Implementation Assignment 2

Rong Yu and Finn Womack

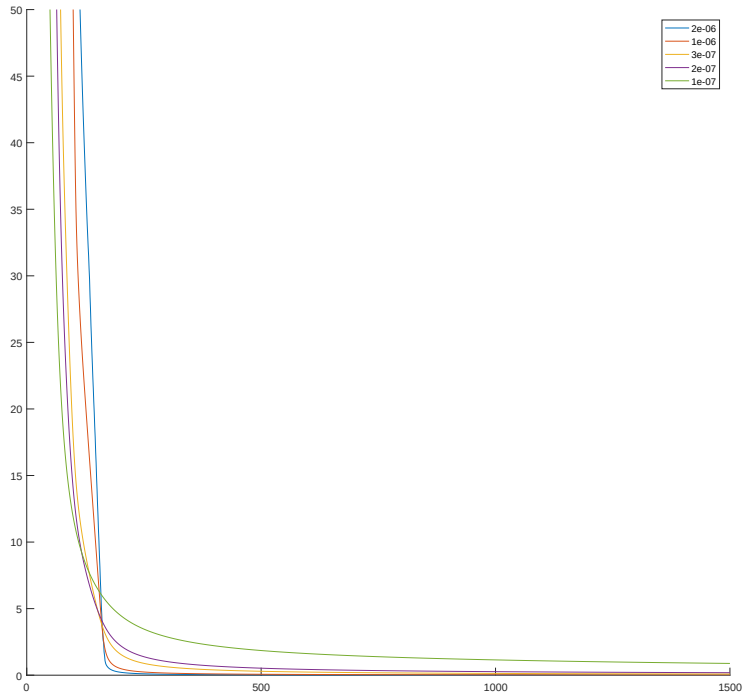
April 21, 2017

Part 1: Loading Data and Implementing Gradient Batch

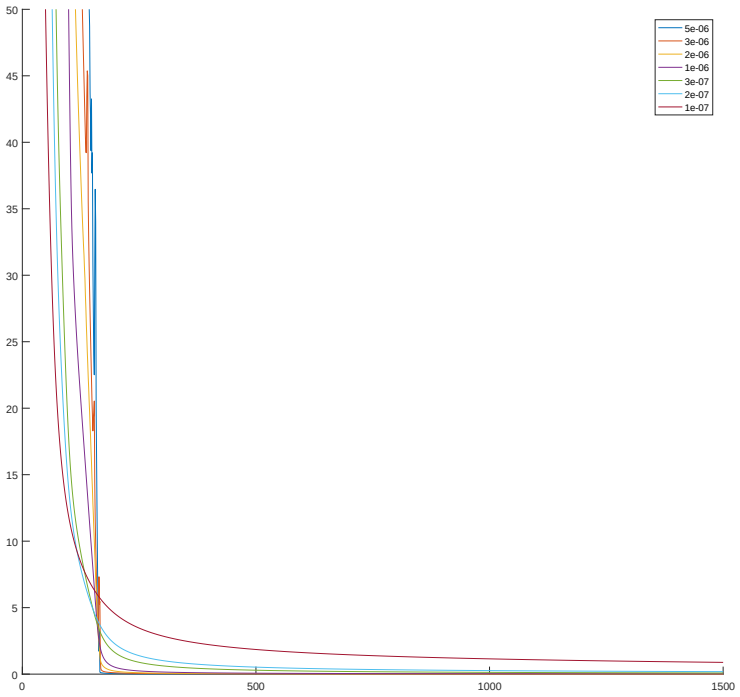
After loading the testing and training data into feature and response matrices we implemented the batch gradient decent algorithm with the following learning rates:

$$Rates = \begin{bmatrix} 2 \cdot 10^{-6} \\ 1 \cdot 10^{-6} \\ 3 \cdot 10^{-7} \\ 2 \cdot 10^{-7} \\ 1 \cdot 10^{-7} \end{bmatrix} \tag{1}$$

Then after running the algorithm on the above rates we plotted the iterations onto the loss function to get a sense of the convergence rate for the different learning rates:



