6. (50 points) **Logistic Regression for Handwritten Digits Recognition**: Implement logistic regression for classification using gradient descent to find the best separator. The handwritten digits files are in the "data" folder: train.txt and test.txt. The starting code is in the "code" folder. In the data file, each row is a data example. The first entry is the digit label ("1" or "5"), and the next 256 are grayscale values between -1 and 1. The 256 pixels correspond to a 16 × 16 image. You are expected to implement your solution based on the given codes. The only file you need to modify is the "solution.py" file. You can test your solution by running "main.py" file. Note that code is provided to compute a two-dimensional feature (symmetry and average intensity) from each digit image; that is, each digit image is represented by a two-dimensional vector before being augmented with a "1" to form a three-dimensional vector as discussed in class. These features along with the corresponding labels should serve as inputs to your logistic regresion algorithm.

   (a) (20 points) Complete the ***logistic_regression()*** function for classifying digits number "1" and "5".

   (b) (15 points) Complete the ***accuracy()*** function for measuring the classification accuracy on your training and test data.

   (c) (10 points) Complete the ***thirdorder()*** function to transfer the features into 3rd order polynomial Z-space.

   (d) (5 points) Run "main.py" to see the classify results. As your final deliverable to a customer, would you use the linear model with or without the 3rd order polynomial transform? Briefly explain your reasoning.

   Deliverable: You should submit one report summarizes your results and the "solution.py" file to BeachBoard.

   Note: Please read the "Readme.txt" file carefully before you start this assignment. Please do NOT change anything in the "main.py" and "helper.py" files when you program.