Bogazici University

CMPE 462 - Machine Learning

Project I Report

TEAM **F2E++**

Enes Turan Özcan - 2015400003

Emirhan Saraç - 2019700213

Muhammet Furkan Gök - 2014400039

April 24, 2020

1 Feature Extraction

1.1 Representation 1

As discussed in the class, intensity and symmetry features are extracted from both train and datasets.

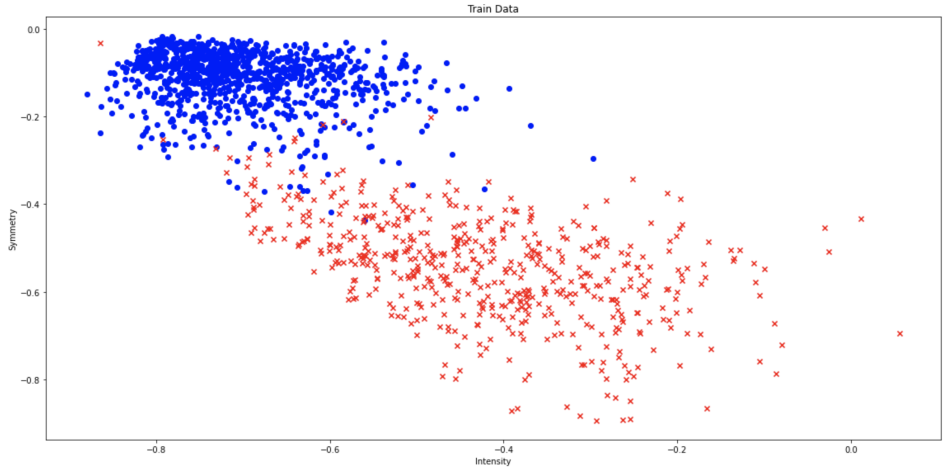

Figure 1: Train set with Representation 1

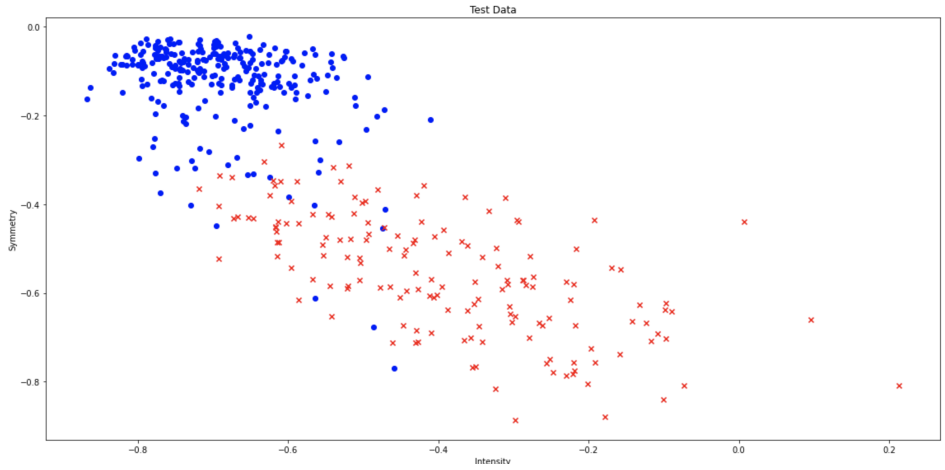

Figure 2: Test set with Representation 1

Observed on the charts, data is reasonably separated using these two

features. Blue dots correspond to 1's and red crosses stand for 5's in the dataset.

1.2 Representation 2

As the second approach to feature extraction, we used a technique based on the black pixels' column indices. We follow these steps to extract two features used in Representation 2:

- **Binarization of Grayscale Images**

  The images in the dataset are all grayscale images. First we convert them to binary images using 0.25 as black pixel threshold.

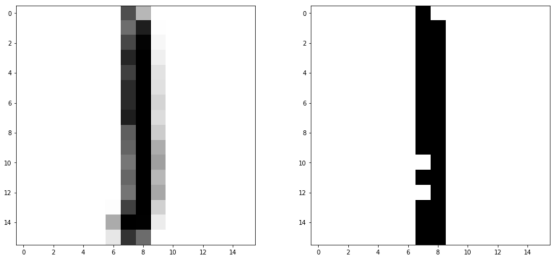

Figure 3: Grayscale to binary transform of digit one image.

