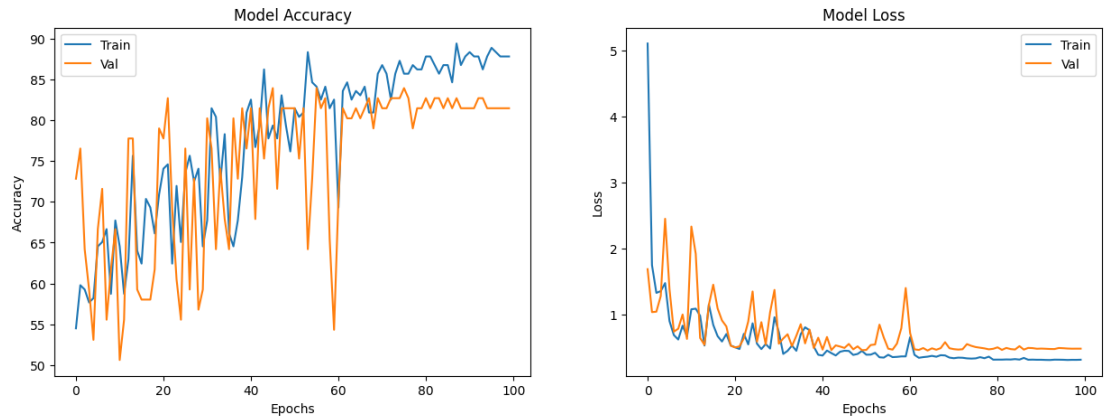


## Homework 2

112034529 游慧榆

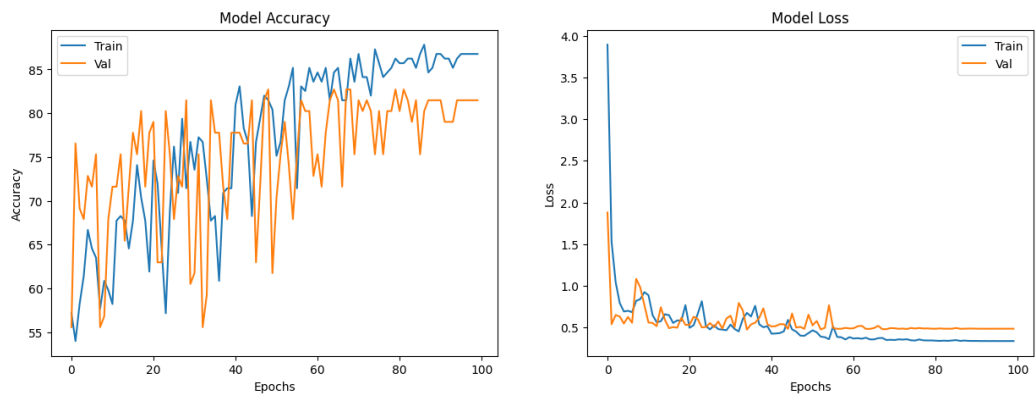
### 1. 更改的超參數為 Batch size 和學習率

(1) batch size=16, 學習率=1e-3



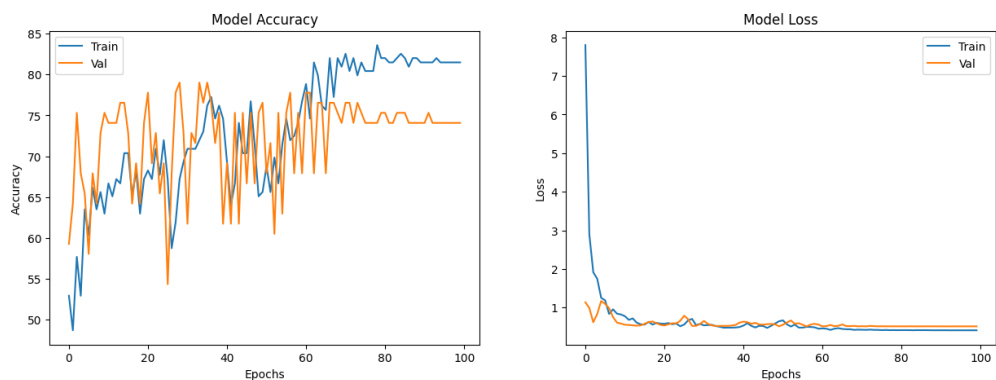
Test accuracy is 74.19354838709677%, 模型準確度初期震盪較大，但後期逐漸平穩。

