CMPE 462: Machine Learning

PROJECT 1

Logistic Regression

Prepared By:

Fish & Fishers

Burak Ömür – 2016400186

Hasan Ramazan Yurt – 2016400078

Ömer Faruk Deniz – 2016400003

1) Explanation about Representation 2:

In representation 2, we have two features as in representation 1.

The first feature is a symmetry derived feature. In this one, firstly, we calculate norm of the difference between the image matrix (16x16) and x-axis symmetry of the image matrix (16x16). Then, we divide it by norm of the difference between image matrix (16x16) and transpose of the image matrix (16x16). The reason to use this calculation is that "1" is generally look like same in the x-axis symmetry and very different in the transpose. Therefore, the symmetry value of 1s are small with respect to 5s.

The second feature is a mean derived feature. In this one, firstly, we take mean of each row and column separately. Then, using diagonal indexes, we check for if ceiling of row mean is equal to 0 and floor of column mean equal to 0. If so, we use these rows and columns to calculate the mean of image matrix(16x16). The reason behind this is to find how likely this image to be a "1" or degree of 1 resemblance in it.

The two feature is derived by above steps for each image matrix(16x16).

We get %99.7 train accuracy and %98.34 test accuracy with this representation.

2) Derivation of the Loss Function:

We know that maximizing the likelihood $P(y_1, \dots, y_N \mid x_1, \dots, x_N)$ leads us to maximizing $\frac{1}{N}\sum_{n=1}^{N} \ln(1 + \exp(-y_n * w^T * x_n))$, therefore we consider it as our cost function and show it as $E(w(t))$ for our w in step t.

This formula helps us to calculate the cost of corresponding w, but w is not in its optimal value usually at the beginning. Therefore, we need a technique that will lead us to w^*, namely the optimal w. For that purpose, we use gradient descent technique where we update w in the direction of the opposite of cost gradient, g_t, multiplied by the learning rate η.

In other words, we want to go in the direction v such that our cost decreases the most, namely

$\Delta E = E(w(t)) - E(w(t+1))$ is maximum. When we approximate this difference, we get as detailly explained in the slides that $v = \frac{\nabla E(w(t))}{||\nabla E(w(t))||_2}$.

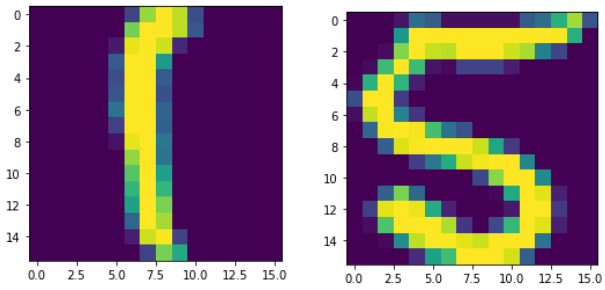
When we approach to the global minimum we want our learning rate to be smaller to not miss the global minimum in our iterations so we model it as $\eta_t = \eta * ||\Delta E(w(t))||_2$. When we put this on our update rule,

$w(t+1) = w(t) - \eta_t * v = w(t) - \eta * ||\nabla E(w(t))||_2 * \frac{\nabla E(w(t))}{||\nabla E(w(t))||_2}$ it becomes

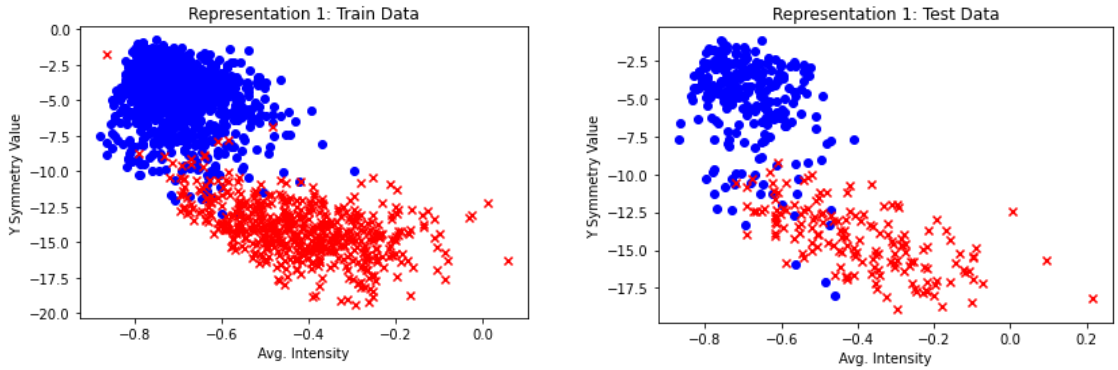
$w(t+1) = w(t) - \eta * \nabla E(w(t))$

where our gradient $g_t = \nabla E(w(t)) = \frac{1}{N}\sum_{n=1}^{N} \frac{-y_n * x_n * \exp(-y_n * w^T * x_n)}{1+\exp(-y_n * w^T * x_n)}$

