NYCU Introduction to Machine Learning, Homework 1

Deadline: Oct. 25, 23:59

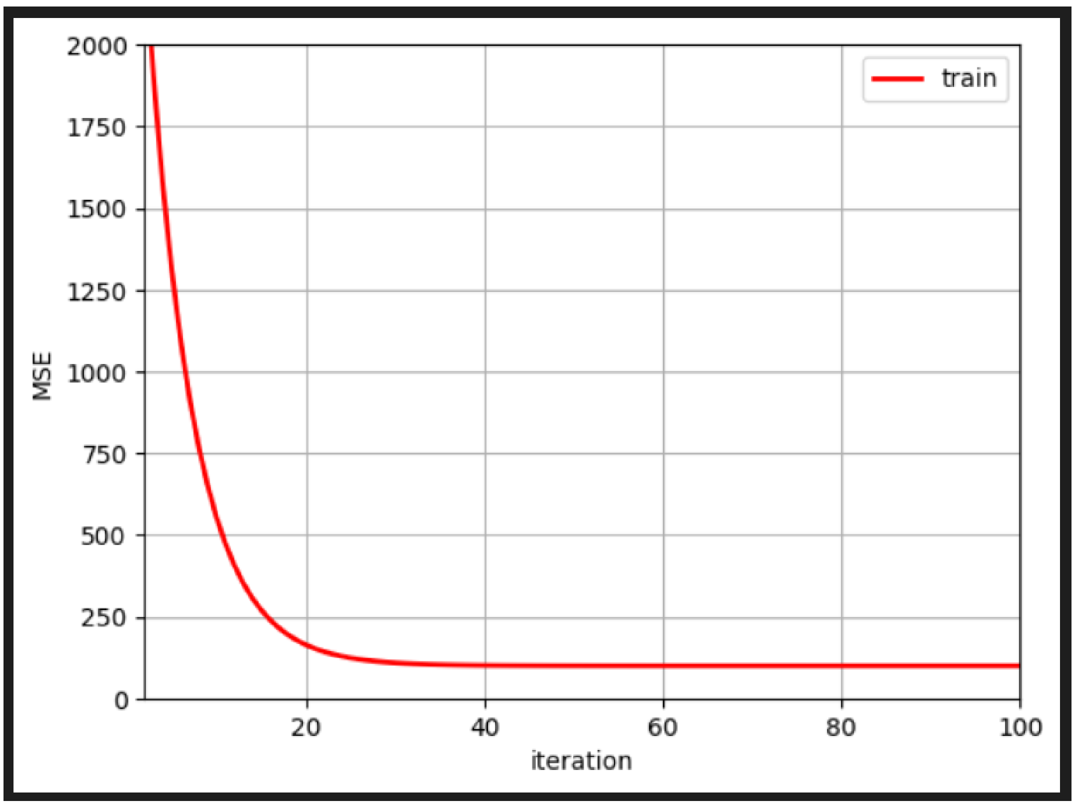
Part. 1, Coding (60%):

In this coding assignment, you need to implement logistic regression and linear regression by using only NumPy, then train your implemented model using **Gradient Descent** by the provided dataset and test the performance with testing data. Find the sample code and data on the GitHub page
https://github.com/NCTU-VRDL/CS_CS20024/tree/main/HW1

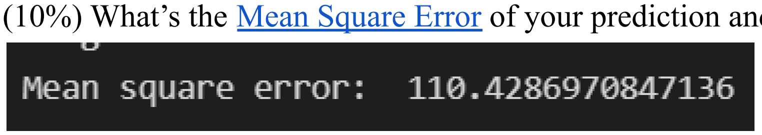
Please note that only NumPy can be used to implement your model. You will get no points by simply calling sklearn.linear_model.LinearRegression. Moreover, please train your linear model using Gradient Descent, not the closed-form solution.

Linear regression model

1. (10%) Plot the learning curve of the training, you should find that loss decreases after a few iterations and finally converge to zero (x-axis=iteration, y-axis=loss, Matplotlib or other plot tools is available to use)



2. (10%) What's the Mean Square Error of your prediction and ground truth?

