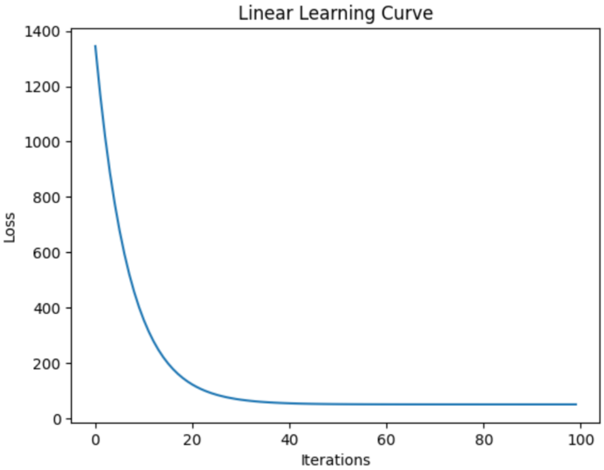
NYCU Introduction to Machine Learning, Homework 1

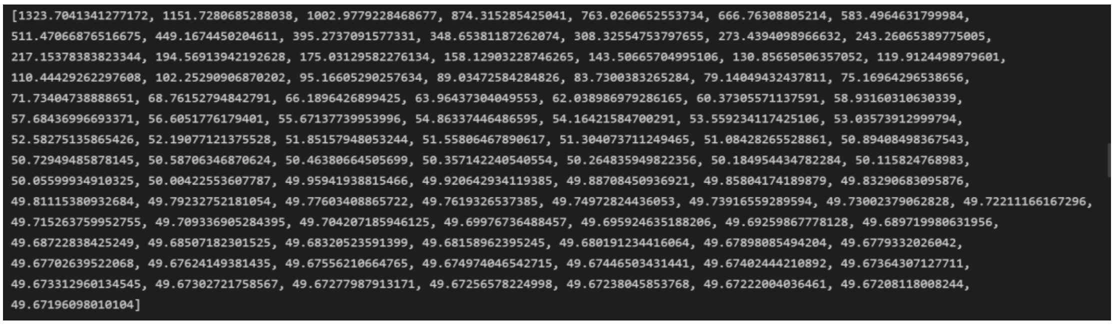
Part. 1, Coding (60%):

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)



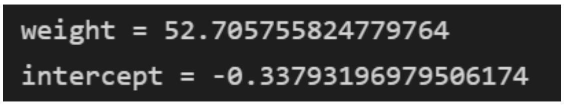
2. (10%) What's the Mean Square Error of your prediction and ground truth?



