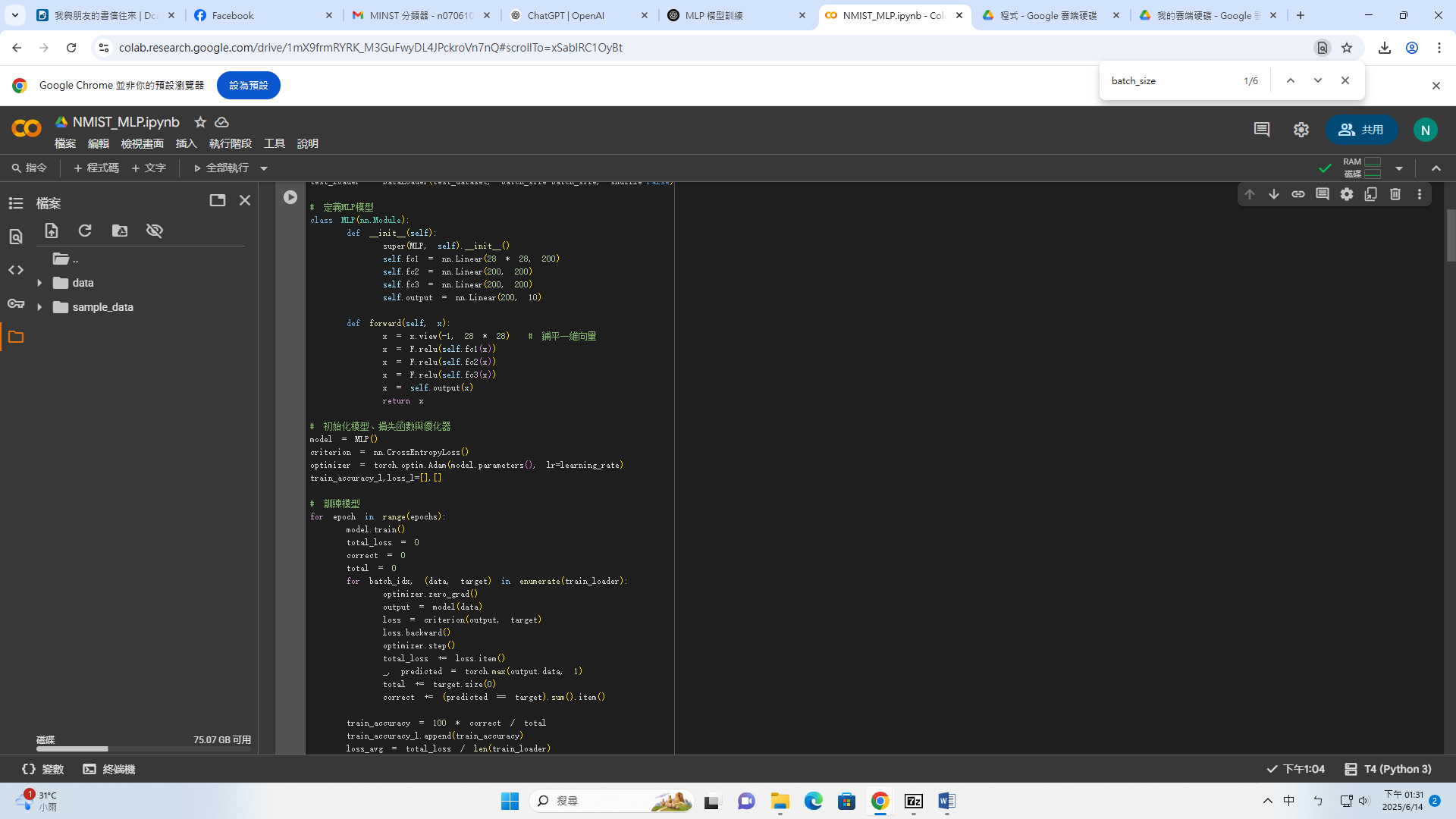
