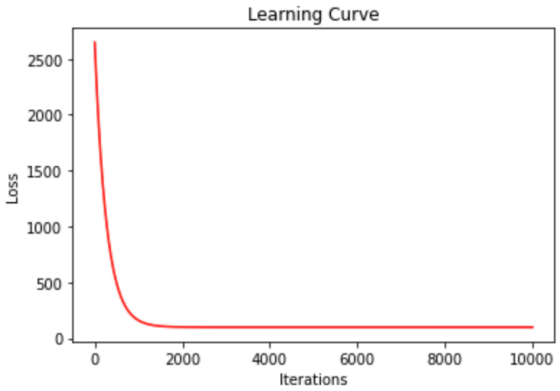
109550069 林宇柔

Machine Learning HW1

Part 1.

Linear Regression Model

1. Learning Curve



2. Mean Square Error of the prediction and ground truth = 110.4381912

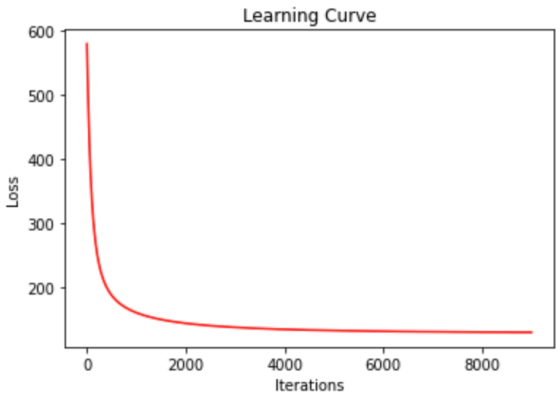
```
Mean Square Error: [110.4381912]
```

3. Weights = 52.74354002, Intercepts = -0.33375898

```
Weight: [52.74354002] Intercept: [-0.33375898]
```

Logistic Regression Model

1. Learning Curve



2. Cross Entropy Error of the prediction and ground truth = 45.646631599949025

```
Cross Entropy Error: 45.646631599949025
```

3. Weights = 4.32572715, Intercepts = 1.3772571724013523

```
Weight: [4.32572715] Intercept: 1.3772571724013523
```

Part 2.

What's the difference between Gradient Descent, Mini-Batch Gradient Descent, and Stochastic Gradient Descent?

In Gradient Descent, all training data is taken into consideration all at once, so we easily get the global optimization after enough iterations, however this only works best when the error strictly decreases so it converges directly to the minimum, and might be time-consuming for a relatively large dataset.

By contrast, Stochastic Gradient Descent is a more efficient method when the dataset is larger since it considers just one example at a time, but it never reaches a minima because the error keeps fluctuating.

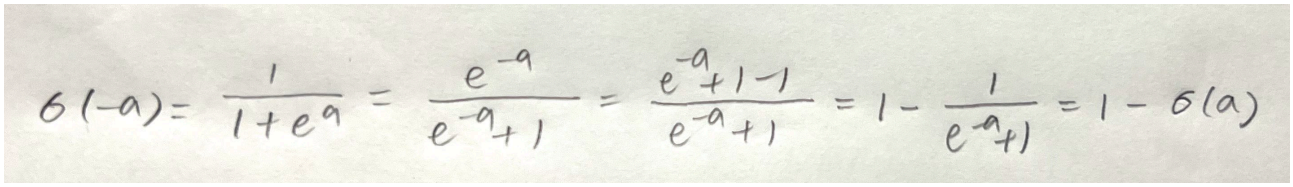
As for Mini-Batch Gradient Descent, it is a mixture of the methods mentioned above, the dataset would be split into small subsets called batches, and we separately calculate the gradients for each batch, this method solves the problem of being too slow for large datasets caused by Gradient Descent, also, the error doesn't fluctuate as obviously as Stochastic Gradient Descent does.

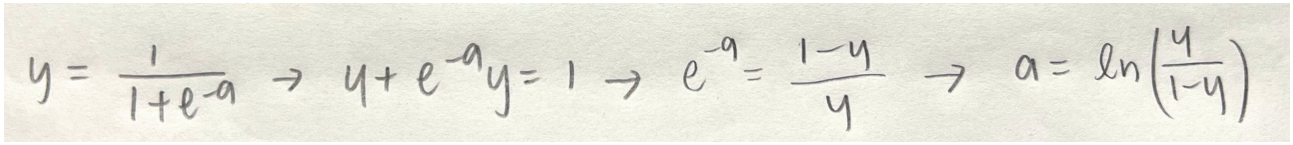
Will different values of learning rate affect the convergence of optimization?

Yes. The learning rate is a hyper-parameter that defines the extent of the speed how the weights move toward the optimal ones, so if the learning rate is too large, we might skip the optimal solution or even increase the error, but if the learning rate is too small, we might need too many iterations which is too slow to reach the best value.

Show that the logistic sigmoid function (eq. 1) satisfies the property $\sigma(-a) = 1 - \sigma(a)$ and that its inverse is given by $\sigma^{-1}(y) = \ln\{y/(1-y)\}$.

$$\sigma(a) = \frac{1}{1+\exp(-a)} \qquad \text{(eq. 1)}$$





Show that the gradients of the cross-entropy error (eq. 2) are given by (eq. 3).

$$E(\mathbf{w}_1,\ldots,\mathbf{w}_K) = -\ln p(\mathbf{T}|\mathbf{w}_1,\ldots,\mathbf{w}_K) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk}\ln y_{nk} \qquad \text{(eq. 2)}$$

$$\nabla_{\mathbf{w}_j} E(\mathbf{w}_1, \ldots, \mathbf{w}_K) = \sum_{n=1}^{N} (y_{nj} - t_{nj})\, \boldsymbol{\phi}_n \qquad \text{(eq. 3)}$$

$$\frac{\partial t_{nk} \ln y_{nk}}{\partial w_j} = \frac{\partial t_{nk} \ln y_{nk}}{\partial y_{nk}} \frac{\partial y_{nk}}{\partial a_j} \frac{\partial a_j}{\partial w_j}$$

$$= t_{nk} \cdot \frac{1}{y_{nk}} \cdot y_{nk}(I_{kj} - y_j) \cdot \phi_n = t_{nk}(I_{kj} - y_j)\phi_n$$

$$\nabla_{w_j} E(w_1, \ldots, w_K) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk}(I_{kj} - y_{nj})\phi_n$$

$$= -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} I_{kj} \phi_n + \sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} y_{nj} \phi_n$$

$$= \sum_{n=1}^{N} y_{nj}\phi_n - \sum_{n=1}^{N} t_{nj}\phi_n = \sum_{n=1}^{N} (y_{nj} - t_{nj})\phi_n \ \#$$