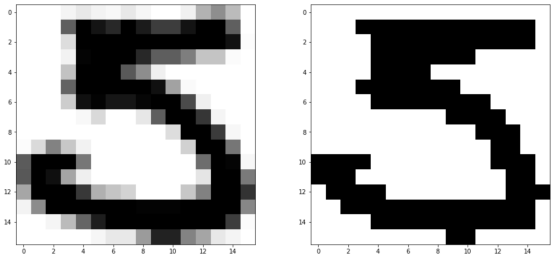

Figure 4: Grayscale to binary transform of digit five image.

- **Finding Row Centers**

  After an image is binarized, we process this image row by row such that we save all the black pixel indices. Then we find their mean and mark them. The process is simply representing each row with an associated single pixel:

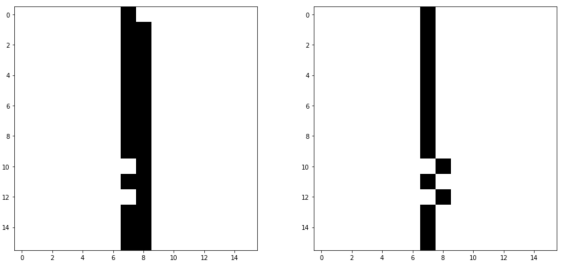

Figure 5: Representing rows with single pixels for digit one.

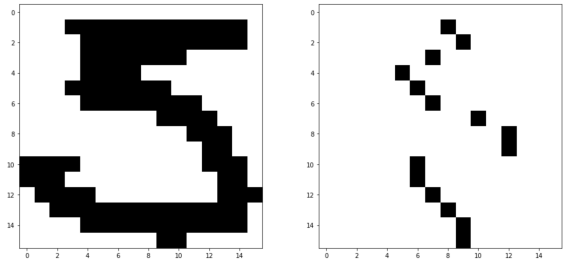

Figure 6: Representing rows with single pixels for digit five.

As seen in the images on the right, digit five has more disperse dots where digit one has dots aligned according to a horizontal line.

- **Extracting Features**

  Examining the means of these single pixels, we concluded that mean will not contribute considerable features. But standard deviation of these pixels differ considerably:

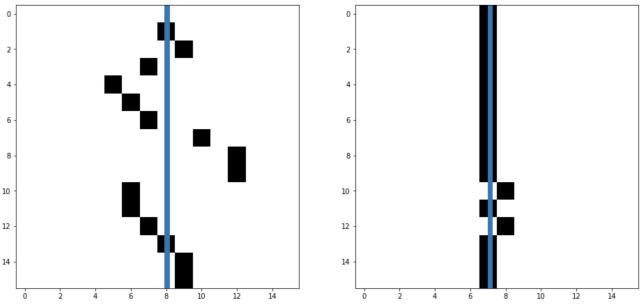

Figure 7: Averages of single pixels.

For instance, the above pixels denote:

	Standard Deviation	Standard mean
Digit 1	0.33	7.13
Digit 5	2.05	8.07

Thus we ended up using standard deviations and means of these pixels per image.**(In data, we set mid point as 0 instead of 7)**

When we drew the plots for (*stdev* x *stmean*) we got the results below which actually prove using these features definitely makes sense:

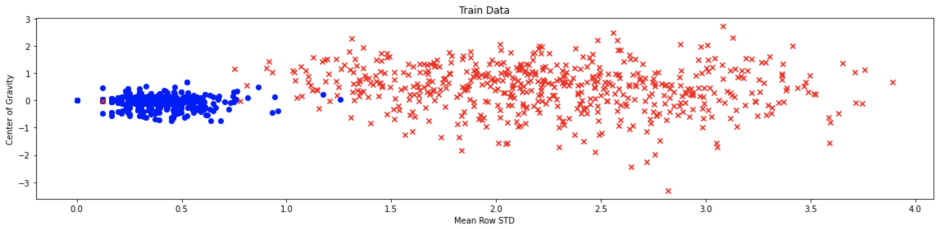

Figure 8: Train set with Representation 2

Figure 9: Test set with Representation 2

In short, the technique is that: At a particular row where black pixels are present at column x_1, x_2 and x_3, then the corresponding single pixel for this row will be $mean(x_1, x_2, x_3)$ th pixel. Although the technique for the extraction is extremely simple, we surprisingly got super successful results explained in the following sections. The technique is completely our brain child. It's not copied from a source or another project. Hence we do not provide a resource for the technique here.

2 Logistic Regression

2.1 Learning Algorithm

For this part we have implemented the learning algorithm which was included in our lecture notes (Logistic regression slides 29-30). Here is the generation of the formula that we used to calculate the gradient of the logistic loss with respect to w:
Error Function:

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

Taking the derivative of error function $E(w)$ with respect to w yields the gradient function:

Note before starting

$$\frac{\partial w^T b}{\partial w} = b$$

Derivation:
Let

$$f(x) = 1 + e^{-yw^T x}$$

and say

$$e(w) = \ln(1 + e^{-yw^T x})$$

Then

$$e'(w) = \frac{f'(x)}{f(x}$$

$$f'(x) = -yx * e^{-yw^T x}$$

Then

$$e'(w) = -yx * \frac{e^{-yw^T x}}{1 + e^{-yw^T x}}$$

Multiplying by $\frac{e^{yw^T x}}{e^{yw^T x}}$ yields

$$e'(w) = -yx * \frac{1}{1 + e^{yw^T x}}$$

Finally we ended up with

$$g_t = -\frac{1}{N}\sum_{n=1}^{N} \frac{y_n x_n}{1 + e^{(yw^T(t)x)}}$$

Figures 10 and 11 display what we get after 5 different learning rates for Representation 1 and 2 respectively, which is indeed proving that the implementation is converging.

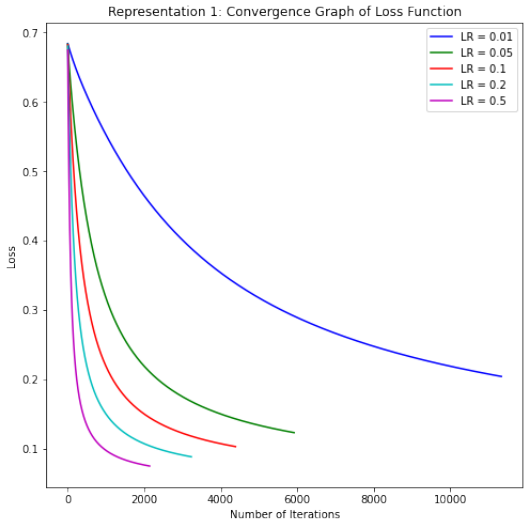

Figure 10: Convergence Graph for Representation 1

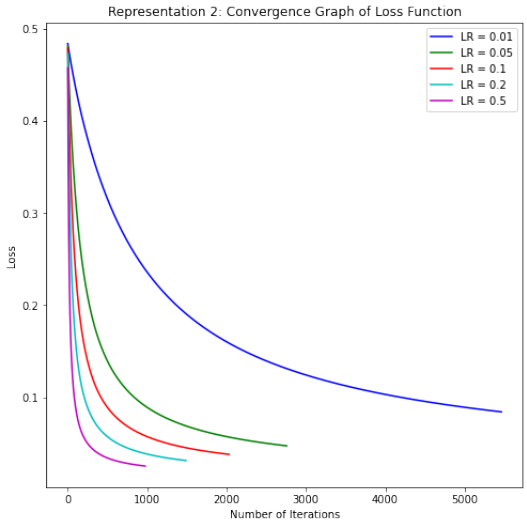

Figure 11: Convergence Graph for Representation 2

2.2 Regularization

Below 2 figures display how the algorithm does with regularized loss function for λ coefficient e^{-8}.

