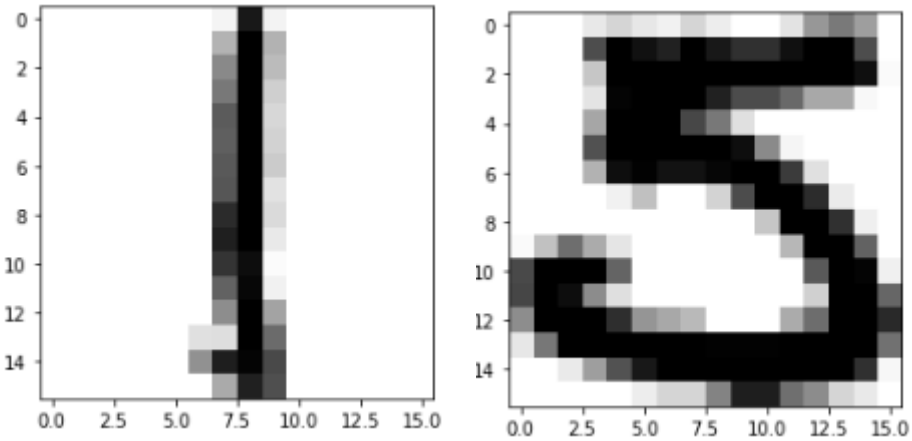
LOGISTIC REGRESSION WITH GRADIENT DESCENT REPORT

Summary

In this project, logistic regression is implemented from scratch. A subset of MNIST data that contain only the digits ‘1’ and ‘5’ is used.



The project includes feature extraction, model training, and evaluation:

1. In the feature extractions step, 4 features are extracted. 2 of the features are joint together to create the Representation 1. The other 2 features are joint to create the Representation 2.

2. In the model training step, logistic regression model is created with and without regularization. 5-fold cross validation is performed to tune the hyperparameters. Model is trained both on Representation 1 and 2.

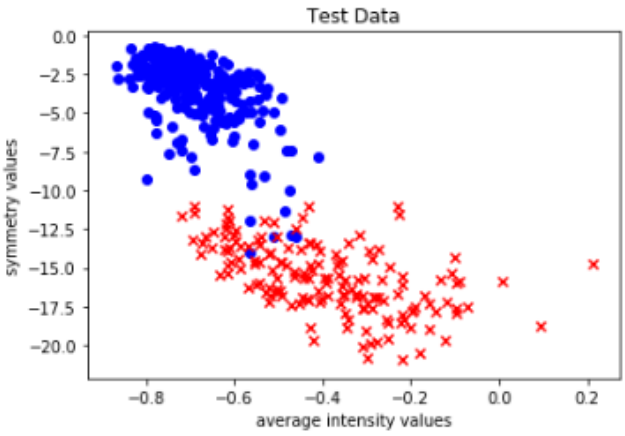
3. In the evaluation step, training and test classification accuracy are calculated. Decision boundary obtained from the logistic regression model is visualized for Representation 1.

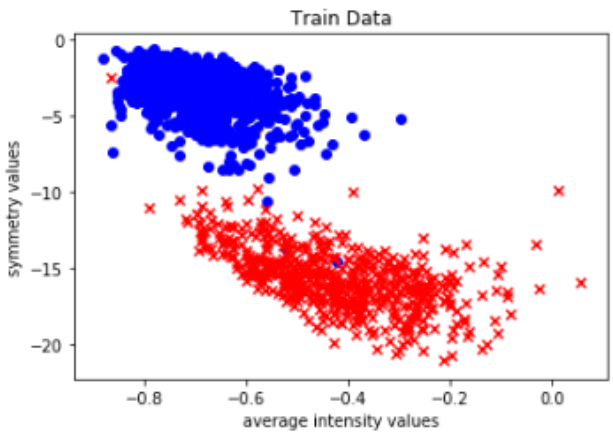
Feature Extraction:

After loading the data and displaying images, two representations are created: Representation 1 and Representation 2.

For Representation 1, ‘*average intensity*’ and *‘symmetry’* features are extracted. Average intensity is calculated by taking the average of the values belong to each data point. Symmetry is calculated by taking the y-axis symmetric of images and computing the negative of the norm of the difference between images and its symmetrical. These extractions are implemented for both the training and test datasets.

