1.(30 points) Describe your model architecture details and PCA method.

PCA method:

使用 USE_PCA 變數設置為 True，表示將使用 PCA 來降低輸入數據的維度。

從 sklearn.decomposition 模塊導入 PCA，將 NUM_COMPOSITIONS 根據題目設置維度為 2，保留主成分為 2。

將輸入的圖像數據進行主成分分析(PCA)處理，返回處理後的特徵數據，具體来說，代碼首先將輸入的圖像數據進行重塑，將每個圖像的像素值展開為一維數組，儲存到 self.images_reshapes 中。

接著，代碼根據輸入的 spilt 參數，分別使用 fit_transform 和 transform 方法對 self.images_reshapes 進行 PCA 處理。這裡需要注意的是，對於訓練集數據，需要先使用 fit 方法擬合 PCA 模型，然後再使用 transform 方法進行處理；對於測試集數據，只需要使用 transform 方法即可，因為模型已經在訓練集上進行擬合。

My Two-layer model architecture details:

我定義了一個名為 NeuralNetwork 的類別，代表一個簡單的神經網路模型。程式碼包含以下變數和函數：

device：代表設備的名稱，這裡設為 "cpu"，表示在 CPU 上運行程式碼。

IMAGE_H 和 IMAGE_W：代表圖像的高度和寬度，設為 32。

NUM_CLASSES：代表類別的數量，這裡設為 3。

NeuralNetwork：類別的構造函數，定義了模型的基本結構和參數。

構造函數的參數包括 input_size、hidden_size = 512 和 output_size = 3，分別代表輸入層、隱藏層和輸出層的神經元數量。

模型的權重和偏差是隨機初始化的。

sigmoid 函數：定義了 sigmoid 函數的計算方法，用於計算神經元的激活值。

softmax 函數：定義了 softmax 函數的計算方法，用於將神經元的輸出轉換為概率分佈。

forward 函數：定義了前向傳播的計算過程。根據模型的權重和偏差，計算神經元的加權輸入、激活值和輸出。

backward 函數：定義了反向傳播的計算過程。根據目標值和模型的輸出，計算神經元的梯度，然後更新權重和偏差。

model：代表神經網路模型的實例，根據 USE_PCA 變數的值選擇使用哪個輸入大小。如果 USE_PCA 為 True，則使用 NUM_COMPONENTS；否則，使用 IMAGE_H * IMAGE_W。

My Three-layer model architecture details:

我定義了一個名為 NeuralNetwork 的類別，代表一個簡單的神經網

路模型。程式碼包含以下變數和函數：

device：代表設備的名稱，這裡設為 "cpu"，表示在 CPU 上運行程式碼。

IMAGE_H 和 IMAGE_W：代表圖像的高度和寬度，設為 32。

NUM_CLASSES：代表類別的數量，這裡設為 3。

NeuralNetwork：類別的構造函數，定義了模型的基本結構和參數。構造函數的參數包括 input_size、hidden1_size = 512 和 hidden2_size = 256 和 output_size，分別代表輸入層、第一層隱藏層、第二層隱藏層和輸出層的神經元數量。