(2) batch size=32, 學習率=1e-3



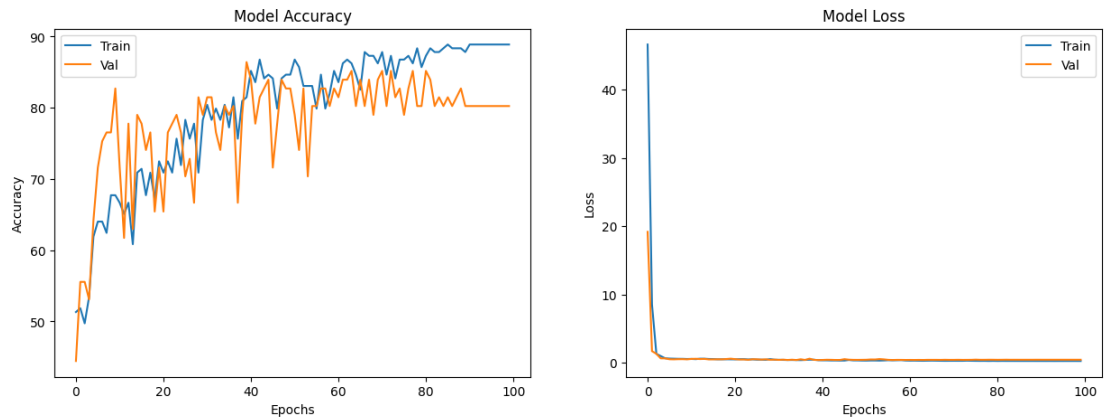
Test accuracy is 70.96774193548387%, batch size 增加後震盪幅度前期稍微減小，但後期準確度波動反而比較大。

(3) batch size=64, 學習率=1e-3



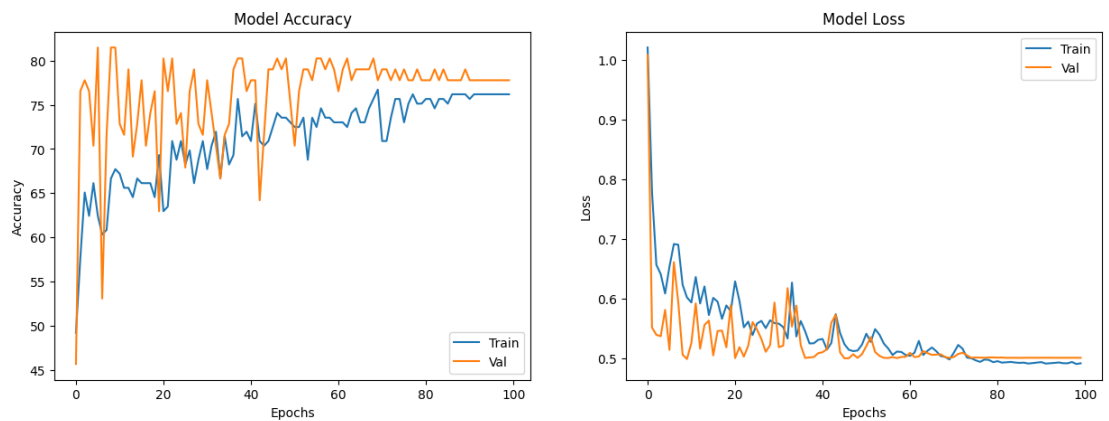
Test accuracy is 61.29032258064516%，batch size 調更大後模型損失趨於平緩，可能是因為資料集批次大學習會比小批次更穩定一點。