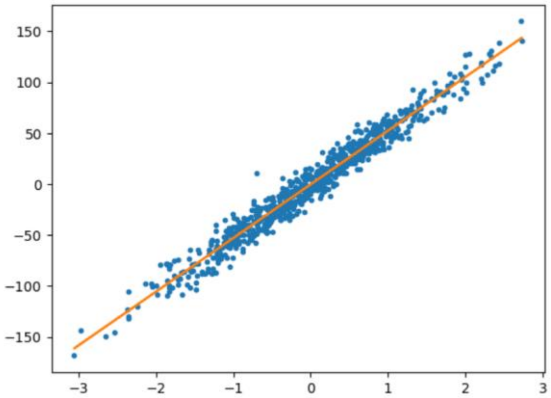
```
[1323.7041341277172, 1151.7280685288038, 1002.9779228468677, 874.315285425041, 763.0260652553734, 666.76308805214, 583.4964631799984,
511.47066876516675, 449.1674450204611, 395.2737091577331, 348.65381187262074, 308.32554753797655, 273.4394098966632, 243.26065389775005,
217.15378383823344, 194.56913942192628, 175.03129582276134, 158.12903228746265, 143.50665704995106, 130.85650506357052, 119.9124498979601,
110.44429262297608, 102.25290906870202, 95.16605290257634, 89.03472584284826, 83.7300383265284, 79.14049432437811, 75.16964296538656,
71.7340473888651, 68.76152794842791, 66.1896426899425, 63.96437304049553, 62.038986979286165, 60.37305571137591, 58.93160310630339,
57.68436996693371, 56.6051776179401, 55.67137739953996, 54.86337446486595, 54.16421584700291, 53.559234117425106, 53.03573912999794,
52.58275135865426, 52.19077121375528, 51.85157948053244, 51.55806467890617, 51.304073711249465, 51.08428265528861, 50.89408498367543,
50.72949485878145, 50.58706346870624, 50.46380664505699, 50.357142240540554, 50.264835949822356, 50.184954434782284, 50.115824768983,
50.05599934910325, 50.00422553607787, 49.95941938815466, 49.920642934119385, 49.88708450936921, 49.85804174189879, 49.83290683095876,
49.81115380932684, 49.79232752181054, 49.77603408865722, 49.7619326537385, 49.74972824436053, 49.73916559289594, 49.73002379062828, 49.7221116167296,
49.715263759952755, 49.709336905284395, 49.704207185946125, 49.69976736488457, 49.695924635188206, 49.69259867778128, 49.689719980631956,
49.68722838425249, 49.68507182301525, 49.68320523591399, 49.68158962395245, 49.680191234416064, 49.67898085494204, 49.6779332026042,
49.67702639522068, 49.67624149381435, 49.67556210664765, 49.674974046542715, 49.67446503431441, 49.67402444210892, 49.67364307127711,
49.673312960134545, 49.67302721758567, 49.67277987913171, 49.67256578224998, 49.67238045853768, 49.67222004036461, 49.67208118008244,
49.67196098010104]
```

3.

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

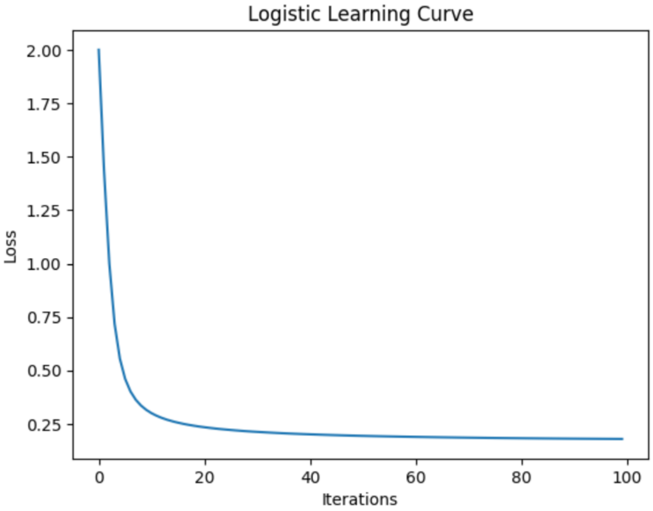


```
weight = 52.705755824779764
intercept = -0.33793196979506174
```

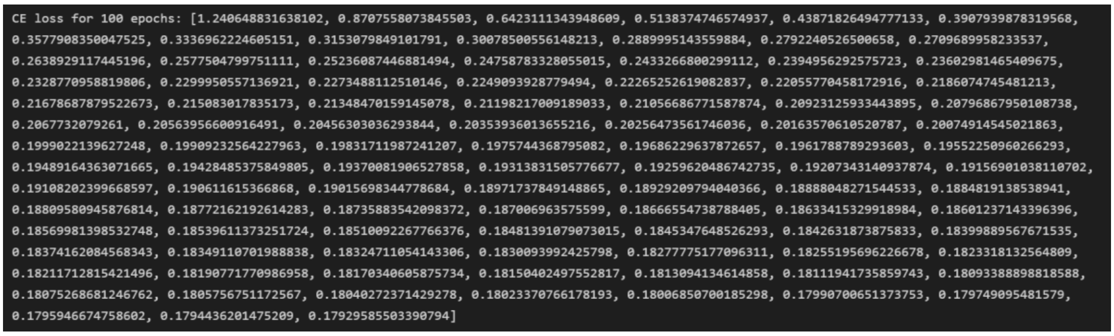


Logistic regression model

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)

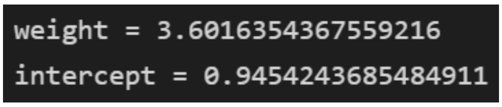


2. (10%) What's the Cross Entropy Error of your prediction and ground truth?