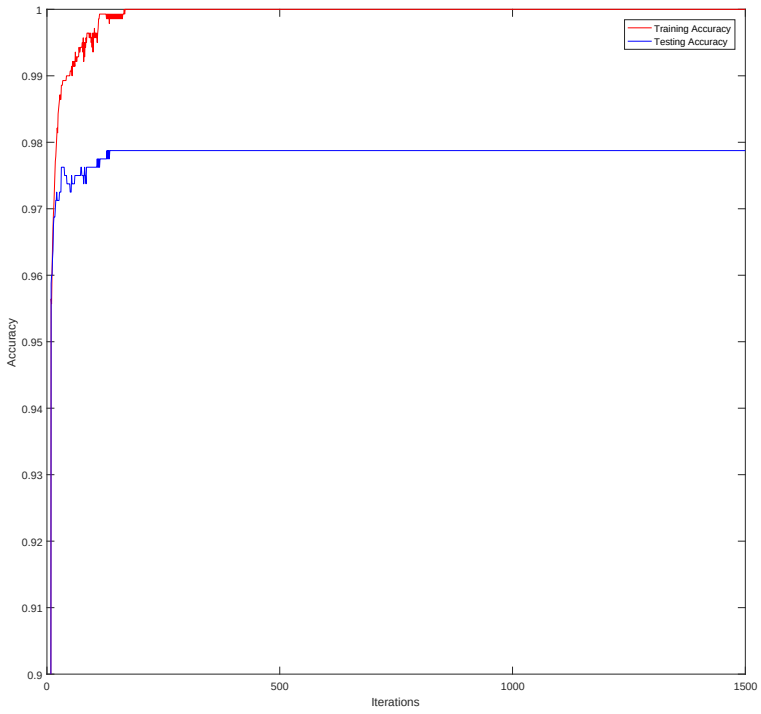
The learning rate of $2 \cdot 10^{-6}$ gives the fastest convergence. When we picked larger learning rates we started to get oscillations. If we include a couple of these rates, $3 \cdot 10^{-6}$ & $5 \cdot 10^{-6}$, the plot looks as follows:



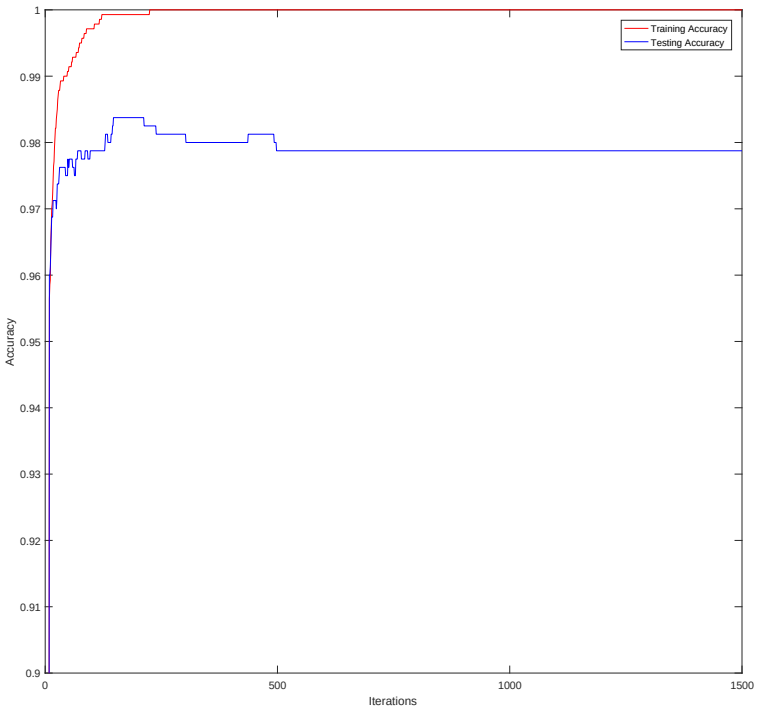
As we can see when the loss function approaches convergence at these rates we get oscillations. This makes sense because with a large learning rate the algorithm can overshoot the optimal solution