Datasets are visualized using scatter plots, where blue marker shaped o represents data points with label ‘1’ and red marker shaped x represents data point with label ‘5’.





In the scatter plots above, even though there are some outliers, data can be clearly separated according to the labels ‘1’ and ‘5’, both in the training and test sets.

For Representation 2, *‘skewness’* and ‘*standard deviation’* features are extracted. Skewness is calculated by the below formula.

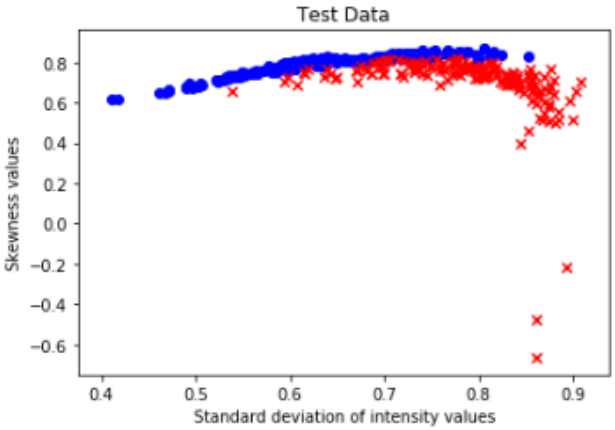
$$\gamma_i = \left(\frac{1}{N} \sum_{j=1}^{N} \left(f_{ij} - \mu_i \right)^3 \right)^{\frac{1}{3}}$$

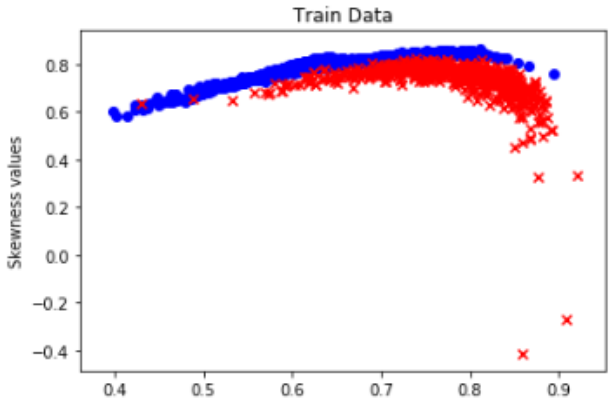
Skewness is a measure of the asymmetry of a probability distribution. It can be positive, negative or undefined. [1] Negative skew shows that the tail on the left side of the probability density function is longer than the right side, whereas positive skew shows that the tail on the right side is longer than the left side. Zero value shows that the values are evenly distributed on both sides of the mean, typically but not necessarily implying a symmetric distribution. [2] Dark glossy surfaces tend to have high skewness and light matte surfaces can have low or negative skewness. It is often useful to measure such quantities locally. [3]

Standard deviation feature is extracted by computing the standard deviation of the values belonging to each data point. It is a measure of variability used in statistics. In terms of image processing, it shows how much variation exists from the mean. A low standard deviation shows that data points tend to be very close to the mean, whereas high standard deviation shows that data points are spread out over a large range of values. [2]

These extractions are implemented for both the training and test datasets.

Datasets of Representation 2 are visualized as in Representation 1 using scatter plots, where blue marker shaped o represents data points with label ‘1’ and red marker shaped x represents data point with label ‘5’.





Although, it is harder to separate the Representation 2 data than the Representation 1, the data can be separated according to the labels '1' and '5', both in the training and test sets.

Logistic Regression:

In this section, logistic regression is implemented on both the Representation 1 and 2. Firstly, 1 is concatenated to the features as the intercept term. Then, logistic regression is applied to the training sets with the learning rate of 0.05.

Logistic regression function works according to the below learning algorithm:

1. Initialize weight vector to 0.
2. Compute the gradient. (b)
3. Record the current loss value. (a)
4. Move in the negative of the gradient direction.
5. Update the weights.
   $w(t+1) = w(t) + \eta v_t$
6. Iterate until the difference between the current loss value and the loss value of the previous step is less than 10^{-5} .
7. After iteration, return the learned weights and training loss records.

