** What would be your next step to improve test accuracy?
- **Answer 3:** We thought merging Representation 1 with Representation 2 to create a hybrid version of set of features would result in a better accuracy. Also, for 2-D images, there could be more than 2 feature per set to distinguish data more accurately.