NYCU Introduction to Machine Learning, Homework 1

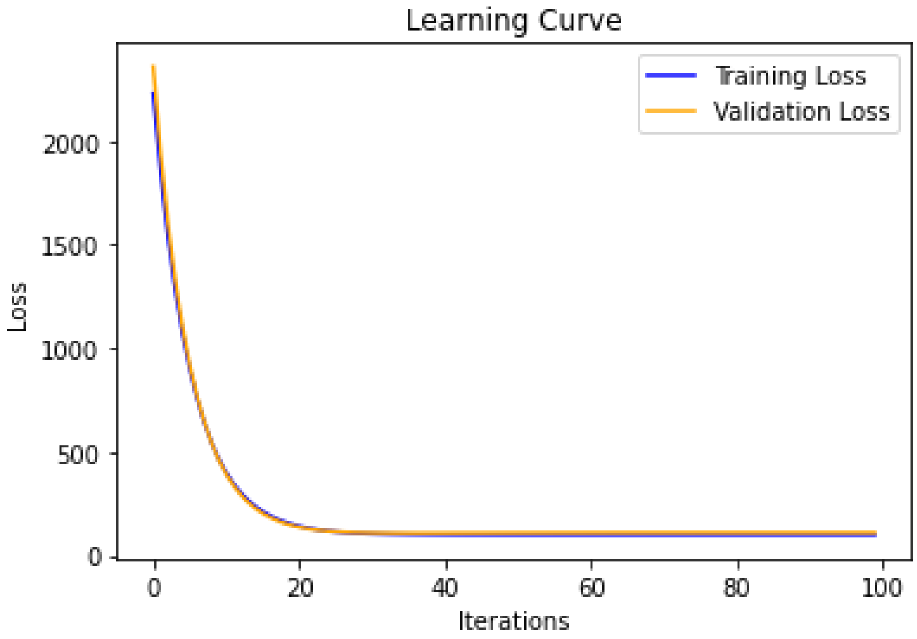
109550064 陳姵帆

Part. 1, Coding (60%):

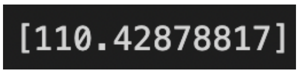
Linear regression model

Learning Rate: 0.05, Epochs: 100, Initial Weights: 0, Initial Intercept: 0

1. (10%) Learning Curve

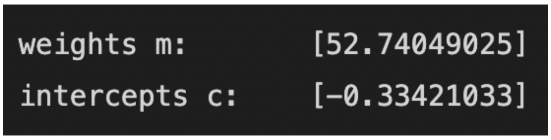


2. (10%) Mean Square Error of my prediction and ground truth



```
[110.42878817]
```

3. (10%) Weights and Intercept of my linear model

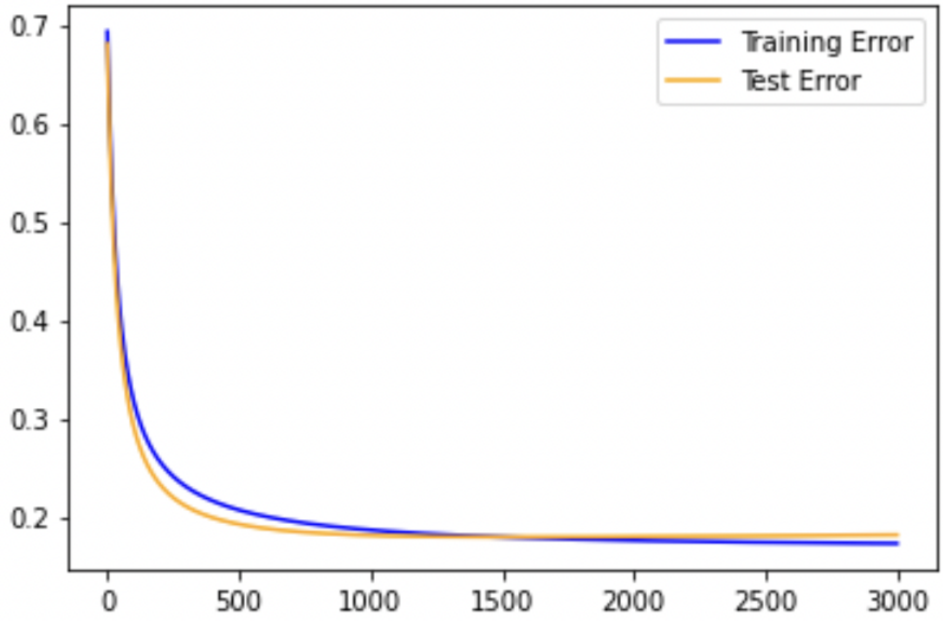


```
weights m:        [52.74049025]
intercepts c:     [-0.33421033]
```