(a) Logistic Loss Function:

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

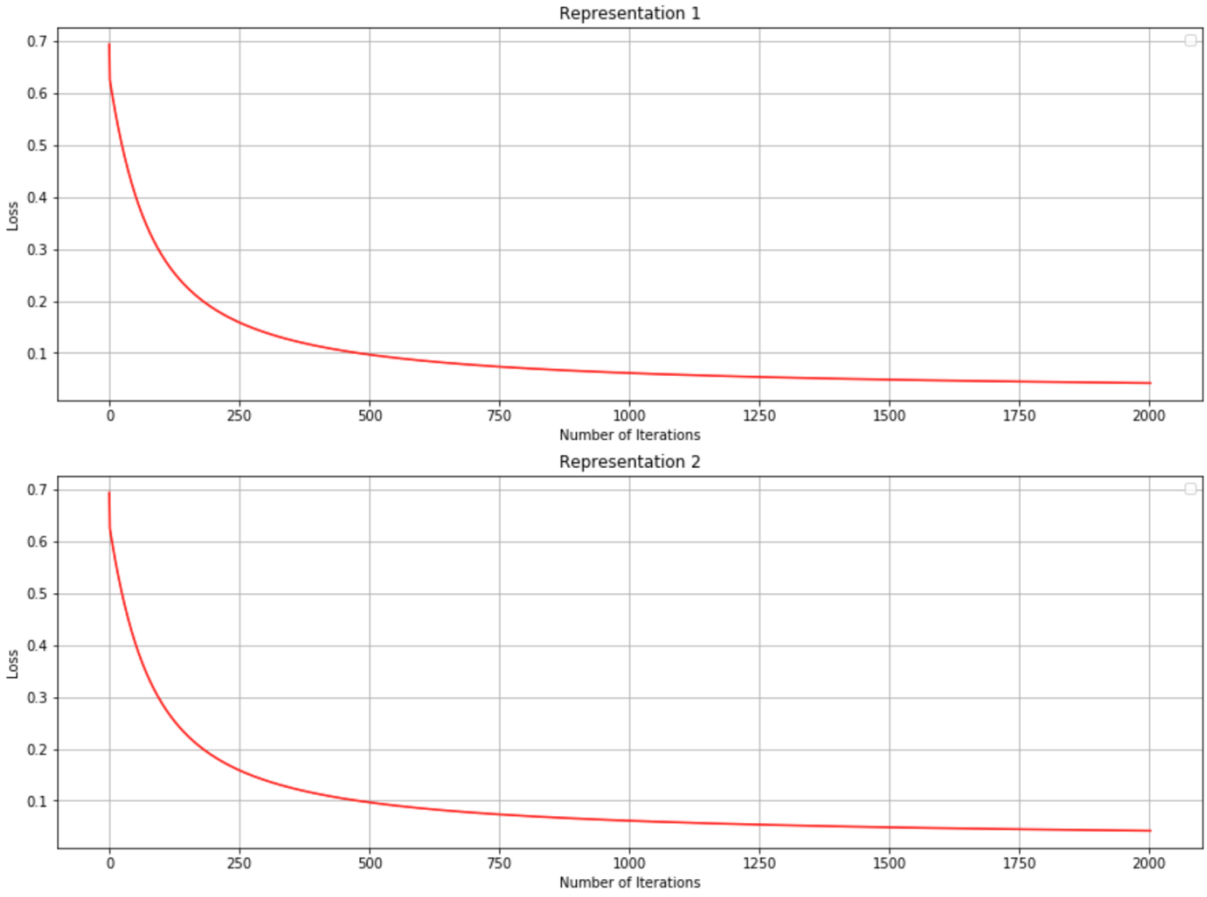
(b) Gradient Function:

$$\nabla Ein(w) = \frac{\partial E(w)}{\partial w} = \frac{1}{N}\sum_{n=1}^{N}(-y_n x_n \, e^{-y_n \cdot w^T x_n}/1 + e^{-y_n \cdot w^T x_n})$$