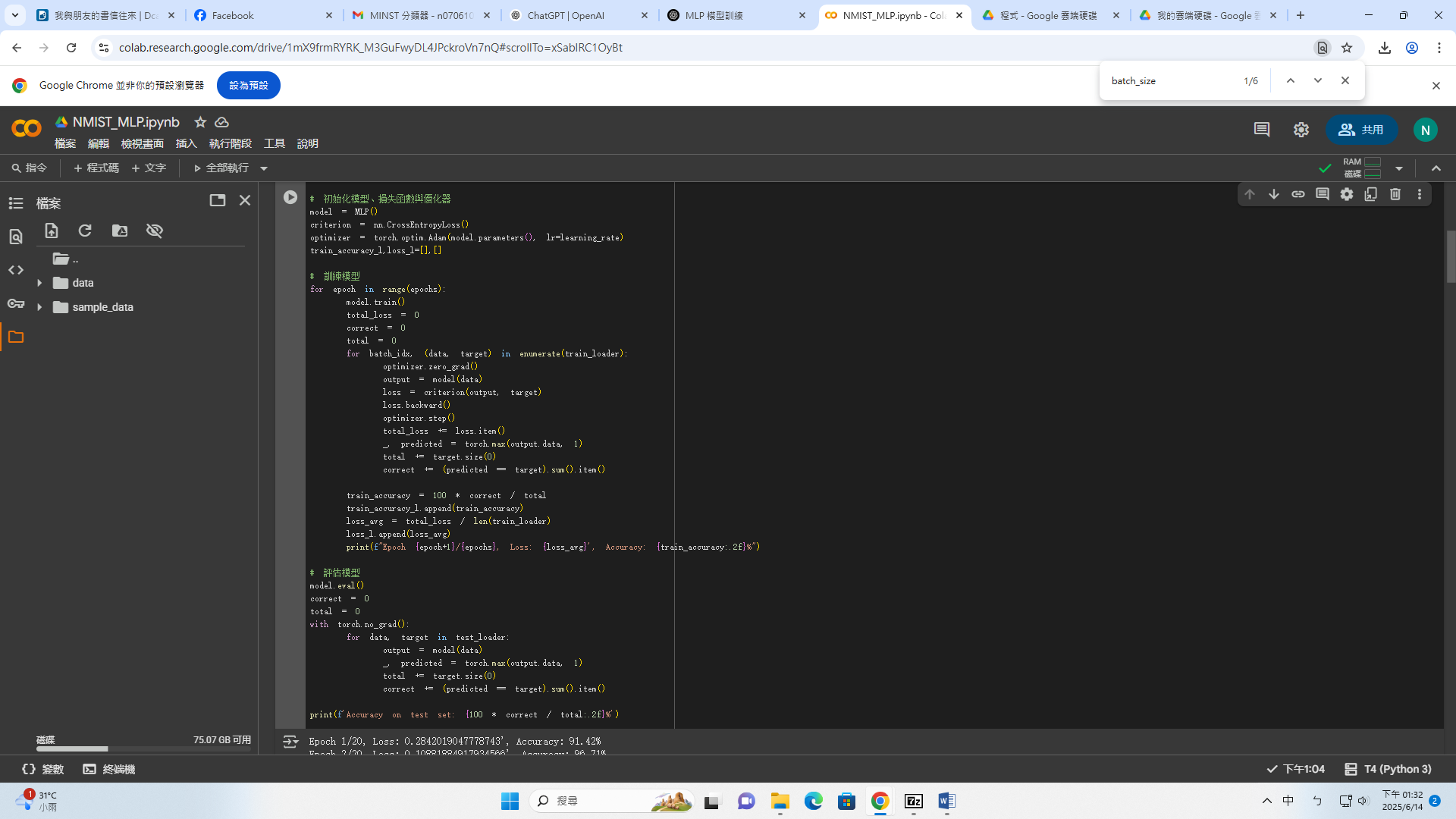
1. 程式碼截圖

