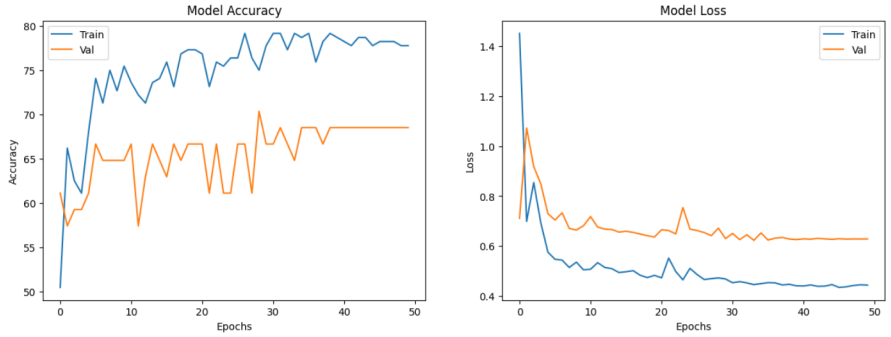
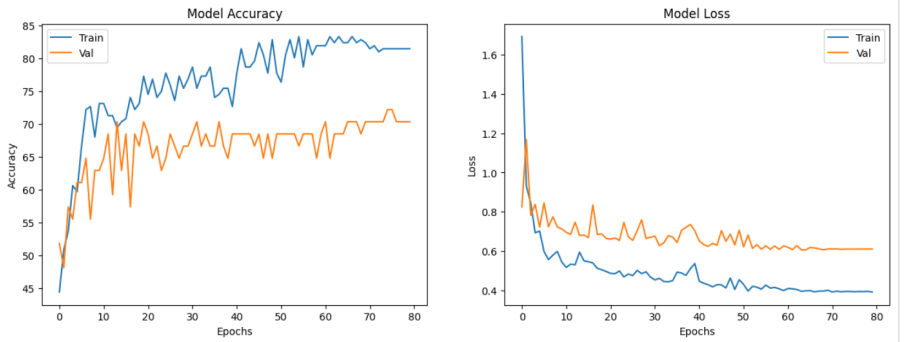
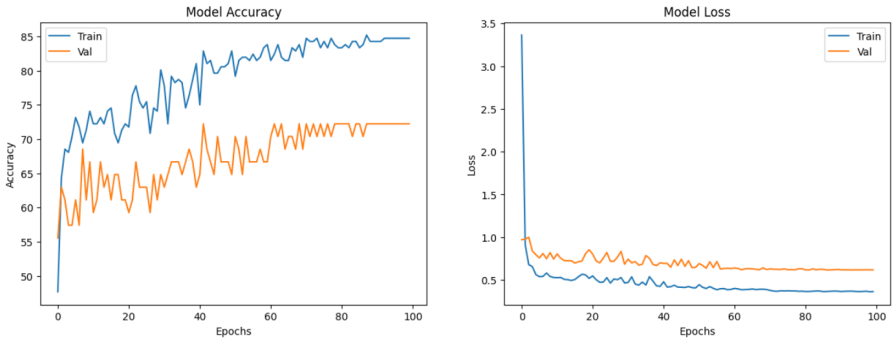
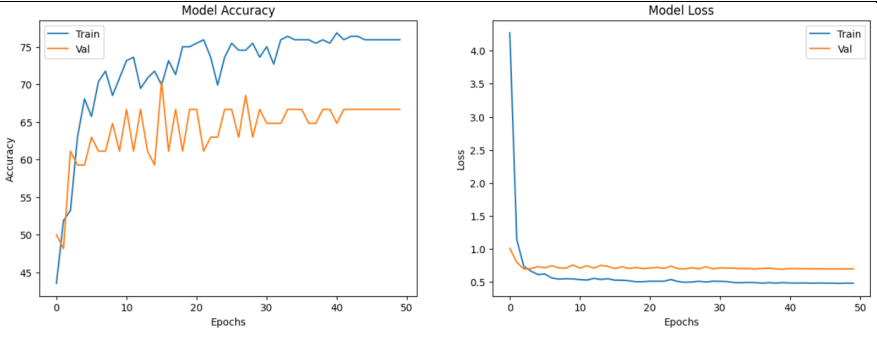
Homework 2

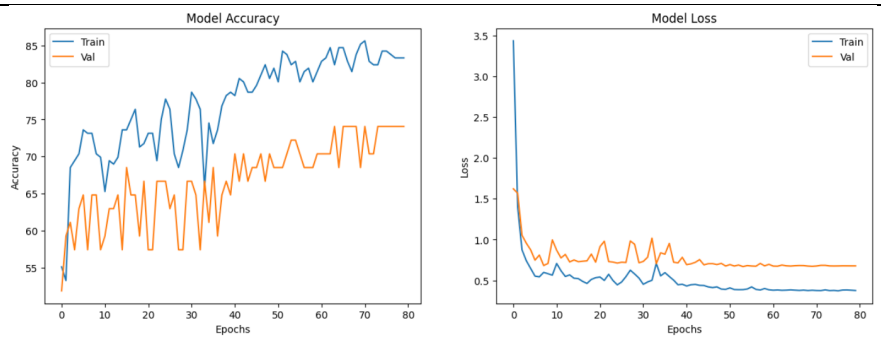
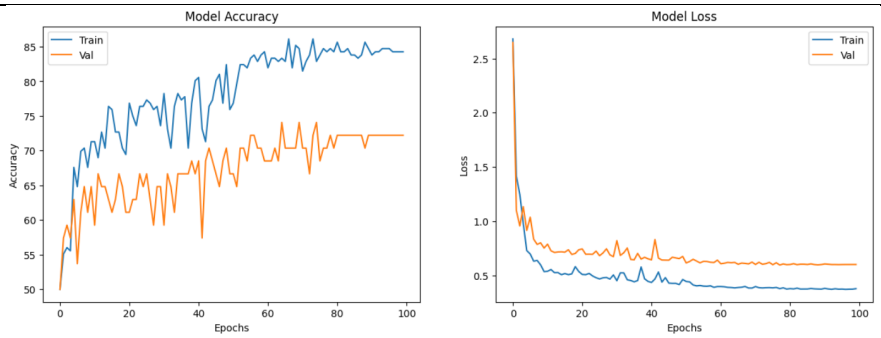
Name: 盧育琦　　Student ID: 112034582