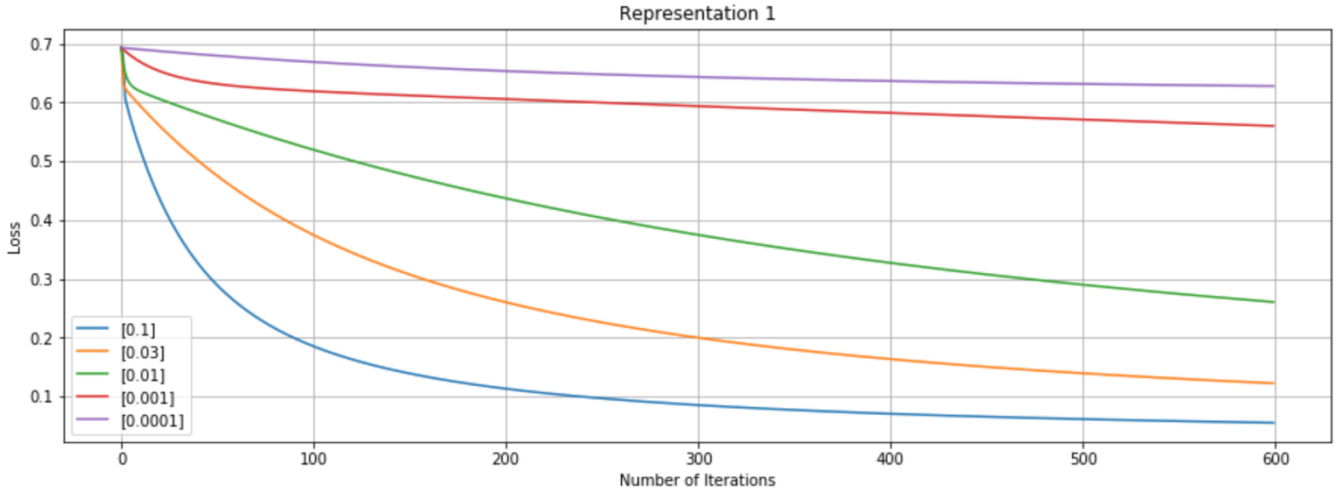
$$= \frac{1}{N}\sum_{n=1}^{N}(-y_n x_n /1 + e^{y_n \cdot w^T x_n})$$

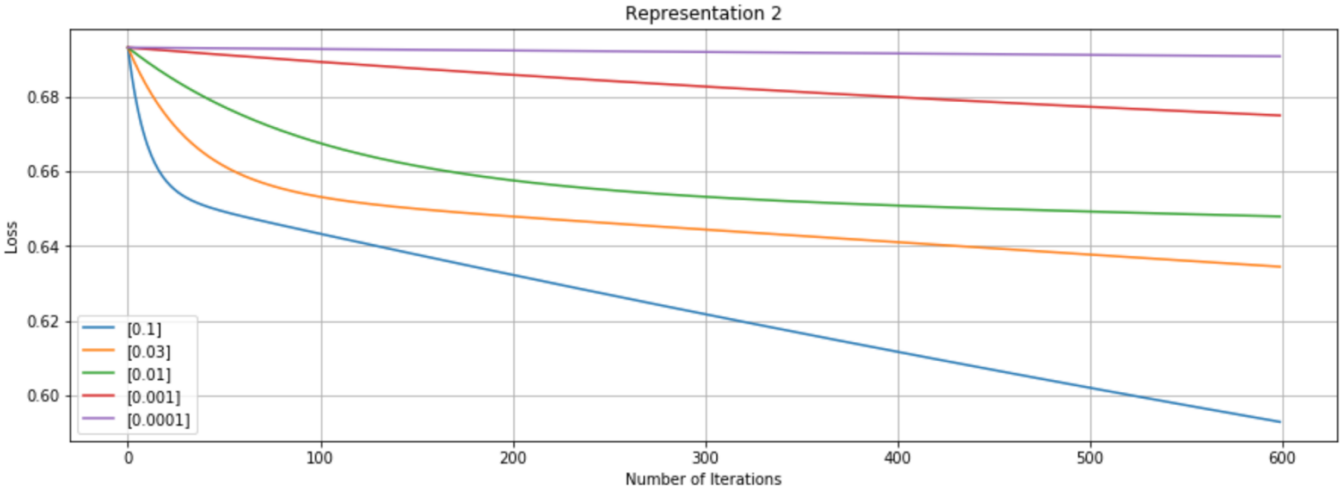
$$= \frac{-1}{N}\sum_{n=1}^{N}(y_n x_n /1 + e^{y_n \cdot w^T x_n}) = \frac{1}{N}\sum_{n=1}^{N}(-y_n x_n \, \theta)(-y_n w^T x_n)$$