模型的權重和偏差是隨機初始化的。

sigmoid 函數：定義了 sigmoid 函數的計算方法，用於計算神經元的激活值。

softmax 函數：定義了 softmax 函數的計算方法，用於將神經元的輸出轉換為概率分佈。

forward 函數：定義了前向傳播的計算過程。根據模型的權重和偏差，計算神經元的加權輸入、激活值和輸出。

backward 函數：定義了反向傳播的計算過程。根據目標值和模型的輸出，計算神經元的梯度，然後更新權重和偏差。

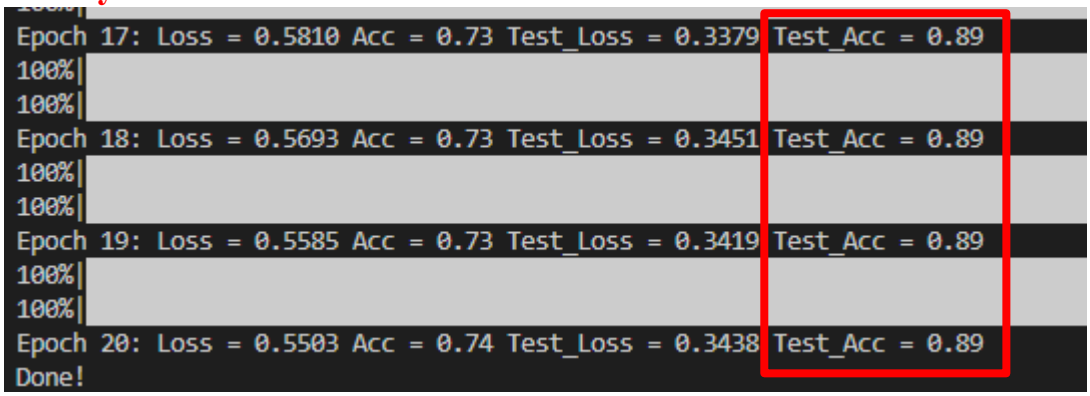
model：代表神經網路模型的實例，根據 USE_PCA 變數的值選擇

使用哪個輸入大小。如果 USE_PCA 為 True，則使用

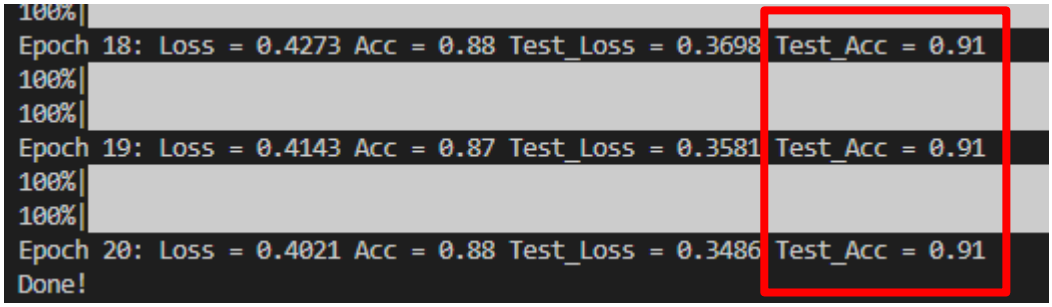
NUM_COMPONENTS；否則，使用 IMAGE_H * IMAGE_W。

2. (10 points) Show your test accuracy.

Two-layer neural Network:

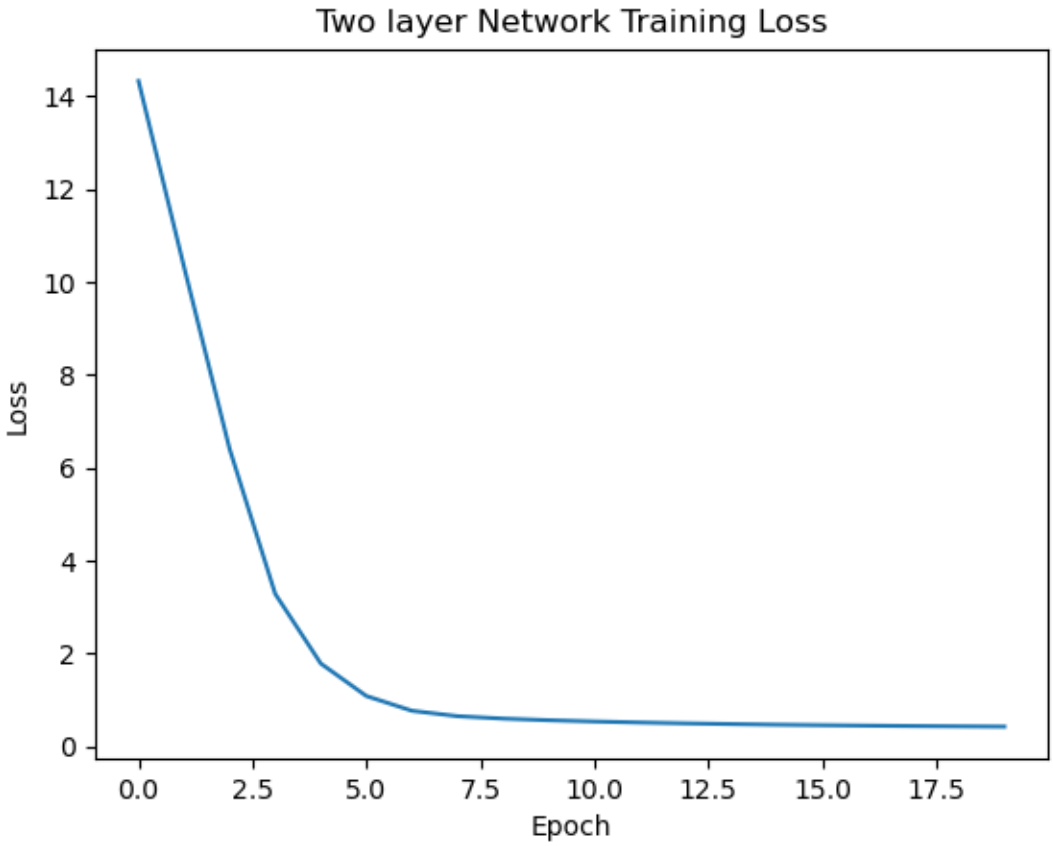


Three-layer neural Network:

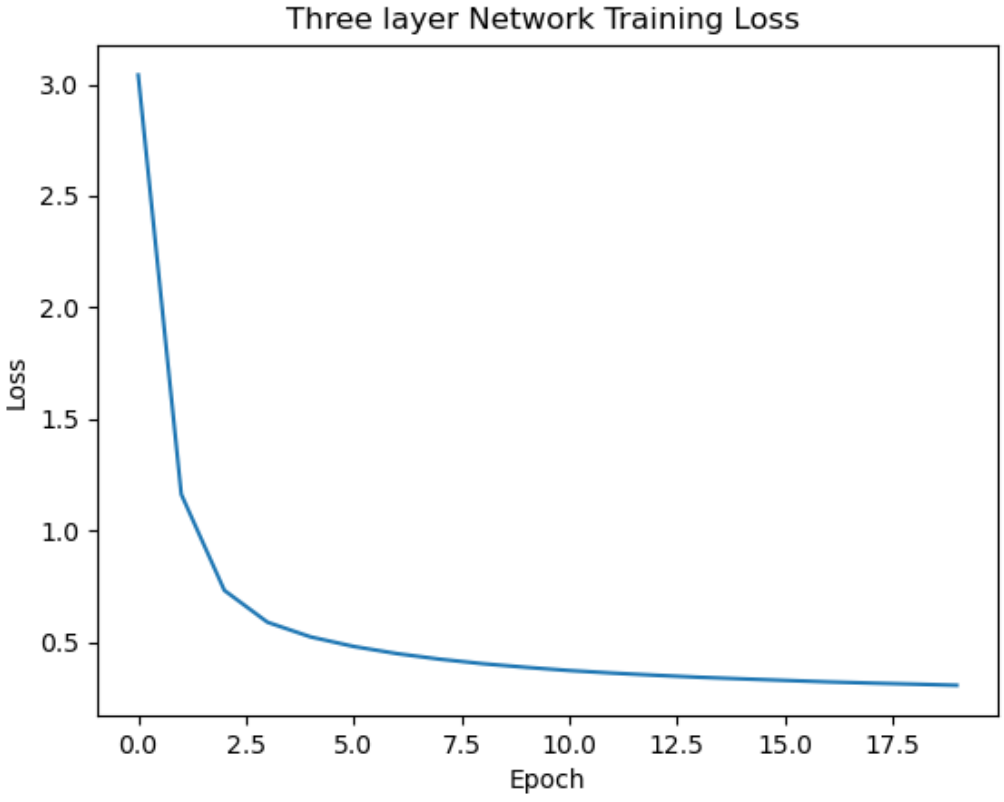


3. (10 points) Plot training loss curves.

Two-layer neural Network:

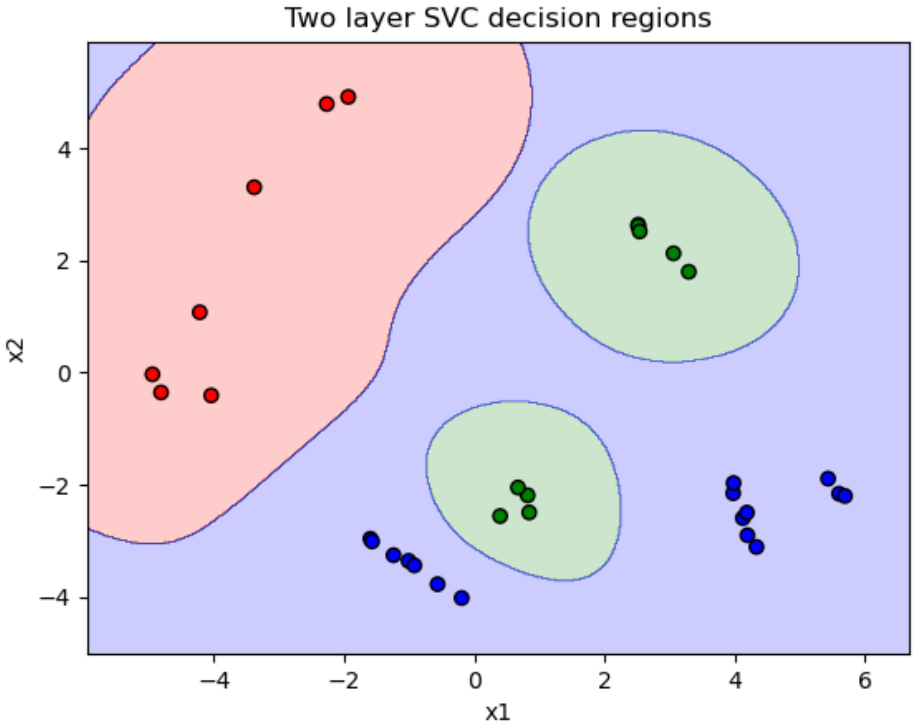


Three-layer neural Network:



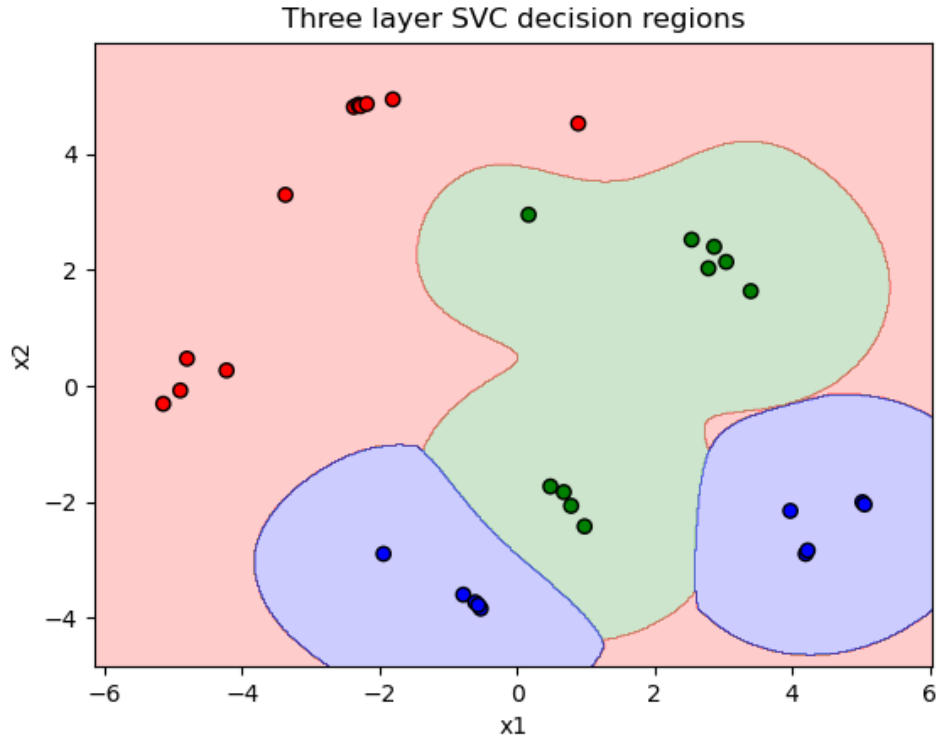
4. (20 points) Plot decision regions and discuss the training / testing performance with different settings designed by yourself

Two-layer neural Network decision regions:



基本上準確值都在預測值的範圍內

Three-layer neural Network decision regions:



基本上準確值都在預測值的範圍內

Discuss the training / testing performance with different settings:

Train_loss 方面：我設計的三層神經網路比兩層神經網路下降要快並且值更小

訓練正確值方面：我設計的三層神經網路比兩層神經網路正確值高，表現得比較好

Test_loss 方面：然而測試資料集我設計三層的神經網路比兩層神經網路下降要快並且值更小

測試正確值方面：我設計三層的神經網路有比兩層神經網路的正確值高一點點，很快就學到不錯的正確率。

增加一層隱藏層能夠提高神經網路的訓練準確率是很有趣的現象，但似乎並不能提高太多測試準確率。這可能是由於過度擬合的原因，也就是模型過度貼近訓練數據，而無法很好地推廣到未見過的測試數據。

為了解決這個問題，我們可以嘗試調整超參數，例如每個隱藏層中神經元的數量、學習速率，或者是正則化技術，例如 dropout 或權重衰減。重要的是要在模型複雜度和泛化性能之間取得平衡，而找到最佳的超參數通常需要一些試驗和錯誤。

總的來說，結果表明，三層神經網路可能具有更好的表示能力，能

夠捕捉訓練數據中的複雜模式，但我們需要謹慎處理過度擬合的問題，並探索提高測試數據泛化性能的方法。