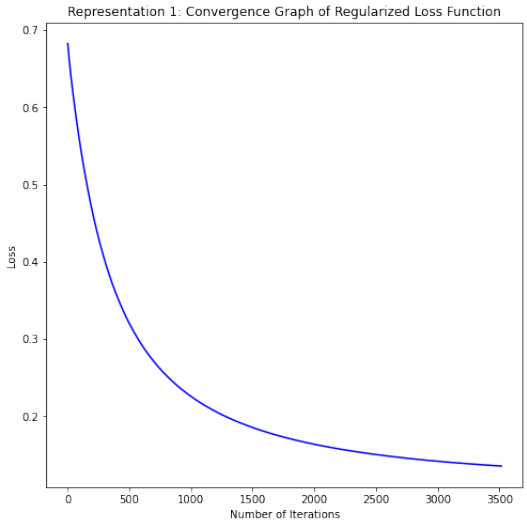

Figure 12: Convergence Graph of Regularized Loss Function

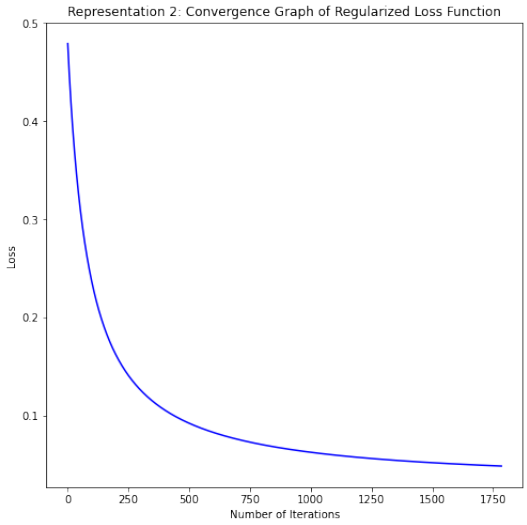

Figure 13: Convergence Graph of Regularized Loss Function

2.3 Cross Validation

In this part, we shuffle the given data with labels in case of orderly classified train data. We have used 5 different λ values.

	Mean	Standard Deviation
$\lambda = e^{-5}$	95.51	0.95
$\lambda = e^{-6}$	96.86	0.42
$\lambda = e^{-7}$	97.37	0.37
$\lambda = e^{-8}$	97.50	0.24
$\lambda = e^{-9}$	97.75	0.34

	Mean	Standard Deviation
$\lambda = e^{-5}$	98.39	2.03
$\lambda = e^{-6}$	98.97	1.28
$\lambda = e^{-7}$	99.10	1.04
$\lambda = e^{-8}$	99.23	0.92
$\lambda = e^{-9}$	99.29	0.79

3 Evaluation

If we set learning rate high numbers, train will stop earlier in general but it can overstep better local minimums or even worse can go infinite loops. As we can see from penalty term, λ, it decreases our accuracy rate. **We have not set best λ from cross validation since selected value e^{-9} is so small and does not affect accuracy. That means without penalty term, we have better accuracy. Even though that, we have set $\lambda = e^{-7}$ in order to show affect of regularization**. Also we have set initial parameter coefficients as random with mean 0 and std 0.5 so resulting accuracy may differ negligible amount.

3.1 Best Representation 1

ACCURACY(%)	Without λ	With λ
TRAIN	97.88	97.50
TEST	96.46	95.51

3.2 Best Representation 2

ACCURACY(%)	Without λ	With λ
TRAIN	99.67	99.48
TEST	98.58	98.34

4 Conclusions & Comments

After applying regularization to the learning algorithm almost no positive effect appeared neither on the graphs nor on the accuracy values. As it was mentioned in the lessons that it is not guaranteed to improve the performance by applying regularization, we decided not to further investigate on it. For both representation models, regularization has negative impact on accuracy rate.

Both Representations 1 and 2 have high accuracy 96.4 and 98.5 respectively. However, the fact that the distinction in the loss plots made us think Representation 2 is more discriminative then Representation 1. After the experiments on test data, we have seen Representation 2 will be better feature to choose.

As we discussed in one of our meetings, finding a better feature set, increasing the dimension of feature set, or enlarging the data set could be our next step to improve test accuracy.