(4) batch size=32, 學習率=1e-2



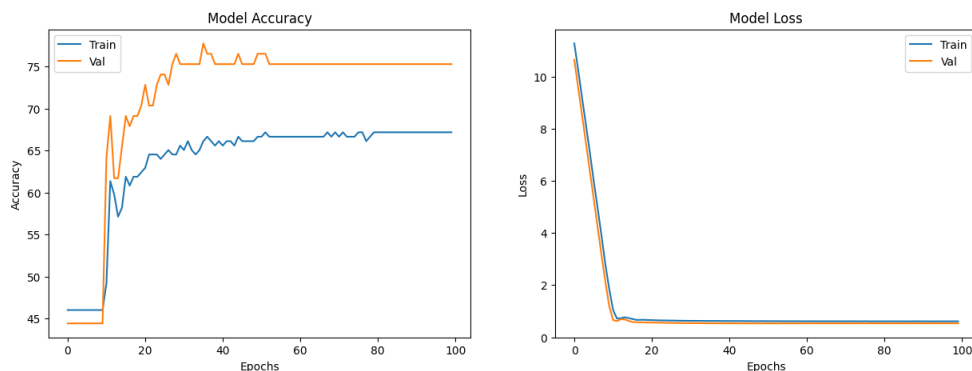
Test accuracy is 67.74193548387096%，學習率增加後模型損失後期幾乎沒有變化，驗證集準確度和訓練集準確度後期明顯分開，表示模型可能步伐過大。

(5) batch size=32, 學習率=1e-4



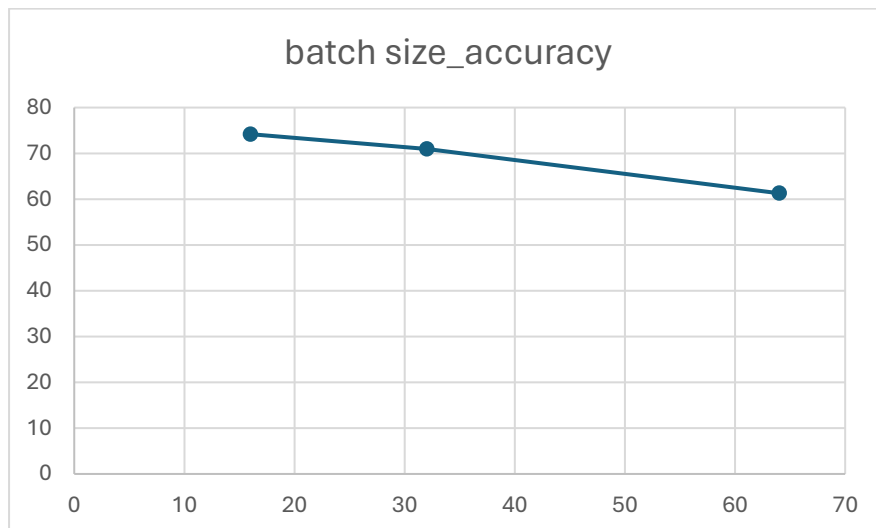
Test accuracy is 61.29032258064516%。學習率降低後模型波動比較明顯，但訓練集和驗證集差別也不小，而且驗證集準確度反而高於訓練集，但測試集比前一次準確度差，可能表示測試集與驗證集分佈不同。

(6) batch size=32, 學習率=1e-5



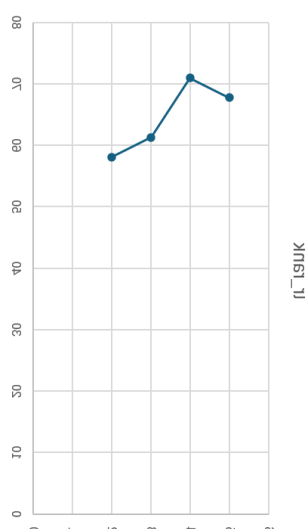
Test accuracy is 58.064516129032256%。學習率太小可能導致模型更新太慢，難以在有限的 epoch 次數內學習到好的參數。

