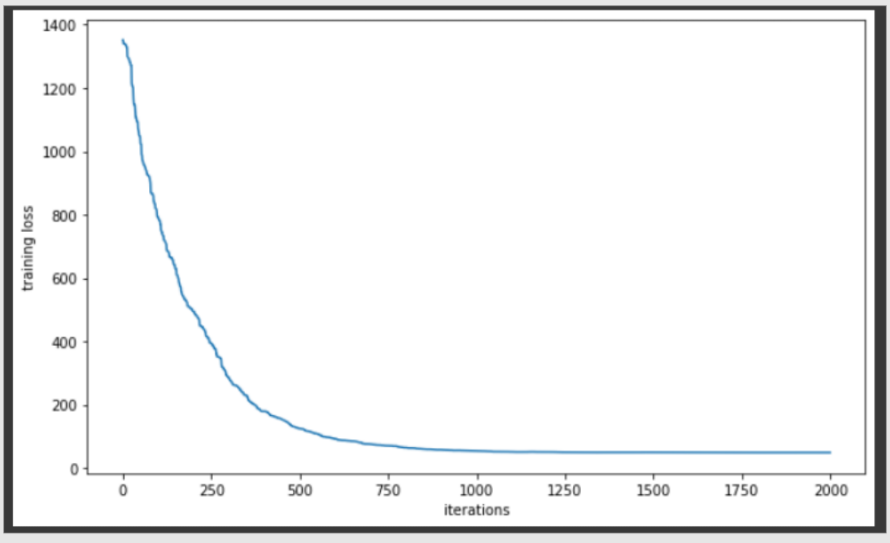
Part 1 Coding（60%）

Linear regression model :

1. Training loss over iterations



2. MSE of testing data

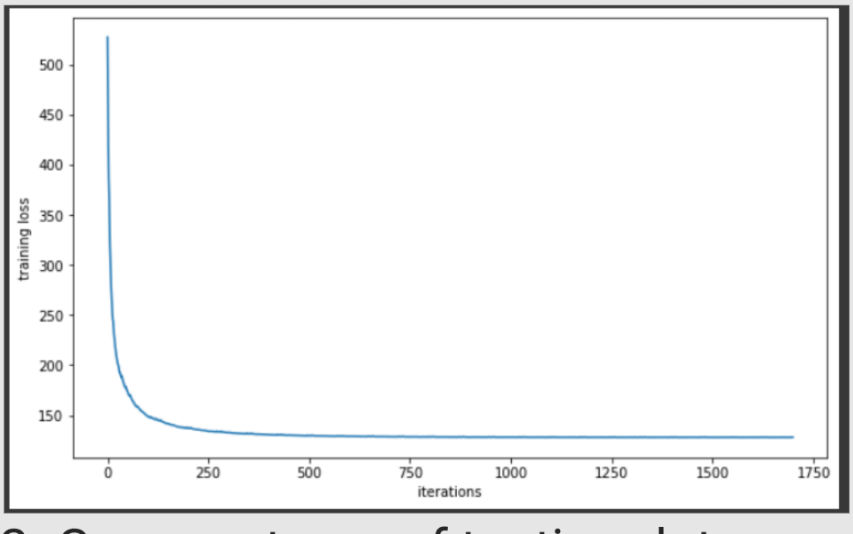
```
MSE:  110.45143802785354
```

3. weights and intercepts of the regression line

```
52.762954387738326 * x + -0.5343233718528416
```

Logistic regression model :

1. Training loss over iterations



2. Cross entropy of testing data

```
Cross Entrophy:  47.38970895952915
```

3. weights and intercepts of the regression line

```
4.776041298265617 * x + 1.7238598687659066
```

Part 2 : Short answer

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

The difference between them is "how much training data is used" for one step of updating the weight. In Gradient Descent, all of the training data is seen before we update the parameter once, which is slow to converge and may result in implementation issue because the training data may be too large to put them all in the memory. For Mini Batch GD, we split training data into small batches and each of them is used for updating weights once. This batch size is prefer to be the power of 2, which is in favor of the way data stored in a computer. Also we need to make sure the batch size is small enough to put in the

