Report of

NYCU Introduction to Machine Learning, Homework 1

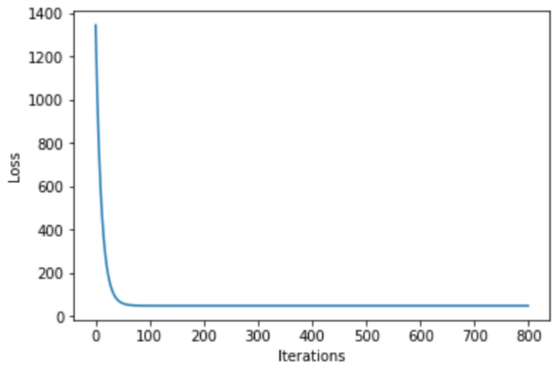
109550027 紀竺均

Part. 1, Coding (60%):

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)

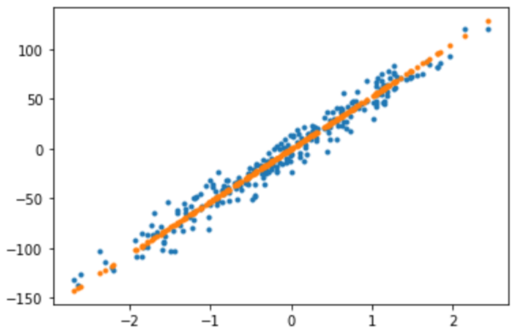
```
[11] plt.plot(np.arange(len(trainloss)), trainloss)
     plt.ylabel('Loss')
     plt.xlabel('Iterations')
     plt.show()
```



2. (10%) What's the Mean Square Error of your prediction and ground truth?
   Ans: Mean_square_error: 55.21909628061997

```
y_pred = predict(x_test, b0, b1)
print("Mean_square_error: ",Mean_square_error(y_test, y_pred))
#plt.plot(x_train, y_train,'.')
plt.plot(x_test, y_test,'.')
plt.plot(x_test, y_pred, '.') # regression line
```

```
Mean_square_error:  55.21909628061997
[<matplotlib.lines.Line2D at 0x7f925cd76b10>]
```



3. (10%) What're the weights and intercepts of your linear model?
   Ans: weights: 52.743540461824786, intercepts: -0.33375889502567796

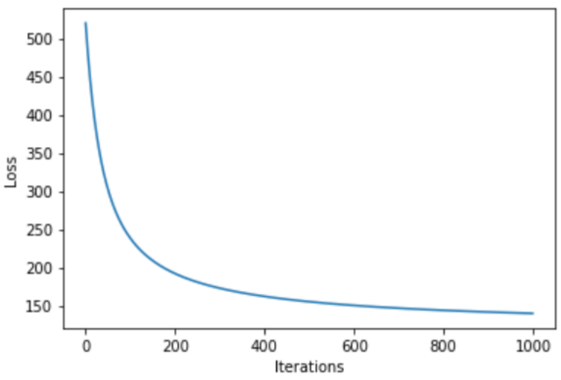
```
b0, b1  =  gradiDes()
print("b0:  ", b0, "b1:  ", b1)
```

```
b0:  -0.33375889502567796 b1:  52.743540461824786
```

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)

```
plt.plot(np.arange(len(trainloss_lo)),  trainloss_lo)
plt.ylabel('Loss')
plt.xlabel('Iterations')
plt.show()
```

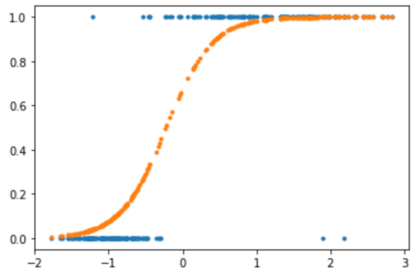


2. (10%) What's the Cross Entropy Error of your prediction and ground truth?