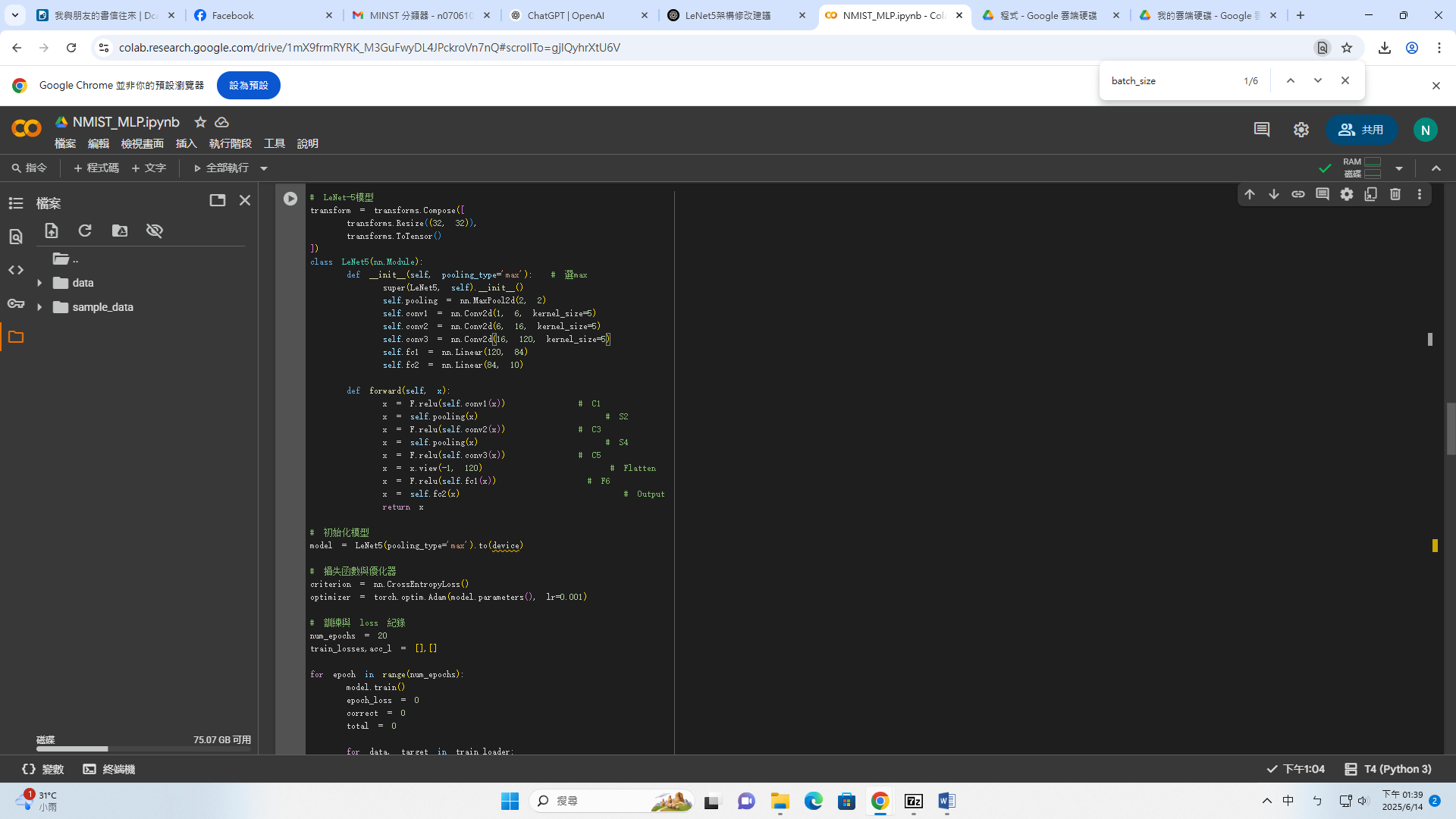
MLP定義模型



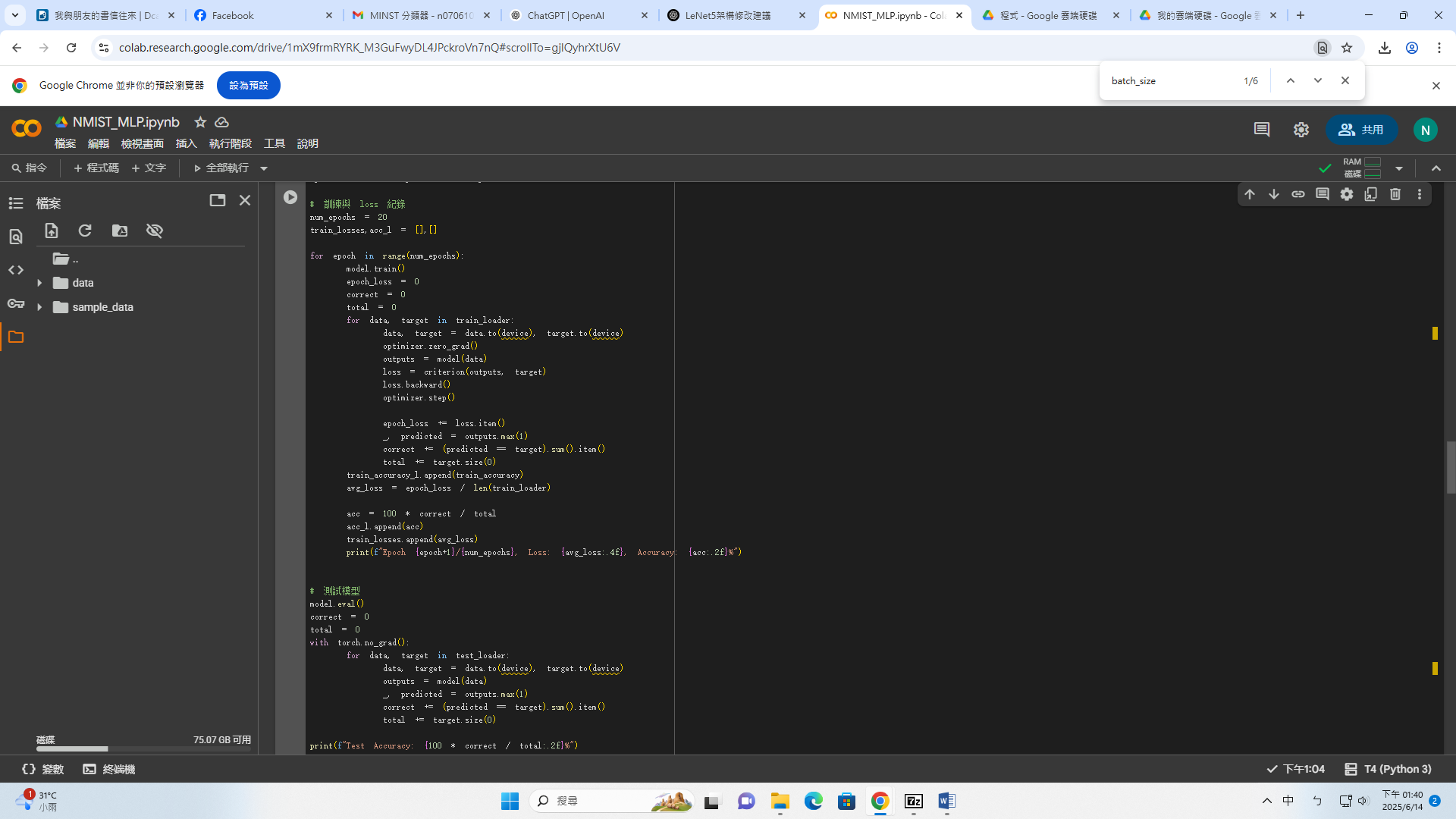
訓練跟評估



LeNet-5定義模型

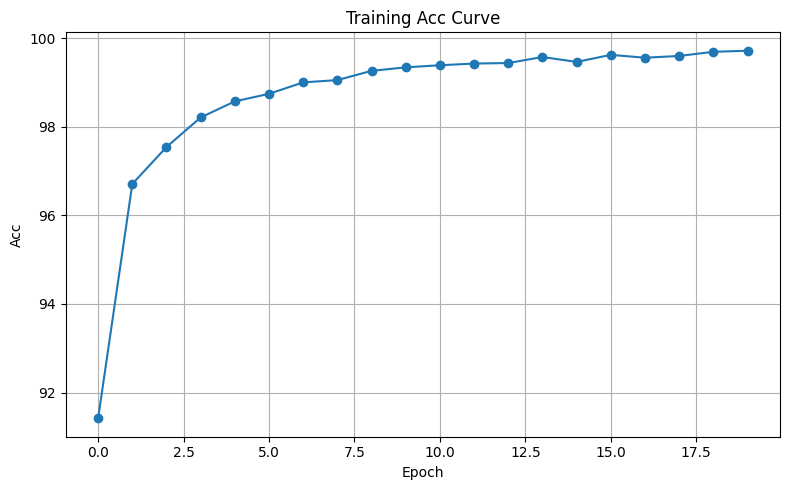


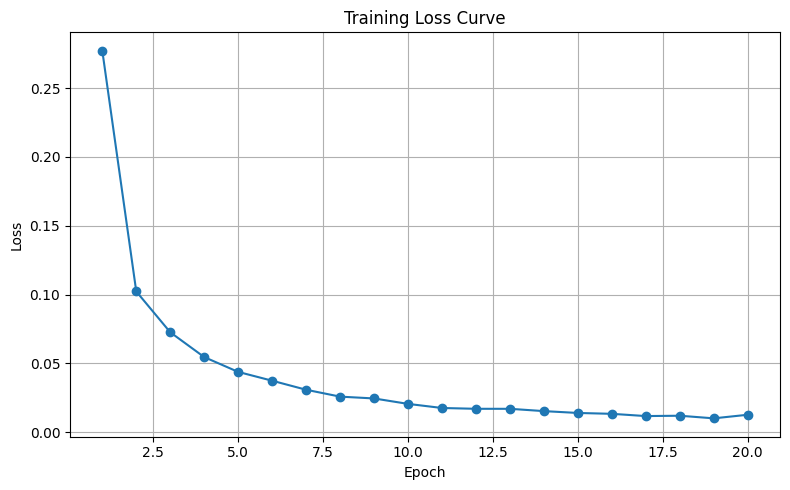
訓練跟評估



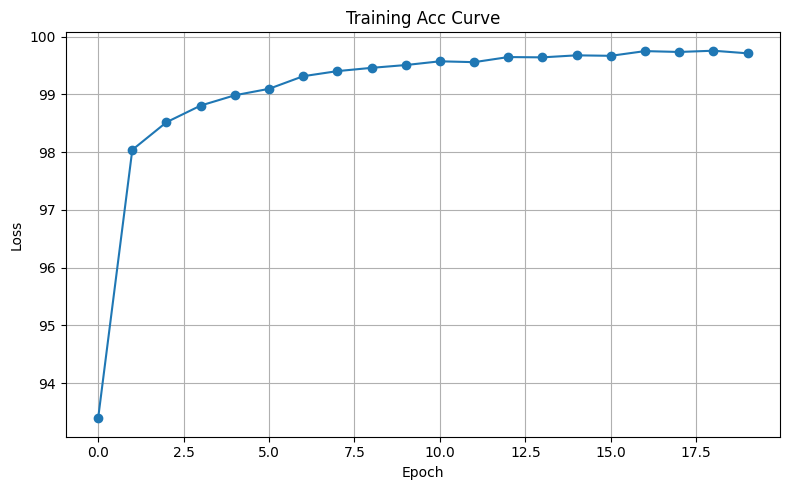
1. 實驗

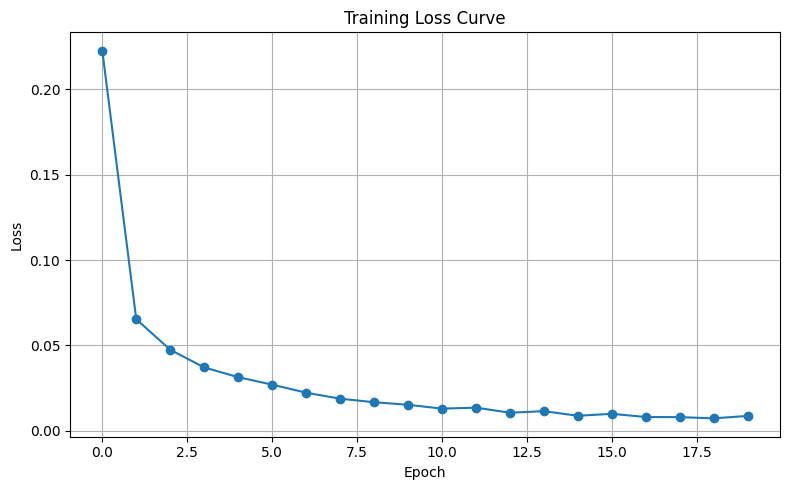
MLP結果





LeNet-5結果





以 learning rate=0.001 和 batch size=64 比較 MLP 和 LeNet-5。

LeNet-5因為局部卷積能提取特徵，並且引入參數共享與池化層，提升泛化能力，因此一開始訓練準確度比MLP還高，最終也是比MLP高，若是數據更複雜應該可以看出更大差異。

