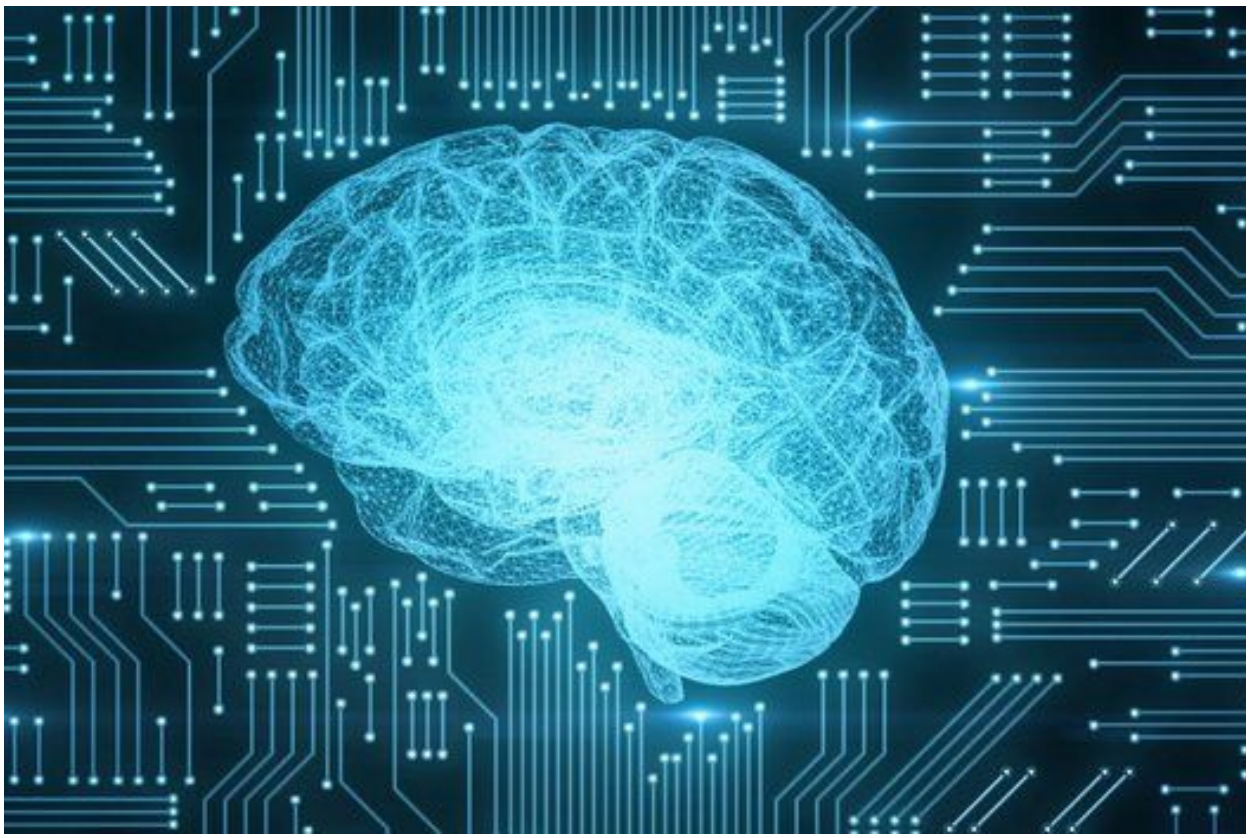

CECS 551 - Homework 1

02.26.2019

RACHAYITA GIRI
017524687
rachayita.giri@student.csulb.edu

Overview

This report is about my implementation of the Logistic Regression model for the recognition of handwritten digits (1 or 5), as a solution to Question-3 of the homework. As was asked, the main.py and solution.py files have been submitted on Beachboard separately. This report lists my observations from the experiments done with the models.

Models implemented

1. Logistic Regression
2. Third-order Logistic Regression

Data

The data consists of two different sets - training data and test data. The dimensions/shape of the data sets are as follows:

1. Training data: (1561, 257)
2. Testing data: (424, 257)

where 1561 and 424 are the number of samples in the training and test data respectively. 257 is the total number of features which contains 256 (16x16) pixel-values for the handwritten digit images. The last feature is 1 or 5 depending on what the digit is.

Since the pixel values don't make much sense because they are just numerical values, meaningful features have been extracted from them. After feature extraction and splitting the data set into features and labels, we are left with the following dimensions/shape:

1. Training data: (1561, 3)
2. Training labels: (1561,)
3. Testing data: (424, 3)
4. Testing labels: (424,)

The two extracted features are **symmetry** and **intensity** of the images, calculated from the 256 pixel values.

Third Order Logistic Regression

For the third order logistic regression model, however, the features were transformed using a third-order polynomial transformation, as follows:

Let the initial feature set be:

$$Ø_1(s, i) = (1, s, i)$$

where s = symmetry, and
i = intensity

then the transformed feature set is:

$$Ø_3(s, i) = (1, s, i, s^2, si, i^2, s^3, s^2i, si^2, i^3)$$

So the data has the following shape for third-order logistic regression:

1. Training data: (1561, 10)
2. Training labels: (1561,)
3. Testing data: (424, 10)
4. Testing labels: (424,)

Experiments and Observations

The table below compares the accuracy-based performance of the two models for 6 different combinations of *iterations* and *learning rate*.

Iteration / Learning Rate	LOGISTIC REGRESSION		THIRD ORDER REGRESSION	
	1000	**10000**	**1000**	**10000**
0.01	94.575472 %	95.754716 %	94.575472 %	95.518868 %
0.5	95.518868 %	95.990566 %	95.283019 %	95.990566 %
1.0	95.990566 %	95.990566 %	95.990566 %	95.283019 %
Top Accuracy	95.990566 %		95.990566 %	

The column charts below summarize the accuracy (in %) for 1000 and 10000 iterations at different learning rates.


Accuracy vs Learning Rate (1000 iterations)
Logistic regression
Third order regression
100
75
50
25
0
94.575472
94.575472
95.518868
95.283019
95.990566
95.990566
0.1
0.5
1


Accuracy vs Learning Rate (10000 iterations)
Logistic regression
Third order regression
100
75
50
25
0
95.754716
95.518868
95.990566
95.990566
95.990566
95.283019
0.1
0.5
1