Week_4 programing problem

1. 資料整理
   - 將資料分為 regression_dataset 和 classification_dataset
   - 以 csv 的型態儲存在 colab 之中，以利後續代碼直接取用
   - Regression – 以(經度，緯度，溫度)的方式一列一列儲存。
   - Classification – 以(經度，緯度，溫度，分類值)的方式一列一列儲存
   - 代碼詳見 CodeOfSort(on github)，代碼提共自與 ChatGPT 討論並在 colab 上運行
   - 輸出結果

     classification_dataset.csv

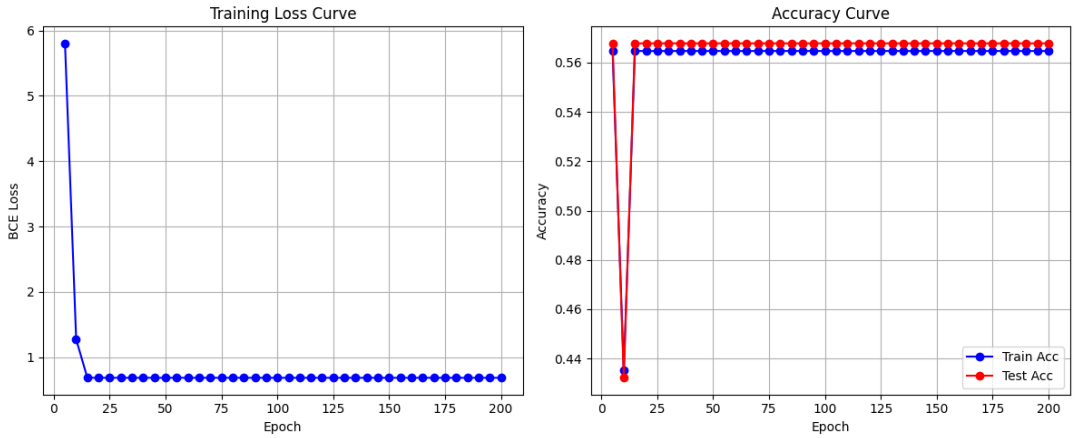
     regression_dataset.csv

2. 分類資料集機器學習模型
   - 輸入(資料來源) – classification_dataset.csv
   - 模型 – MLP (一層隱藏層)
     - 輸入:經度、緯度
     - 隱藏層 16 個神經元
     - 激活函數(activation function):ReLU
     - 輸出一維結果
   - 訓練策略:
     - 將資料隨機分為 80%的訓練集與 20%的測試集，為了確保訓練出來的結果不是特別針對某些特定的數據。
     - 優化器:SGD(梯度下降)
     - 學習率(learning rate)設定 – 使用 r = 0.01 迭代 100 次 r = 0.001 迭代 100 次。
     - Loss function -

$$L = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\sigma(x_i)) + (1-y_i)\log(1-\sigma(x_i))\right]$$

     - 觀察方式:每五次迭代輸出訓練與測試集的 loss
     - 結果呈現:
       - **Loss 曲線**：訓練過程中 Loss 隨 迭代次數 下降的趨勢
       - Accuracy 曲線

```
Epoch   5 | LR=0.01 | Loss: 5.804045 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch  10 | LR=0.01 | Loss: 1.271129 | Train Acc: 0.4353 | Test Acc: 0.4322
Epoch  15 | LR=0.01 | Loss: 0.683272 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch  20 | LR=0.01 | Loss: 0.683215 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch  25 | LR=0.01 | Loss: 0.683190 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch  30 | LR=0.01 | Loss: 0.683165 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch  35 | LR=0.01 | Loss: 0.683141 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch  40 | LR=0.01 | Loss: 0.683118 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch  45 | LR=0.01 | Loss: 0.683095 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch  50 | LR=0.01 | Loss: 0.683073 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch  55 | LR=0.01 | Loss: 0.683051 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch  60 | LR=0.01 | Loss: 0.683030 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch  65 | LR=0.01 | Loss: 0.683009 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch  70 | LR=0.01 | Loss: 0.682988 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch  75 | LR=0.01 | Loss: 0.682969 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch  80 | LR=0.01 | Loss: 0.682949 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch  85 | LR=0.01 | Loss: 0.682931 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch  90 | LR=0.01 | Loss: 0.682912 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch  95 | LR=0.01 | Loss: 0.682894 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 100 | LR=0.01 | Loss: 0.682877 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 105 | LR=0.001 | Loss: 0.682872 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 110 | LR=0.001 | Loss: 0.682870 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 115 | LR=0.001 | Loss: 0.682868 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 120 | LR=0.001 | Loss: 0.682867 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 125 | LR=0.001 | Loss: 0.682865 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 130 | LR=0.001 | Loss: 0.682863 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 135 | LR=0.001 | Loss: 0.682862 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 140 | LR=0.001 | Loss: 0.682860 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 145 | LR=0.001 | Loss: 0.682858 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 150 | LR=0.001 | Loss: 0.682856 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 155 | LR=0.001 | Loss: 0.682855 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 160 | LR=0.001 | Loss: 0.682853 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 165 | LR=0.001 | Loss: 0.682851 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 170 | LR=0.001 | Loss: 0.682850 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 175 | LR=0.001 | Loss: 0.682848 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 180 | LR=0.001 | Loss: 0.682847 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 185 | LR=0.001 | Loss: 0.682845 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 190 | LR=0.001 | Loss: 0.682843 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 195 | LR=0.001 | Loss: 0.682841 | Train Acc: 0.5647 | Test Acc: 0.5678
Epoch 200 | LR=0.001 | Loss: 0.682840 | Train Acc: 0.5647 | Test Acc: 0.5678
```