## 2. 結果分析



Batch size 分別嘗試了 16、32、64 三種組合，我將以測試集的準確率來評估三種超參數變化會怎麼影響模型的準確度。Batch size 的比較是建立在模型的結構與其他參數都一樣的情況來進行比較。

結果如上圖所示，縱軸為測試集準確率，橫軸由左到右代表 batch size 分別為 16、32、64 的結果，顯示隨著 batch size 增加準確度是下降的，這可能表示此資料集需要較小的批次來多次調整模型才能找到比較好的參數，得到比較好的準確度。



第二個調整的超參數是學習率，我嘗試了  $1e-2$ 、 $1e-3$ 、 $1e-4$ 、 $1e-5$  的四種組合，模型的其他參數同樣保持一致。由於這四個數字相差的範圍巨大，我改用 rank 重新編碼來繪圖。

圖表的縱軸同樣表示準確率，橫軸的 2、3、4、5 分別代表學習率為  $1e-5 \sim 1e-2$ 。如圖中所示，學習率從  $1e-5$  往上增加到  $1e-3$  時，隨著學習率增加測試集的準確率是上升的，但如果再增加到  $1e-2$  時準確率又往下，可見得對於此資料集來說， $1e-3$  是比較剛好的學習率，學習率太小的話難以學到資料的特性，但若增加到  $1e-2$  可能又學得太快、容易錯過正確的學習方向。

結論；整合這 6 個超參數組合的實驗結果，我們發現較適合的 batch size 設定為 16、學習率為  $1e-3$ 。

### 3. 分析為什麼訓練集準確度和測試集準確度會有差異。

關於訓練集準確度和測試集準確度有差異的原因，我覺得主要有兩個原因。首先，最有可能的是模型還不夠好，由於前面的超參數實驗僅調整兩個參數，這次的資料也幾乎沒有做前處理，所以應該還有更泛化的模型能同時提高訓練集和測試集的準確度。其次可能是訓練集、驗證集、測試集沒有適當的分割，導致這三者的分佈不同，準確度也會有所差異。理想的情況應該是這三者分佈都相同，訓練出來的準確度才會一致。

### 4. 特徵選擇的方法

關於特徵選擇的方法，首先，我認為可以從專家或是一般常識來判斷是否要選取該特徵。例如根據一般人的邏輯判斷，在訓練一個預測房價模型時，會很直觀的判斷房間大小、地段可能是影響房價的關鍵因素，在特徵選擇時就可以優先納入考量。此外根據數據分析師的分享，特徵選擇主要分成過濾法（依照特定指標如相關性選擇）、包裝法（依照目標函數結果逐次嘗試）、嵌入法（先用其他機器學習技術訓練後再篩選特徵）三種。

<https://medium.com/ai%E5%8F%8D%E6%96%97%E5%9F%8E/%E7%89%B9%E5%BE%B5%E5%B7%A5%E7%A8%8B%E4%B9%8B%E7%89%B9%E5%BE%B5%E9%81%B8%E6%93%87%E6%A6%82%E5%BF%B5-call1745db63c>

### 5. 除了 ANN 以外適用於表格數據的深度學習方法

Google 的 TabNet 適用是一種適用於表格數據的深度學習方法，利用注意力機制選擇性關注輸入特徵的不同部分來提高模型的解釋能力。另一種適合用於複雜表格數據的深度學習方法是 SAINT（Self-Attention and Interspersed Transformer），是特別為表格數據建立的 transformer 架構，運用自注意力機制來處理表格中的行和列，透過交錯插入自注意力機制和前饋神經網路層的方式以在處理特徵的時候保持高度的靈活和適應性。