Logistic regression model

Learning Rate: 0.05, Epochs: 3000, Initial Weights: 0, Initial Intercept: 0

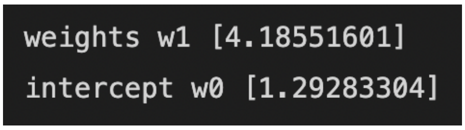
1. (10%) Learning Curve



2. (10%) Cross Entropy Error

```
0.18141152276625616
```

3. (10%) Weights and Intercept of my linear model



```
weights w1 [4.18551601]
intercept w0 [1.29283304]
```

Part. 2, Questions (40%):

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

Gradient Descent:

- train data in every epoch
    - Use all train data to calculate the gradient in every epoch.
- learning process & finded answer
    - Gradually close to the local min at that part of the cost function in every step.(The local min might also be the global min if the start point is on the right point which is in the same part with the global min on the cost function graph.)
    - If the cost function is convex, which means the function only has global min and does not have local min, gradient descent can find the best answer.
- training time
    - If the data set is large, it may take a huge amount of time to find the best answer.

Stochastic Gradient Descent:

- train data in every epoch
    - Only use 'one' randomly chosen train data to calculate the gradient in every epoch.
- learning process & finded answer
    - The cost function output value will jump up and down during the training process(not gradually decreasing compared to gradient descent), but on average the value is decreasing. As the time passes by, the cost will ultimately close to min but it is still jumping and never stops. Thus, after the algorithm is terminated, we may find a relatively good answer which is close to the min value, but might not be the best.
    - We've just mentioned the issue that gradient descent might only find the local min but not global min. With irregular movement of the answer point during the learning process, it has the probability to jump out of the local min part to the global min part of the graph. Thus, on a cost function with very irregular terrain, Stochastic gradient descent has higher probability to find the global min than the gradient descent.
- corresponding process optimization
    - We can decrease the learning rate during the learning process to try to let the answer point stay close to the best answer and prevent it from jumping a lot when close to the end of the training.
- training time
    - We can use much less time to find a good answer on a huge dataset compared to gradient descent.

Mini-Batch Gradient:

- train data in every epoch:
    - Use ‘part of’ randomly chosen train data to calculate the gradient in every epoch.
- learning process & finded answer
    - Similar to the Stochastic gradient descent, but the movement of the answer point will be more regular(=closer to the gradient descent’s learning process) as the batch size increases.
    - May also find a relatively good answer, which is close to the best answer but not exactly the best answer.
- training time
    - Use less time than gradient descent but more time than Stochastic.

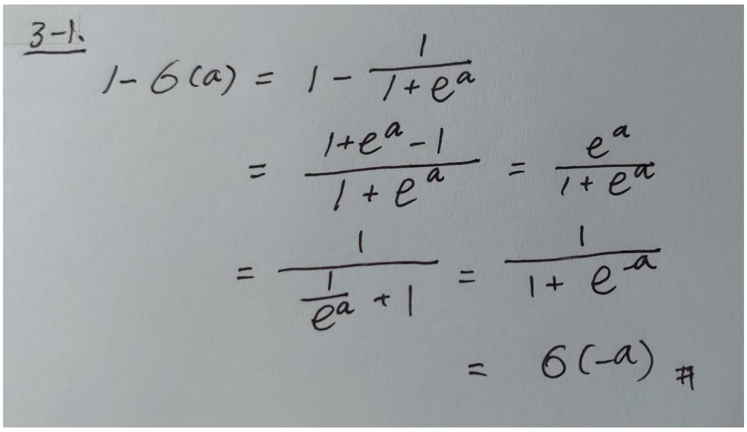
2. Will different values of learning rate affect the convergence of optimization? Please explain in detail.

- Learning rate too low
    - The algorithm needs to take many epochs calculation so that the cost value can converge to min. Will take a lot of time to train the data.

- Learning rate too high
    - If the learning rate is too high, the step to the local min valley will be too big so that it directly crosses over the valley to the other side of the mountain, and the position may be higher than the original value. In which situation the algorithm will diverge and the cost value will be higher and higher and we cannot find a good answer in the end.