以 LeNet-5,batch size=64,比較不同 learning rate:0.1、0.01、0.001。

|  |  |
| --- | --- |
| 條件 | 最終準確率 |
| 0.1 | 10.3% |
| 0.01 | 98.7% |
| 0.001 | 98.6% |

以 LeNet-5,learning rate=0.001,比較不同 batch size:8、64、128。

|  |  |
| --- | --- |
| 條件 | 最終準確率 |
| 8 | 98.6% |
| 64 | 99.7% |
| 128 | 99.6% |

以 LeNet-5,learning rate=0.001,batch size=64,比較 SGD、SGD + momentum (0.95) 和 Adam optimizer。

|  |  |
| --- | --- |
| 條件 | 最終準確率 |
| SGD | 99.8% |
| 64 | 99.7% |
| 128 | 99.9% |

1. 根據上述的實驗進行說明

過高的 learning rate 會導致模型在 loss surface 上跳來跳去，學習不穩定，也容易找不到最佳結果，過低則學習緩慢，可能在有限epoch內無法學好模型，而實驗中0.01或0.001就差異不大，訓練時間差不多。

對batch size而言，小batch size提供更隨機的梯度更新，大batch size對記憶體需求高，梯度平均效果好，而實驗中各種條件差異不大。

比較MLP與LeNet-5，MLP缺乏特徵處理能力，只看數值，不知道像素間關係，因此在圖像分類理論上不好，LeNet-5用卷積層學習特徵，並用 pooling 降低維度與計算量，因此能更有效學習影像特徵，本實驗猜測是數據過於簡單導致最終成果差異不大，有觀察到LeNet運算時間快，推測是因為訓練參數少所致。