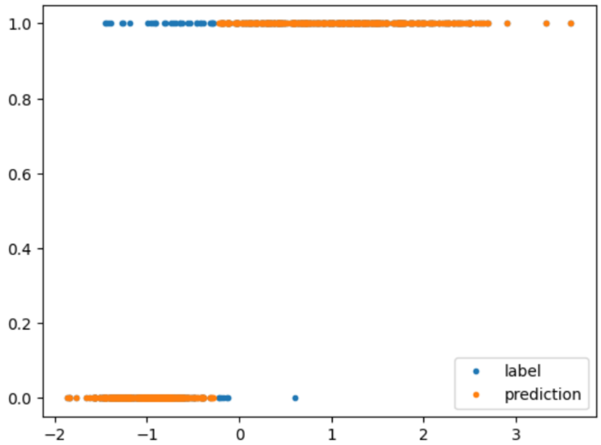


```
CE loss for 100 epochs: [1.240648831638102, 0.8707558073845503, 0.6423111343948609, 0.5138374746574937, 0.43871826494777133, 0.3907939878319568,
0.3577908350047525, 0.3336962224605151, 0.3153079849101791, 0.30078500556148213, 0.2889995143559884, 0.2792240526500658, 0.2709689958233537,
0.2638929117445196, 0.2577504799751111, 0.2523608744688149 4, 0.24758783328055015, 0.2433266800299112, 0.2394956292575723, 0.2360298146540967 5,
0.2328770958819806, 0.2299950557136921, 0.2273488112510146, 0.2249093928779494, 0.2226525261908283 7, 0.22055770458172916, 0.218607474548121 3,
0.2167868787952267 3, 0.215083017835173, 0.2134847015914507 8, 0.21198217009189033, 0.21056686771587874, 0.20923125933443895, 0.20796867950108738,
0.2067732079261, 0.20563956600916491, 0.20456303036293844, 0.20353936013655216, 0.2025647356174603 6, 0.20163570610520787, 0.2007491454502186 3,
0.1999022139627248, 0.19909232564227963, 0.19831711987241207, 0.1975744368795082, 0.19686229637872657, 0.1961788789293603, 0.1955225096026629 3,
0.19489164363071665, 0.19428485375849805, 0.19370081906527858, 0.19313831505776677, 0.19259620486742735, 0.19207343140937874, 0.19156901038110702,
0.19108202399668597, 0.190611615366868, 0.19015698344778684, 0.18971737849148865, 0.18929209794040366, 0.18888048271544533, 0.18848191385389 41,
0.18809580945876814, 0.18772162192614283, 0.18735883542098372, 0.187006963575599, 0.18666554738788405, 0.18633415329918984, 0.18601237143396396,
0.18569981398532748, 0.18539611373251724, 0.18510092267766376, 0.18481391079073015, 0.1845347648526293, 0.1842631873875833, 0.18399889567671535,
0.18374162084568343, 0.18349110701988838, 0.18324711054143306, 0.1830093992425798, 0.18277775177096311, 0.18255195696226678, 0.182331813256480 9,
0.18211712815421496, 0.18190771770986958, 0.18170340605875734, 0.18150402497552817, 0.1813094134614858, 0.18111941735859743, 0.18093388898818588,
0.18075268681246762, 0.1805756751172567, 0.18040272371429278, 0.18023370766178193, 0.18006850700185298, 0.17990700651373753, 0.179749095481579,
0.1795946674758602, 0.1794436201475209, 0.17929585503390794]
```

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



```
weight = 3.6016354367559216
intercept = 0.9454243685484911
```



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 compute gradients every epoch, while Mini-Batch Gradient Descent calculates gradients every mini-batch, and Stochastic Gradient Descent calculate gradients for every single input data.