Part 2: Testing and Training Accuracies

We then selected the learning rate of $2 \cdot 10^{-6}$ to test the effects of iterations on the testing and training accuracies:



As we can see the training accuracy converges to 1 and the testing accuracy peaks at around 200 iterations and then remains constant. Something else that we thought was interesting was that if we used the learning rate of $1 \cdot 10^{-7}$ and plot the accuracies we get a better performance for the testing set at around 175 iterations and then it decreases to about the same performance as the $2 \cdot 10^{-6}$ rate:



Part 3: Deriving the Regularization Term

Consider the new objective function:

$$L(w) = \sum_{i=1}^{n} l(w^T x_i, y_i) + \frac{\lambda}{2}\|W\|_2^2$$

This is the the same as the original objective function with an added term and since the gradient of a sum of functions in the sum of the gradients all we need to do is find the gradient of $\frac{\lambda}{2}\|W\|_2^2$:

$$\nabla \frac{\lambda}{2}\|W\|_2^2 = \frac{\lambda}{2}\nabla\|W\|_2^2 \tag{2}$$

$$= \frac{\lambda}{2}\nabla \sum_{i=1}^{m} w_i^2 \tag{3}$$

$$= \frac{\lambda}{2}\begin{bmatrix} \frac{\partial}{\partial w_1}\sum_{i=1}^{m} w_i^2 \\ \frac{\partial}{\partial w_2}\sum_{i=1}^{m} w_i^2 \\ \vdots \\ \frac{\partial}{\partial w_m}\sum_{i=1}^{m} w_i^2 \end{bmatrix} \tag{4}$$

$$= \frac{\lambda}{2}\begin{bmatrix} 2w_1 \\ 2w_2 \\ \vdots \\ 2w_m \end{bmatrix} \tag{5}$$

$$= \lambda W \tag{6}$$

Thus, the new batch gradient code would be as follows:

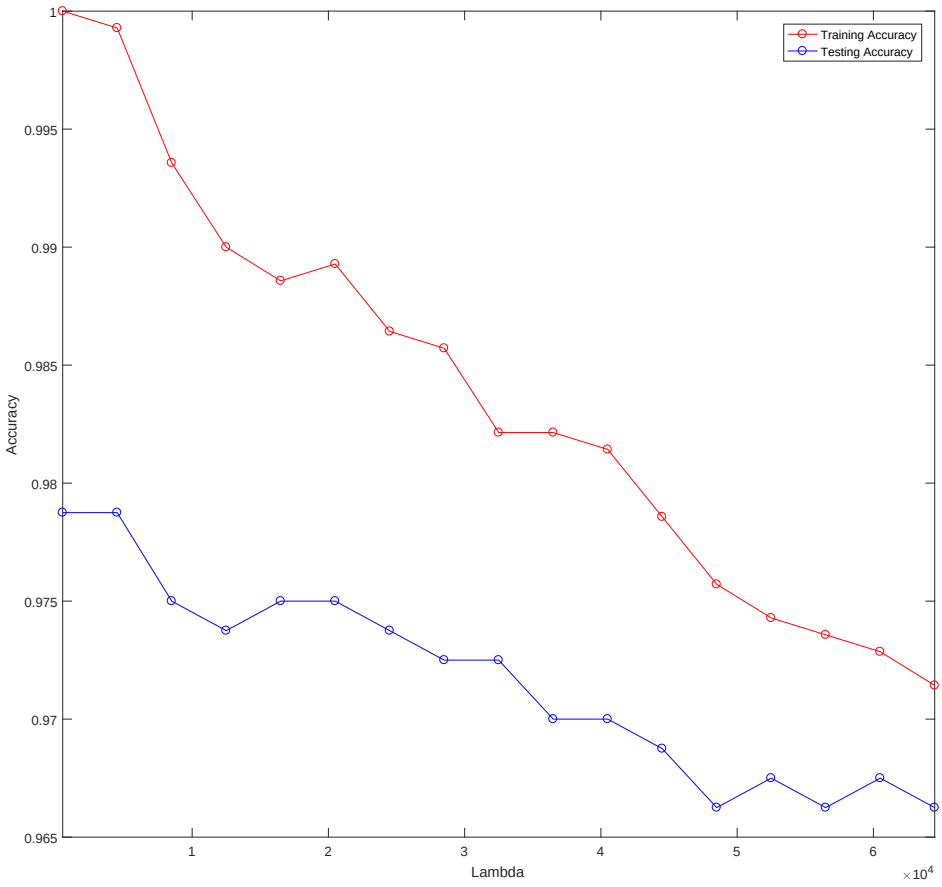
Given: training examples $(x_i, y_i), i = 1, ..., N$
$W \leftarrow [0, 0, ..., 0]$
Repeat until convergence:
    $d \leftarrow [0, 0, ..., 0]$
    For i = 1 to N do
        $\hat{y}_i \leftarrow \frac{1}{1+e^{-w \cdot x_i}}$
        $error = y_i - \hat{y}_i$
        $d = d + error \cdot x_i$
    $w \leftarrow w + \eta \cdot (d + \lambda w)$