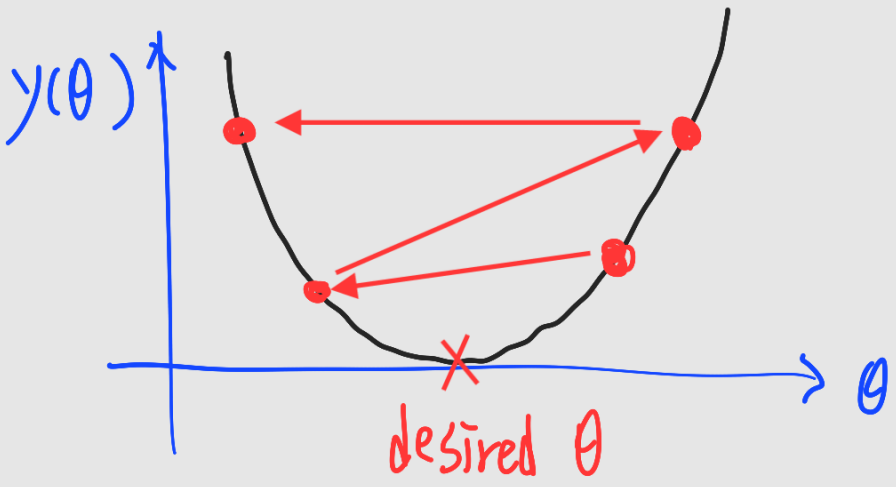
memory of CPU/GPU. SGD is the case that whenever we see a (x, y) pair, we then update the parameter immediately. The pros of this method is we can derive a not-bad solution in same amount of execution time compared to GD and Mini-batch GD, while the cons is, if we plot the loss-vs-iteration trend, we will observe a escalated curve, hot showing how the error decreases after each iteration smoothly.

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

get

Yes, it will. If the learning rate (η) is too small, we may need to spend lots of updating round to the result, which is not desired if the dataset is large. On the otherside, if η is too big, we may step across the point of local minimum and lead to divergent behavior. So it is simply a

ex:



kind of trade-off problem.

3. Show that the logistic sigmoid function (eq. 1) satisfies the property σ(−a) = 1 − σ(a) and that its inverse is given by σ−1(y) = ln {y/(1 − y)}.

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

$$=\frac{1}{\frac{e^{0}}{e^{-a}}+1}=\frac{1}{1+e^{-(-a)}}=\sigma(-a)$$

$$\because f(f^{-1}(x))=x$$

$$\therefore \sigma\left(\ln\left(\frac{y}{1-y}\right)\right)=\frac{1}{1+\frac{1-y}{y}}$$

$$=\frac{1}{1+e^{-\left(\ln\left(\frac{y}{1-y}\right)\right)}}=\frac{y}{y+1-y}=y \quad \#$$

4.

int(第n筆資料是∈第k類)

① $E = E(w_1, w_2, \dots w_K) = -\ln P(T \mid w_1, w_2, \dots w_K)$

$$= -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \cdot \ln P(C_k \mid \phi_n)$$

$$= -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \cdot \ln y_{nk}$$

↳第n筆資料∈第k類的softmax結果

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

② $$P(C_k \mid \phi_n) = y_k(\phi_n) = \frac{e(a_{nk})}{\sum_j e(a_{nj})}$$

$$\Rightarrow \frac{\partial y_{nk}}{\partial a_{nj}} = \frac{e^{a_{nk}}\sum_j e^{a_{nj}} - e^{a_{nk}} \cdot e^{a_{nk}}}{\left[\sum_j e(a_{nj})\right]^2} = y_k - y_k^2$$

(when $j=k$) $= y_{nk}(1 - y_{nk})$

$$\Rightarrow \frac{\partial y_{nk}}{\partial a_{nj}} = \frac{-e^{a_{nk}} \cdot e^{a_{nj}}}{\left[\sum_j e(a_{nj})\right]^2} = -y_{nk} \cdot y_{nj}$$

($j \neq k$)

$$\Rightarrow \frac{\partial y_{nk}}{\partial a_{nj}} = y_{nk}(I_{kj} - y_{nj}), \quad I_{kj} = \begin{cases} =1, & \text{if } k=j \\ =0, & \text{if } k \neq j \end{cases}$$

③

$$a_{nj} = w_j^T \phi_n$$

$$\Rightarrow \frac{\partial a_{nj}}{\partial w_j} = \phi_n$$

Goal

By ① + ② + ③, $E \leftarrow y \leftarrow a \leftarrow w$

④

$$\frac{\partial E}{\partial a_{nj}} = \sum_{k=1}^{K} \frac{\partial E}{\partial y_{nk}} \times \frac{\partial y_{nk}}{\partial a_{nj}}$$

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

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

$$= -t_{nj} + \sum_{k=1}^{K} t_{nk} y_{nj} = y_{nj} - t_{nj}$$

$-t_{nj}$: $j=k$ 時那項

$\sum_{k=1}^{K} t_{nk} = 1$

→ Why this summation?

$E \leftarrow \begin{matrix} y_{n1} - a_{nj} \\ y_{n2} - a_{nj} \\ \vdots \\ y_{nk} - a_{nj} \\ \vdots \\ y_{nK} - a_{nj} \end{matrix}$

By ④

$$\frac{\partial E}{\partial w_j} = \boxed{\sum_{n=1}^{N}} \frac{\partial E}{\partial a_{nj}} \times \frac{\partial a_{nj}}{\partial w_j}$$

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

Why this summation:

$$E \begin{cases} a_{1j} - w_j \\ a_{2j} - w_j \\ \vdots \\ a_{nj} - w_j \\ \vdots \\ a_{Nj} - w_j \end{cases}$$