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

   Yes, if the learning rate is too large, it can cause the model to converge into local minimum, which is a suboptimal solution. On the other hand, if the learning rate is too small, it might converge very slowly.

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)}.

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

(eq. 1)

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

$$E(\mathbf{w}_1,\ldots,\mathbf{w}_K) = -\ln p(\mathbf{T}|\mathbf{w}_1,\ldots,\mathbf{w}_K) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk}\ln y_{nk} \tag{4.108}$$

(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)

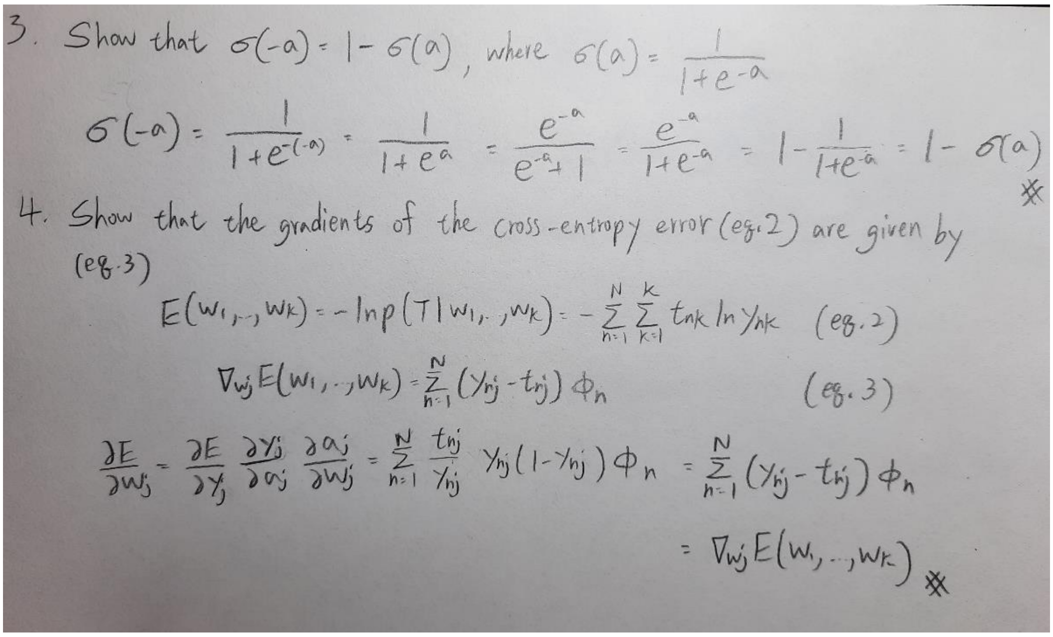
Hints:

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

(eq. 4)

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

(eq. 5)



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

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

⋕

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

$$E(w_1, \ldots, w_K) = -\ln p(T \mid w_1, \ldots, w_K) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_{nk} \quad \text{(eq.2)}$$

$$\nabla_{w_j} E(w_1, \ldots, w_K) = \sum_{n=1}^{N} (y_{nj} - t_{nj}) \phi_n \quad \text{(eq.3)}$$

$$\frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial a_j} \frac{\partial a_j}{\partial w_j} = \sum_{n=1}^{N} \frac{t_{nj}}{y_{nj}} y_{nj}(1 - y_{nj}) \phi_n = \sum_{n=1}^{N} (y_{nj} - t_{nj}) \phi_n$$

$$= \nabla_{w_j} E(w_1, \ldots, w_K)$$

⋕