Bil 470 HW1 Report
Burak Han Demirbilek - 151201022

In this homework, we were asked to do a logistic regression implementation with using mini-batch gradient descent algorithm for making our trained model accurately predicts. While coding this homework, I've used python3 and not used any external libraries except numpy and matplotlib. In this report, the results will be discussed.

In big datasets, the training time is extremely slow in vanilla algorithm because of each iteration, all of the examples are being considered. So, selecting mini-batch gradient descent in big datasets is more effective and the slightly decrease in accuracy is in acceptable margin.
I've chosen the batch size to a very common value, 16. The reason of my selection is, considering the training data set sizes(166 and 281), I would like to have at least 8 batches to take advantage positive sides of the mini-batch algorithm.

In gradient descent, we can come up two solutions for when to stop iterations. We can limit the iterations if the maximum iteration number has reached. The other solution is, we can stop iterations if the difference between the current iteration and the previous iteration cost values is smaller than a value epsilon. When we are getting closer to the local minima, the difference in cost function will be getting smaller, if it is smaller than a threshold value(epsilon), we can say that we are very close to the local optima and this precision is acceptable for numerical calculations. In my code implementation, iterations will be stopped if the maximum number of iterations reached or difference in cost values are smaller than the epsilon value.
For the parameters, I chose:

- Learning rate=0.1
- Max number of iterations=500
- Batch-size=16
- Epsilon value=0.0001
- The initial beta values=0.

As you can see in Figure-1 and Figure-2, after some iterations(350+ for sonar and 300+ for ionosphere), the cost functions seems to be converged to a local minima because of the difference in costs is very small(epsilon). So, the implementation is exiting the loop for optimization purposes we talked about.
Also, the accuracy of the training data will keep improving itself. However, we can see that the maximum accuracy of test data is not following like that because of over-fitting.

References:

- https://arxiv.org/abs/1804.07612
- https://arxiv.org/abs/1609.04836
- https://www.quora.com/What-stopping-condition-for-Gradient-Descent-GD-if-we-want-to-be-epsilon-close-to-a-local-minimum-of-a-Convex-function
- https://stats.stackexchange.com/questions/33136/how-to-define-the-termination-condition-for-gradient-descent

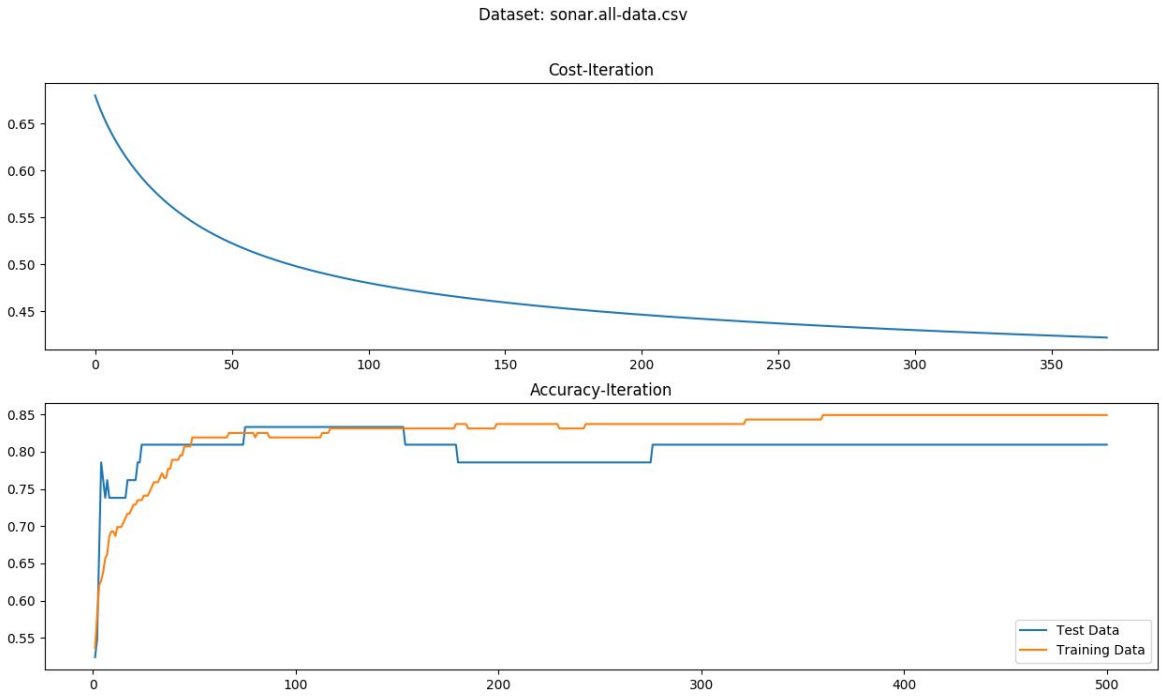

Figure 1: Dataset-1(Sonar) Cost-Iteration and Accuracy-Iteration graphs

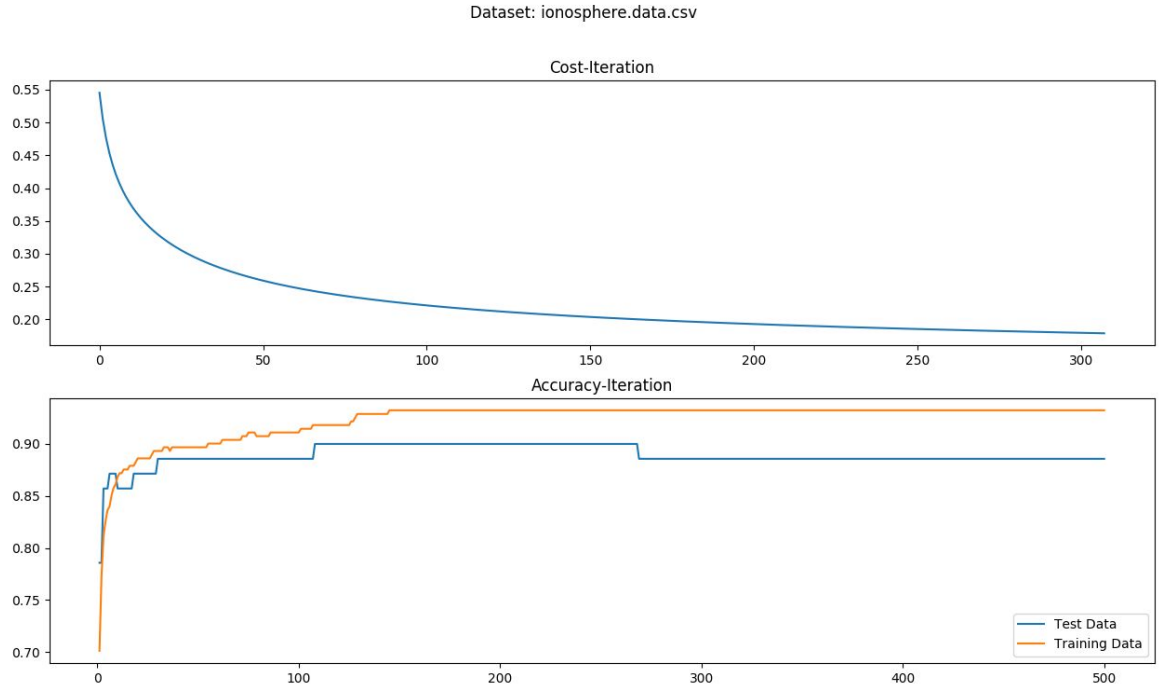

Figure 2: Dataset-1(Ionosphere) Cost-Iteration and Accuracy-Iteration graphs