Loss value at each iteration is recorded to create a convergence plot. In both representations, as number of iterations increase, loss value converges to zero as below.



Then, model is trained with 5 different learning rates and another convergence plot is made showing the effects of these learning rates.





With the learning rate 0.1, loss values converge more quickly to zero. When learning rate is higher than 0.1, we see jumps on the plot. When it is lower than 0.1, convergence takes much more time. That is why, the learning rate 0.1 is chosen to train the model.

ℓ_2 norm regularization is implemented to logistic regression model in order to control overfitting. Regularized logistic regression works according to the below learning algorithm:

1. Initialize weight vector to 0.
2. Compute the gradient. [(b)]
3. Record the current loss value. [(a)]
4. Move in the negative of the gradient direction.
5. Update the weights. [(c)]
   $w(t+1) = w(t) + \eta(v_t - \lambda w)$
6. Iterate until the difference between the current loss value and the loss value of the previous step is less than 10^{-5}.
7. After iteration, return the learned weights and training loss records.

(a) Logistic Loss Function:

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

(b) Gradient Function with Regularization:

$$E(w) = \frac{1}{N}\sum_{n=1}^{N} ln(1 + e^{-y_n \cdot w^T x_n}) + \lambda/2 \, ||w||^2_2$$

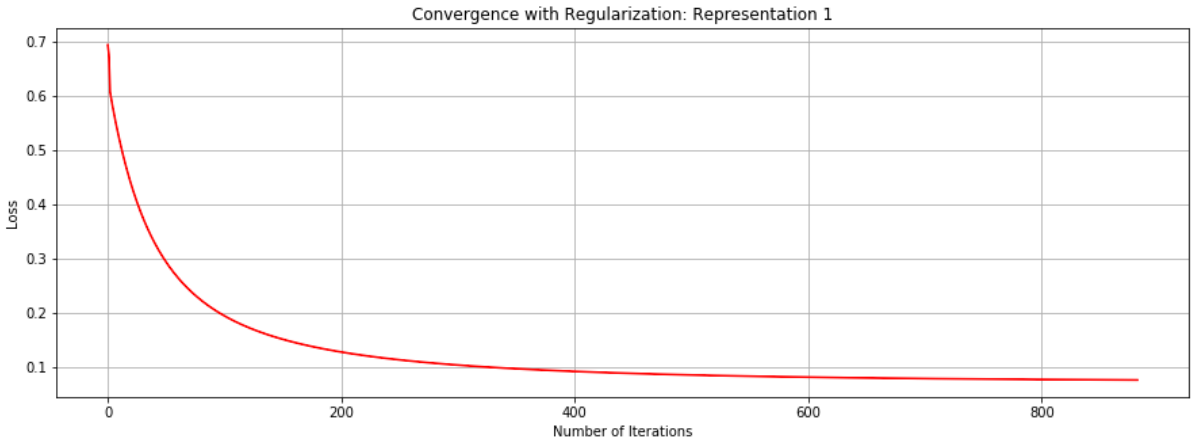
Gradient is computed by setting the derivative of the above function to zero.