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

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



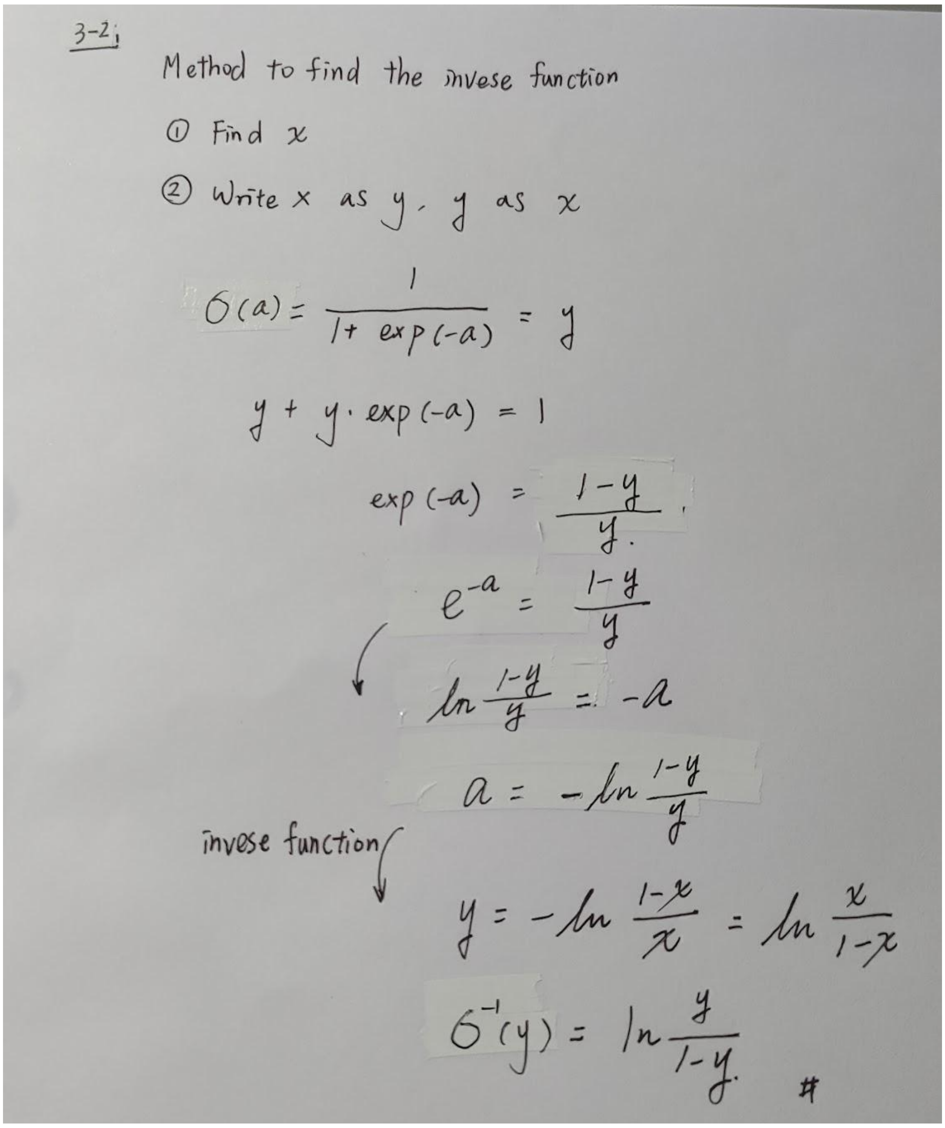
3-1.

$$1-\sigma(a) = 1 - \frac{1}{1+e^{a}}$$

$$= \frac{1+e^{a}-1}{1+e^{a}} = \frac{e^{a}}{1+e^{a}}$$

$$= \frac{1}{\frac{1}{e^{a}}+1} = \frac{1}{1+e^{-a}}$$

$$= \sigma(-a)$$ #



3-2:

Method to find the invese function

① Find x

② Write x as y, y as x

$$\sigma(a) = \frac{1}{1+\exp(-a)} = y$$

$$y + y\cdot\exp(-a) = 1$$

$$\exp(-a) = \frac{1-y}{y}.$$

$$e^{-a} = \frac{1-y}{y}$$

$$\ln\frac{1-y}{y} = -a$$

$$a = -\ln\frac{1-y}{y}$$

invese function

$$y = -\ln\frac{1-x}{x} = \ln\frac{x}{1-x}$$

$$\sigma^{-1}(y) = \ln\frac{y}{1-y}.$$ #

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

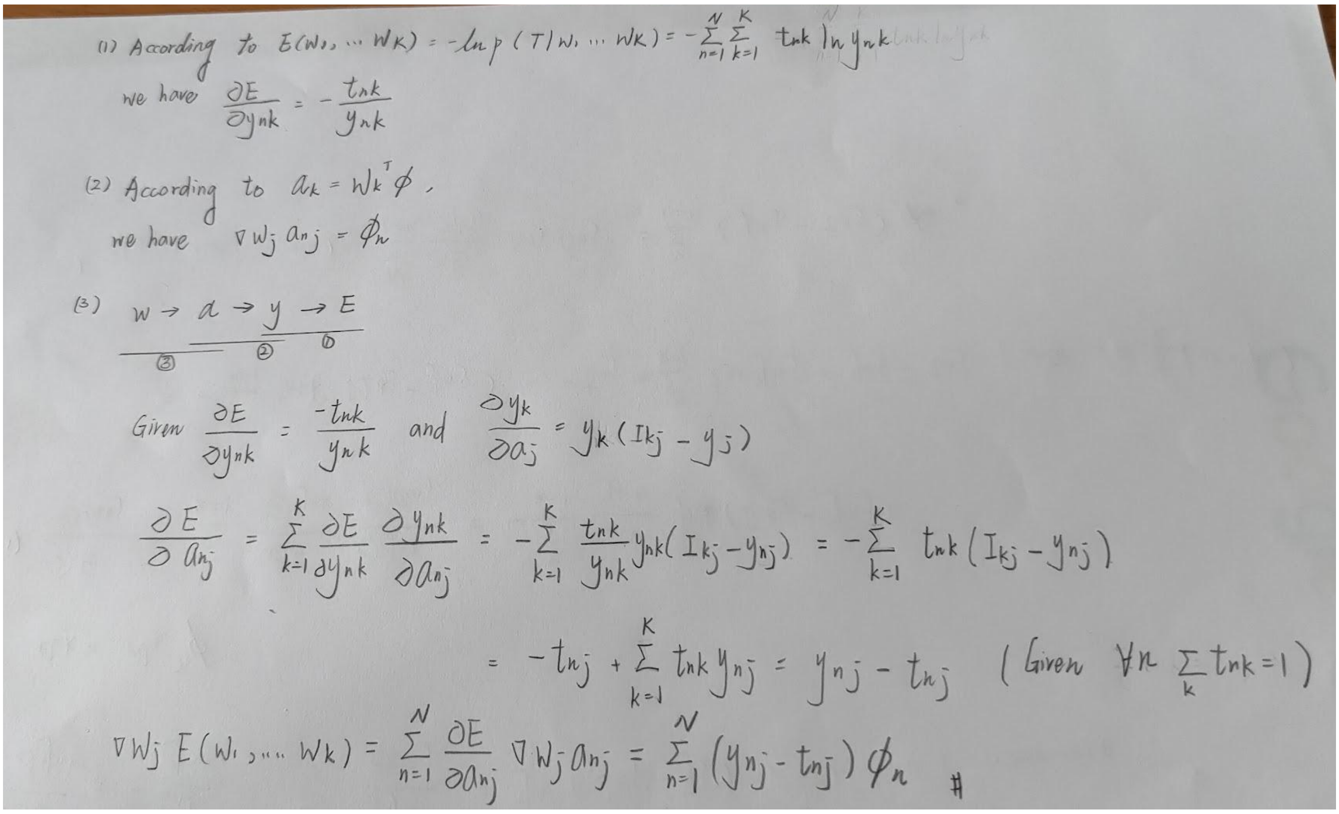
$$E(\mathbf{w}_1, \ldots, \mathbf{w}_K) = -\ln p(\mathbf{T}|\mathbf{w}_1, \ldots, \mathbf{w}_K) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_{nk}$$ (eq. 2)

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

Hints:

$$a_k = \mathbf{w}_k^{\mathrm{T}} \boldsymbol{\phi}.$$ (eq. 4)

$$\frac{\partial y_k}{\partial a_j} = y_k(I_{kj} - y_j)$$ (eq. 5)



(1) According to $E(W_1, \cdots W_K) = -\ln p\,(T/W_1 \cdots W_K) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_{nk}$

we have $\frac{\partial E}{\partial y_{nk}} = -\frac{t_{nk}}{y_{nk}}$

(2) According to $a_k = W_k^T \phi$,

we have $\nabla W_j\, a_{nj} = \phi_n$

(3) $$\underbrace{w \to a}_{\textcircled{3}} \underbrace{\to y}_{\textcircled{2}} \underbrace{\to E}_{\textcircled{1}}$$

Given $\frac{\partial E}{\partial y_{nk}} = \frac{-t_{nk}}{y_{nk}}$ and $\frac{\partial y_k}{\partial a_j} = y_k(I_{kj} - y_j)$

$$\frac{\partial E}{\partial a_{nj}} = \sum_{k=1}^{K} \frac{\partial E}{\partial y_{nk}} \frac{\partial y_{nk}}{\partial a_{nj}} = -\sum_{k=1}^{K} \frac{t_{nk}}{y_{nk}} y_{nk}(I_{kj} - y_{nj}) = -\sum_{k=1}^{K} t_{nk}(I_{kj} - y_{nj})$$

$$= -t_{nj} + \sum_{k=1}^{K} t_{nk} y_{nj} = y_{nj} - t_{nj} \quad \left(\text{Given } \forall n \ \sum_{k} t_{nk} = 1\right)$$

$$\nabla W_j\, E(W_1, \ldots W_K) = \sum_{n=1}^{N} \frac{\partial E}{\partial a_{nj}} \nabla W_j\, a_{nj} = \sum_{n=1}^{N} (y_{nj} - t_{nj})\, \phi_n \quad \#$$