- 結論:
  - 在測試的過程中，在 0.01 學習率的 50 次迭代之後，無論如何調整學習率與迭代次數都很難繼續降低 Loss 並有效的提高精準度，可以從數據與圖表中看到調整的幅度微乎極微
  - 模型的結構過於簡單，對比資料是一個非線性的大量資料，導致模型能夠逼近的程度有限，因此，增加整體模型的複雜度也許能讓逼近的程度上升。

3. 回歸資料集機器學習模型

- 輸入(資料來源) – regression_dataset.csv
- 模型 – MLP (一層隱藏層)
  - 輸入:經度、緯度
  - 隱藏層 16 個神經元
  - 激活函數(activation function):ReLU
  - 輸出一維結果(溫度)
- 訓練策略:
  - 將資料隨機分為 80%的訓練集與 20%的測試集，為了確保訓練出來的結果不是特別針對某些特定的數據。
  - 優化器:SGD(梯度下降)
  - 學習率(learning rate)設定 – 使用 r = 0.01 迭代 400 次 r = 0.001 迭代 100 次。
  - Loss function - MSE (Mean Squared Error)

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

  - 觀察方式:每 20 次迭代輸出訓練與測試集的 loss
  - 結果呈現:
  - Loss 圖表 ：訓練過程中 Loss 隨 迭代次數下降的趨勢

```
== 第一階段訓練: lr=0.01, 400 epochs ==
Epoch  20 | LR=0.01 | Loss: 4614361.000000
Epoch  40 | LR=0.01 | Loss: 2056643.625000
Epoch  60 | LR=0.01 | Loss: 916667.812500
Epoch  80 | LR=0.01 | Loss: 408580.125000
Epoch 100 | LR=0.01 | Loss: 182125.109375
Epoch 120 | LR=0.01 | Loss: 81194.000000
Epoch 140 | LR=0.01 | Loss: 36208.960938
Epoch 160 | LR=0.01 | Loss: 16159.110352
Epoch 180 | LR=0.01 | Loss: 7222.881348
Epoch 200 | LR=0.01 | Loss: 3240.001465
Epoch 220 | LR=0.01 | Loss: 1464.830200
Epoch 240 | LR=0.01 | Loss: 673.635620
Epoch 260 | LR=0.01 | Loss: 320.999817
Epoch 280 | LR=0.01 | Loss: 163.829956
Epoch 300 | LR=0.01 | Loss: 93.779282
Epoch 320 | LR=0.01 | Loss: 62.557648
Epoch 340 | LR=0.01 | Loss: 48.642139
Epoch 360 | LR=0.01 | Loss: 42.440010
Epoch 380 | LR=0.01 | Loss: 39.675709
Epoch 400 | LR=0.01 | Loss: 38.443668

== 第二階段訓練: lr=0.001, 100 epochs ==
Epoch  20 | LR=0.001 | Loss: 38.334713
Epoch  40 | LR=0.001 | Loss: 38.266853
Epoch  60 | LR=0.001 | Loss: 38.204224
Epoch  80 | LR=0.001 | Loss: 38.146412
Epoch 100 | LR=0.001 | Loss: 38.093048

測試集 MSE: 39.273720
```

- 結論:和第一題有著一樣的問題，在 loss 達到 38 左右時，無論如何調整學習率，皆很難有效繼續減少 loss，若是增加模型內隱藏層的複雜度，也許能更加有效的訓練模型逼近實際數據。