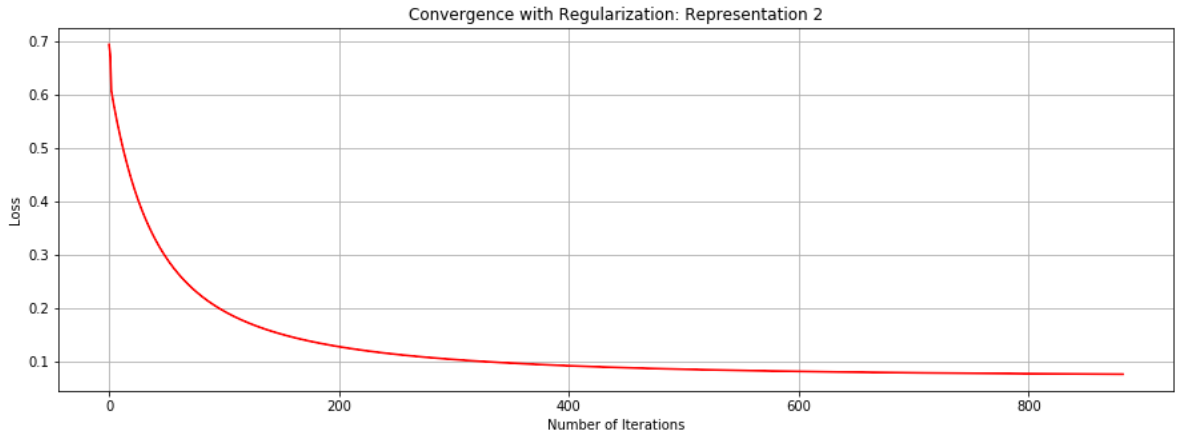
$$\nabla Ein(w) = \frac{-1}{N} \sum_{n=1}^{N} (y_n x_n / 1 + e^{y_n \cdot w^T x_n}) + \lambda w$$

(c) Since v(t) = - $\nabla Ein(w)$:

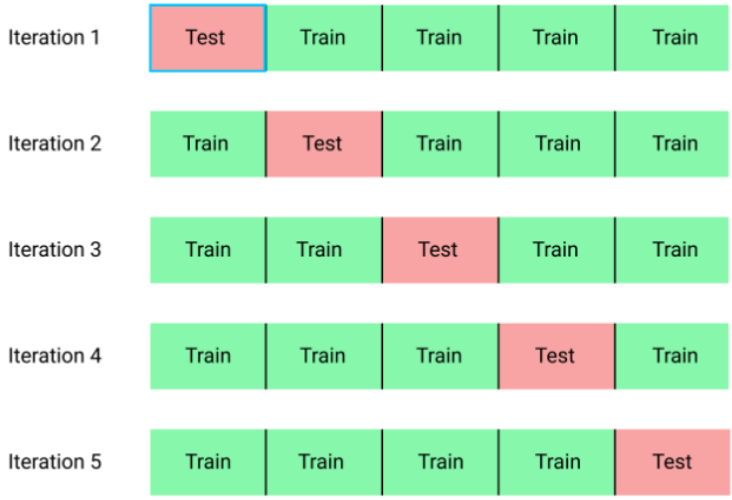
w(t+1) = w(t) + η(v_t - λw)

Loss value at each iteration is recorded to create a convergence plot. In both representations, as number of iterations increase, loss value converges to zero as below during regularized logistic regression.





5-fold cross validation is implemented to find the optimal lambda value for both Representation 1 and 2. Train datasets are split into 5 folds of equal size. 4 groups are used to train the model, the remaining 1 group is used to evaluate the model. This procedure is repeated 5 times so that each fold can become the validation data.



[4]

5-fold cross validation is implemented to training datasets with 5 different lambdas (with constant learning rate of 0.1). The effects of lambdas on the mean and standard deviation of accuracy can be seen below.

Representation 1:

	Lambda	Mean of Accuracy	SD of Accuracy
1st CV	0.05	0.9903989514213156	0.007827535517071837
2nd CV	0.09	0.9852727942983535	0.010824294580944138
3rd CV	0.1	0.9833497173752764	0.012712476821840965
4th CV	0.3	0.9205865487015646	0.06281126019497975
5th CV	0.6	0.7829073482428115	0.1751533302758587

Accuracy is the highest (0.9904) with the lambda = 0.05 for Representation 1.