   Ans: Cross_entropy_error: 45.225920481381024

```
y_pred  =  predict_lo(x_test, b0, b1)
print("Cross_entropy_error:  ", Cross_entropy_error(y_test,  y_pred))
#plt.plot(x_train, y_train, '.')
plt.plot(x_test, y_test, '.')
plt.plot(x_test,  y_pred,  '.')   #  regression  line
```

```
Cross_entropy_error:  45.225920481381024
[<matplotlib.lines.Line2D at 0x7f2fc4b2c890>]
```



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

   Ans: weights: 3.239820397263054, itercepts: 0.7365083362524347

```
    return  b0, b1
b0, b1  =  gradiDes_lo()
print("b0:  ", b0, "b1:  ", b1)
```

```
b0:  0.7365083362524347 b1:  3.239820397263054
```

Part. 2, Questions (40%):

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

Ans: The main difference is the number of training data taken into consideration within a single step. For Gradient Descent, it takes **all the training data** into consideration; For Stochastic Gradient Descent, it only consider **one example** at a time; Finally, Mini-Batch Gradient Descent is a mixture of Gradient Descent and SGD, it takes **a batch of fixed number** of training data into consideration.

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

Ans: Yes, if the learning rate is too large, gradient descent might overshoot the minimum. It may fail to converge. On the other hand, if the learning is too small, it would take more steps and more time to reach local minimum.

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

$$\sigma(a) = \frac{1}{1 + \exp(-a)} \tag{4.59}$$

Ans: Next Page

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)

$$\nabla_{\mathbf{w}_j} E(\mathbf{w}_1, \ldots, \mathbf{w}_K) = \sum_{n=1}^{N} (y_{nj} - t_{nj})\, \boldsymbol{\phi}_n \tag{4.109}$$

(eq.3)

$$a_k = \mathbf{w}_k^{\mathrm{T}} \boldsymbol{\phi}. \tag{4.105}$$

(eq. 4)

$$\frac{\partial y_k}{\partial a_j} = y_k (I_{kj} - y_j) \tag{4.106}$$

(eq. 5)

Ans: Next Page

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

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

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

$$=\frac{2+e^{a}+e^{-a}}{1+e^{-a}+e^{a}+e^{0}}=\frac{2+e^{a}+e^{-a}}{2+e^{a}+e^{-a}}=1$$

$\Rightarrow \sigma(-a)=1-\sigma(a)$ ※

② $\sigma(y)=\frac{1}{1+e^{-y}}\rightarrow 1+e^{-y}=\frac{1}{\sigma(y)}$

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

$\xrightarrow{\text{取}\ln}$ $-y=\ln\frac{1-\sigma(y)}{\sigma(y)}\Rightarrow y=-\ln\frac{1-\sigma(y)}{\sigma(y)}$

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

4. $E(w_1,\ldots w_K)=-\sum_{n=1}^{N}\sum_{k=1}^{K}t_{nk}\ln y_{nk}$

对其中一项 $k=j$ 作微分 $\left(\frac{\partial E}{\partial w_j}\right)$

$$\frac{\partial E}{\partial w_j}=\frac{\partial E}{\partial y_{nk}}\cdot\frac{\partial y_{nk}}{\partial a_{nj}}\cdot\frac{\partial a_j}{\partial w_j}$$

$$=\sum_{n=1}^{N}\left(-\sum_{k=1}^{K}t_{nk}\frac{1}{y_{nk}}\right)\frac{\partial y_{nk}}{\partial a_{nj}}\cdot\frac{\partial a_{nj}}{\partial w_j}$$

(eq.5)

$$=\sum_{n=1}^{N}\left(-\sum_{k=1}^{K}\frac{t_{nk}}{y_{nk}}\cdot y_{nk}(I_{kj}-y_{nj})\right)\cdot\frac{\partial a_{nj}}{\partial w_j}$$

$$=\sum_{n=1}^{N}\left(-\sum_{k=1}^{K}t_{nk}(I_{kj}-y_{nj})\right)\frac{\partial a_{nj}}{\partial w_j}$$

1-of-K coding
$\sum_{k}t_{nk}=1$

$$=\sum_{n=1}^{N}\left(-t_{nj}+\sum_{k=1}^{K}t_{nk}y_{nj}\right)\cdot\frac{\partial a_{nj}}{\partial w_j}$$

$$=\sum_{n=1}^{N}(-t_{nj}+y_{nk})\frac{\partial a_{nj}}{\partial w_j}$$

∵ $a_{nk}=w_k^T\phi_n$
$\frac{\partial a_{nk}}{\partial w_k}=\phi_n$

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