Part 4: Implementing Regularization

To examine the effect of regularization on the algorithm we plotted the testing and training accuracies against the following choices of λ:

$$\{\lambda = (4x + .5) \cdot 10^3 | x \in \mathbb{N} \cap [0, 16]\}$$

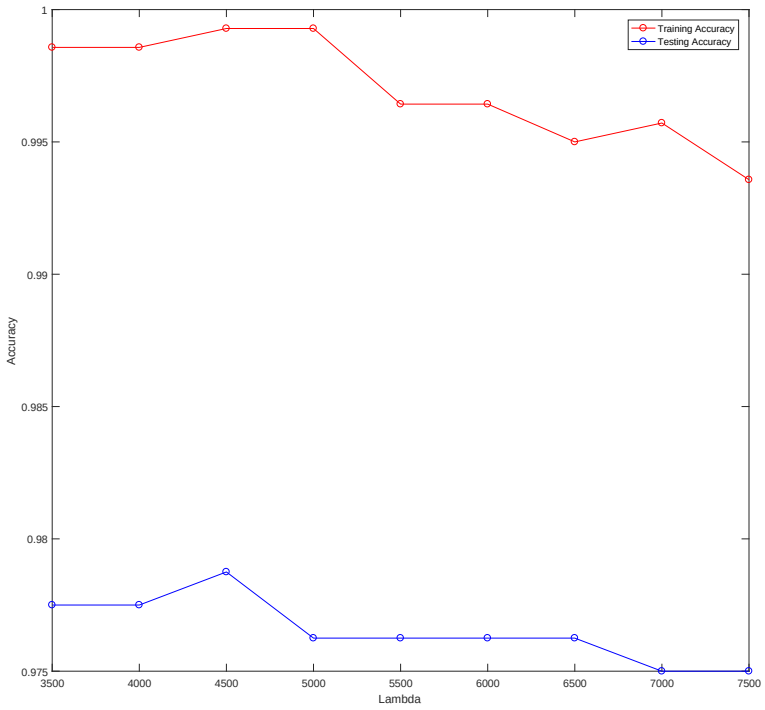
It appears that for both the testing and training accuracies it starts at the maximum accuracy and then decreases from there:



We then decided to zoom in a bit around where the function starts to decrease be examining the following lambdas:

$$\lambda = \begin{bmatrix} 3.5 \cdot 10^3 \\ 4 \cdot 10^3 \\ 4.5 \cdot 10^3 \\ 5 \cdot 10^3 \\ 5.5 \cdot 10^3 \\ 6 \cdot 10^3 \\ 6.5 \cdot 10^3 \\ 7 \cdot 10^3 \\ 7.5 \cdot 10^3 \end{bmatrix} \tag{7}$$

On closer look we can see a local maximum around 4500:



Though when examining the value at the local maximum, 4500, we found that it is the same accuracy value achieved at lower lambdas.