1. 我改的是「B. Defineing Neural Networks」中以下兩個參數：

```python
[21] import torch.nn as nn

     class Model(nn.Module):
         def __init__(self):
             super().__init__()
             self.model = nn.Sequential(
                 nn.Linear(13, 100),
                 nn.ReLU(),
                 nn.Linear(100, 50),
                 nn.ReLU(),
                 nn.Linear(50, 2)
             ).cuda()

         def forward(self, x):
             return self.model(x)
```

```python
[22] import torch.optim as optim
     from torch.optim.lr_scheduler import CosineAnnealingLR, StepLR
     from tqdm.auto import tqdm

     train_losses = []
     val_losses = []
     train_accuracies = []
     val_accuracies = []

     epochs = 50

     model = Model()
     # print(model)

     best_val_loss = float('inf')
     best_val_acc = -1
```

nn.Linear(13, 100), nn.ReLU(), nn.Linear(100, 50), nn.ReLU(), nn.Linear(50, 2) epochs=50	
nn.Linear(13, 100), nn.ReLU(), nn.Linear(100, 50), nn.ReLU(), nn.Linear(50, 2) epochs=80	
nn.Linear(13, 100), nn.ReLU(), nn.Linear(100, 50), nn.ReLU(), nn.Linear(50, 2) epochs=100	
nn.Linear(13, 256), nn.ReLU(), nn.Linear(256, 50), nn.ReLU(), nn.Linear(50, 2) epochs=50	

nn.Linear(13, 256), nn.ReLU(), nn.Linear(256, 50), nn.ReLU(), nn.Linear(50, 2) epochs=80	
nn.Linear(13, 256), nn.ReLU(), nn.Linear(256, 50), nn.ReLU(), nn.Linear(50, 2) epochs=100	

2. 我發現 epochs 比較小的時候，Model Accuracy 也會比較小，Model Loss 也會比較大，但當訓練次數越來越多的時候，會發現 Model Loss 的曲線變得比較平穩，雖然剛開始的動盪比較大（下面三張圖）。而 nn.Linear 層的輸出維度從 100 增加至 256 使模型更好地擬合，Model Accuracy 也變得比較大。

3. 造成訓練集和測試集之間的準確度差異可能由以下原因產生：
   I. Overfitting：模型在訓練集上表現良好，但在測試集上表現不佳，因為它無法泛化到未見過的數據。
   II. Underfitting：模型在訓練集和測試集上的表現都不理想，通常是因為模型過於簡單，無法捕捉數據中的複雜模式。這導致了訓練集和測試集上的準確性都較低，並且準確性差距不大。

4. 在機器學習模型的表格資料集中，選擇的相關特徵會直接影響模型的性能和泛化能力。適當地選擇特徵可以幫助模型更好地捕捉數據、減少過擬合。
   I. 過濾法：通常基於統計量（如相關性、信息增益等）來評估每個特徵與目標變量之間的關係，可以快速篩選出關聯性較低的特徵，從而降低了模型的維度，減少計算成本。
   II. 包裝法：通常基於特徵子集的性能來選擇特徵，根據性能指標（如準確性、交叉驗證分數等）來選擇最佳的特徵子集。雖然可能更耗時，但通常會有更好的結果，因為它考慮了特徵之間的相互作用。
   III. 嵌入法：將特徵選擇嵌入到模型訓練過程中，通常模型具有內置選擇機制，例如決策樹、隨機森林等。這些模型可以通過計算每個特徵的重要性來自動選擇最佳特徵子集。然而有時可能會忽略特徵之間的相互作用。

   （資料來源）Chatgpt 參考 Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), 1157-1182.

5. 在處理表格資料時，除了人工神經網路（ANN）之外，還有 TabTransformer。它是一種針對表格數據設計的注意力機制模型，旨在處理具有不規則列之間關係的表格數據。它引入了多頭自注意力機制，使模型能夠在不同層次和範圍上捕捉特徵之間的關係。此外，TabTransformer 還包括特徵檢測器和列關係編碼器，有助於有效地處理不同類型的特徵和列之間的相互作用。這使得它能夠在處理具有高度結構化和多樣化特徵的表格數據時表現出色，並提供了更好的解釋性和泛化能力。

（資料來源）Chatgpt 參考 Chen, W., Li, L., Jia, R., Yang, Q., Liu, Y., Zhang, T., & Gao, H. (2022). TabTransformer: Tabular Data Transformer for Machine Learning. arXiv preprint arXiv:2202.07945.