Representation 2:

	Lambda	Mean of Accuracy	SD of Accuracy
1st CV	0.05	0.6435897435897436	0.4419191902114979
2nd CV	0.09	0.6435897435897436	0.4419191902114979
3rd CV	0.1	0.6435897435897436	0.4419191902114979
4th CV	0.3	0.6435897435897436	0.4419191902114979
5th CV	0.6	0.6435897435897436	0.4419191902114979

Accuracy is not affected by the change in lambda.

Task 3 – Evaluation:

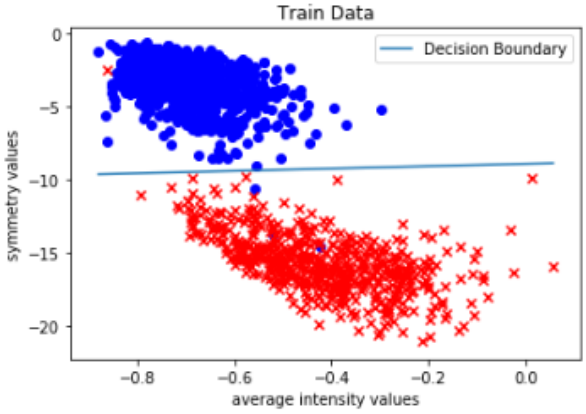
Logistic regression classifier is trained on Representation 1 and 2 with the learning rate of 0.1. Accuracy is computed by dividing the number of correctly classified samples to the total number of samples.

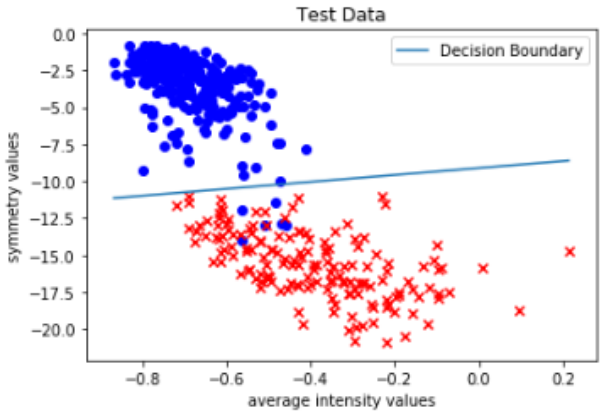
Training accuracy for Representation 1*: 0.9974375400384369*
Test accuracy for Representation 1: *0.9787735849056604*
Training accuracy with regularization for Representation 1: *0.9929532351057014*
Test accuracy with regularization for Representation 1: *0.9693396226415094*

Regularized logistic regression classifier is also trained with the learning rate 0.1 and lambda 0.05. Accuracy is computed by dividing the number of correctly classified samples to the total number of samples.

Training accuracy for Representation 2: *0.8737988468930173*
Test accuracy for Representation 2: *0.8632075471698113*
Training accuracy with regularization for Representation 2*: 0.6450992953235106*
Test accuracy with regularization for Representation 2: *0.6297169811320755*

Lastly, the decision boundary obtained from the logistic regression classifier is visualized using Representation 1. The line is given by $w^T x = 0$.





Comments:

- Using Representation 1 and 2 gives different results. Accuracy of the logistic classifier trained with Representation 1 is so much higher than the accuracy of the classifier trained with Representation 2. Also, even if regularization does not affect the accuracy of Representation 1 that much, it affects the accuracy of Representation 2 considerably.
- Regularization do not improve model performance for our data.
- Representation 1 gives the best results.
- To improve the test accuracy:
    - New data can be add.
    - Outliers and missing values (if exists) can be treated.
    - Other algorithms can be used along with logistic regression. (Dimensionality reduction, neural networks.)
    - Different values of lambdas and learning rates can be experimented.

REFERENCES

[1] Motoyoshi, Isamu & Nishida, Shin'ya & Sharan, Lavanya & Adelson, Edward. (2007). Image statistics and the perception of surface qualities. Nature. 447. 206-9. 10.1038/nature05724.

[2] V. Kumar, P. Gupta, “Importance of Statistical Measures in Digital Image Processing, International Journal of Emerging Technology and Advanced Engineering”, August 2012, vol. 2.

[3] E.H. Adelson, “Image statistics and surface perception”, MIT, MA, USA 02139

[4] R. Shaikh, https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85