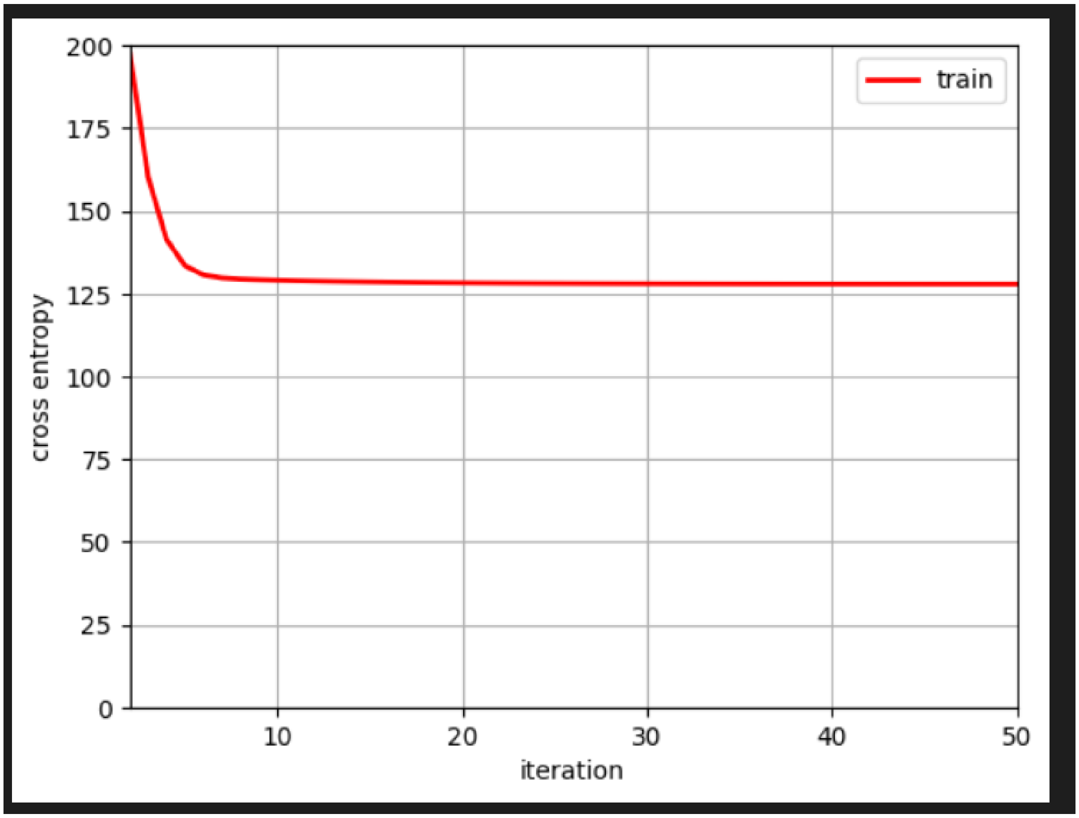


3. (10%) What’re the weights and intercepts of your linear model?

```
Intercept:  -0.33416883445261714
Weight:  52.74054395685851
```

Logistic regression model

1. (10%) Plot the learning curve of the training, you should find that loss decreases after a few iterations and finally converge to zero (x-axis=iteration, y-axis=loss, Matplotlib or other plot tools is available to use)



2. (10%) What’s the Cross Entropy Error of your prediction and ground truth?

```
Cross entropy:  46.975369108431934
```

3.
4. (10%) What’re the weights and intercepts of your linear model?

```
Intercept:  1.6625744220990193
Weight:  4.7963816709920195
```

__**Print the answers from your code and paste them onto the report**__

Part. 2, Questions (40%):

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

Gradient Descent: 每次都使用全部資料，沿著negative gradient direction尋找error最低點。可以準確找到方向，但當資料量太大時會需要大量時間。

Mini-Batch Gradient Descent: SGD和Gradient Descent的折衷版，一次使用m筆資料尋找方向，快速且能大約找到正確方向。

Stochastic Gradient Descent: 一次使用一筆資料尋找方向，快速，但因為資料過少，找到的gradient不一定會指向正確的方向。。

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

Gradient descent: 新方向 = 舊方向 - learning rate*gradient direction。learning rate為新方向中參考gradient direction的權重。當learning rate太低時，會使方向更新速度緩慢，花費大量時間。而learning rate太高時，可能會造成overshoot，無法收斂至最低點。

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

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

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

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

$$\sigma(a) = y \rightarrow \sigma^{-1}(y) = a$$

$$\frac{1}{1+e^{-a}} = y \qquad 1 = y + ye^{-a} \qquad \frac{1-y}{y} = e^{-a} \qquad \ln\left(\frac{1-y}{y}\right) = -a$$

$$-\ln\left(\frac{1-y}{y}\right) = a$$

$$\sigma^{-1}(y) = a = \ln\left(\frac{y}{1-y}\right) \#$$

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

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

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

Hints:

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

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

eq 2: Negative log likehood function

$\hookrightarrow \dfrac{\partial E}{\partial y_{nk}} = \dfrac{-t_{nk}}{y_{nk}}$ A

$\dfrac{\partial y_k}{\partial a_j} = y_k(I_{kj} - y_j)$, $I_{kj} = \begin{cases} 1, & j = k \\ 0, & \text{otherwise} \end{cases}$ B

A B

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

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

$\nabla_{w_j} a_{nj} = \phi_n$

$\Rightarrow \nabla_{w_j} E(w_1 \cdots w_K) = \sum_{n=1}^{N} \boxed{\dfrac{\partial E}{\partial a_{nj}}} \boxed{\nabla_{w_j} a_{nj}} = \sum_{n=1}^{N} (\boxed{y_{nj} - t_{nj}}) \boxed{\phi_n}$