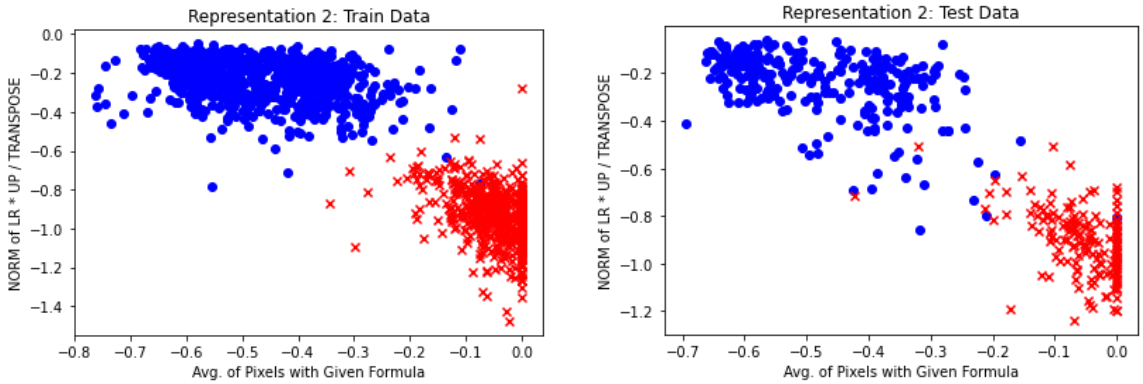
3) Display of two digit from data:



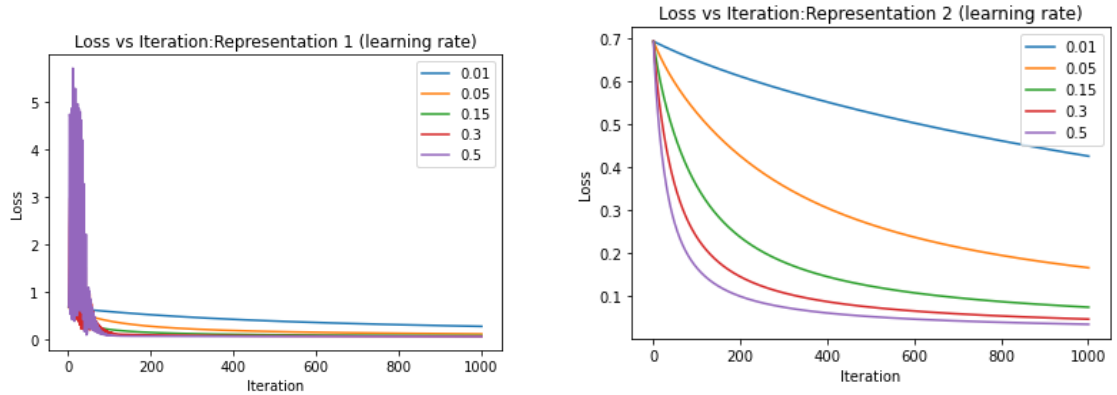
4) Graphics of test and train data of representation 1:



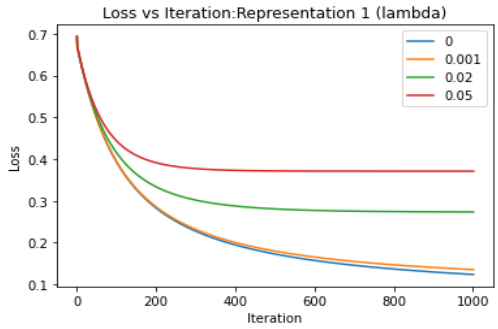
5) Graphics of test and train data of representation 2:

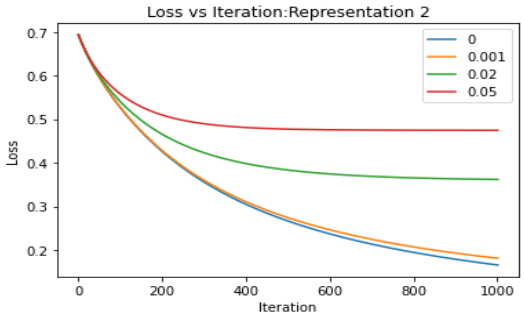


6) Graphics of loss function with respect to different step size:



7) Graphics of loss function with respect to different lambdas:





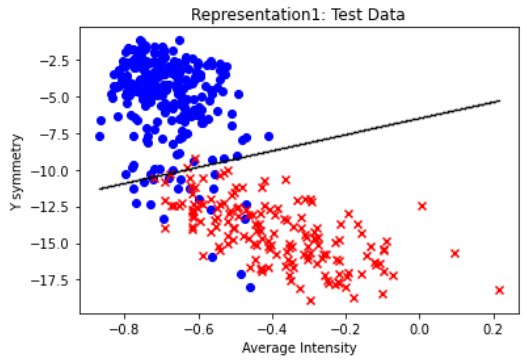
8) Cross validation accuracies according to given lambda:

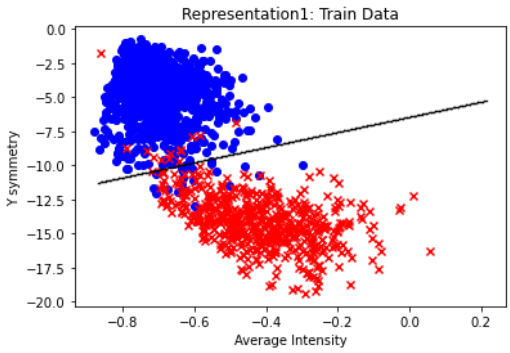
Value	Representation 1	Representation 2
0.0	0.9744 (+/- 0.0058)	0.9968 (+/- 0.0020)
0.0005	0.9750 (+/- 0.0062)	0.9968 (+/- 0.0020)
0.001	0.9750 (+/- 0.0062)	0.9968 (+/- 0.0020)
0.0015	0.9750 (+/- 0.0062)	0.9968 (+/- 0.0020)
0.002	0.9750 (+/- 0.0062)	0.9968 (+/- 0.0020)

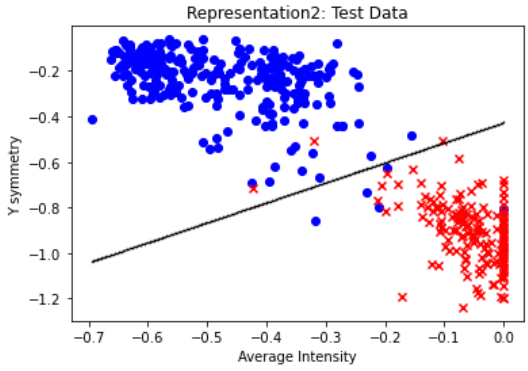
9) Classification Accuracies:

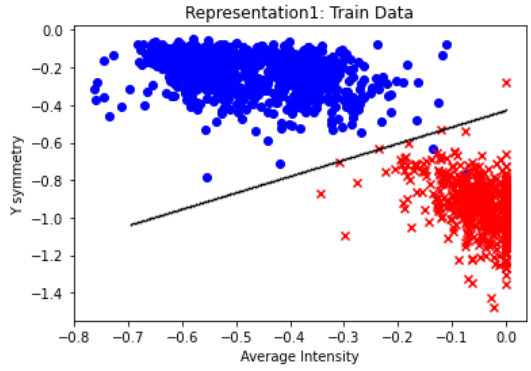
Repr	Lambda	Accuracy: Test	Accuracy: Train
R1	0.0	95.0472	97.5016
R2	0.0	98.1132	99.6156

10) Visualization of the decision boundaries with respect to representation 1 and 2:









11) Personal Comments and Thoughts:

Each of us have given same amount of contribution to this project. We studied the logistic regression and implemented together. We have done many online meetings and prepared this project.

- **Review**

Regularization, in general, did not improve accuracy. In some shuffling cases, it gave some low lambdas that might improve accuracy, but using those lambdas did not change accuracy. The reason behind this might be underfitting or not-overfitting. We stay with not-overfitting side. Also, representation 1 and 2, gave same response to the lambdas. Representation 1 and 2 did not improved by regularization.

Representation 2 is obviously better than representation 1. Because it discriminates data much more. For example: Representation 2 not only uses symmetry but also uses the difference characteristics of "1" and "5" with respect to each other.

We would add more features because, in both of representations, we could not overfit the data. Because of limited features (2), we could only draw a line between "1" and "5". However, if we have used 3-4 feature and more, we could overfit the data then using regularization improve accuracy. Another approach would be using two features again but not features selected manually. We can use 256 features that corresponds to pixels of each image. This representation is the most suitable for singularities of the images. Then we can apply a dimensionality reduction algorithm such as PCA(Principal Component Analysis) to reduce to number of features to two in which most properties are combined while correlated features are being eliminated. However, we did not implement this since manuel feature extraction was mandatory.