1.

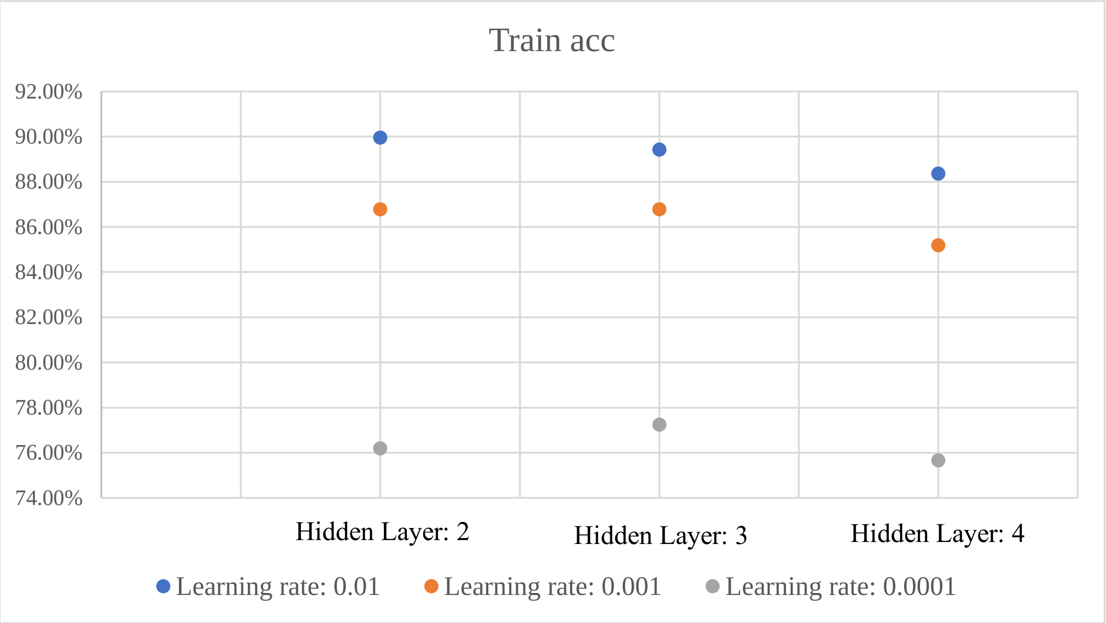
Hyper-parameters	nn.Sequentail:
	`self.model = nn.Sequential(`
	`    nn.Linear(13, 256),`
	`    nn.ReLU(),`
	`    nn.Linear(256, 256),`
	`    nn.ReLU(),`
	`    nn.Linear(256, 2)`
	`).cuda()`
		nn.Sequentail:
		`self.model = nn.Sequential(`
		`    nn.Linear(13, 256),`
		`    nn.ReLU(),`
		`    nn.Linear(256, 256),`
		`    nn.ReLU(),`
		`    nn.Linear(256, 256),`
		`    nn.ReLU(),`
		`    nn.Linear(256, 2)`
		`).cuda()`
			nn.Sequentail:
			`self.model = nn.Sequential(`
			`    nn.Linear(13, 256),`
			`    nn.ReLU(),`
			`    nn.Linear(256, 256),`
			`    nn.ReLU(),`
			`    nn.Linear(256, 256),`
			`    nn.ReLU(),`
			`    nn.Linear(256, 256),`
			`    nn.ReLU(),`
			`    nn.Linear(256, 2)`
			`).cuda()`

Learning rate:0.01	Train acc: 89.9471% Train loss: 0.2731 Val acc: 79.0123% Val loss: 0.4941 Test acc: 70.9677% Test loss: 0.4597	Train acc: 89.4180% Train loss: 0.2746 Val acc: 77.7778% Val loss: 0.5755 Test acc: 80.6452% Test loss: 0.4125	Train acc: 88.3598% Train loss: 0.2774 Val acc: 79.0123% Val loss: 2.4321 Test acc: 67.7419% Test loss: 0.4612
Learning rate:0.001	Train acc: 86.7725% Train loss: 0.3484 Val acc: 79.0123% Val loss: 0.4613 Test acc: 70.9677% Test loss: 0.5350	Train acc: 86.7725% Train loss: 0.3672 Val acc: 81.4815% Val loss: 0.5360 Test acc: 70.9677% Test loss: 0.5383	Train acc: 85.1852% Train loss: 0.3472 Val acc: 75.3086% Val loss: 0.6932 Test acc: 74.1935% Test loss: 0.5873
Learning rate:0.0001	Train acc: 76.1905% Train loss: 0.4826 Val acc: 67.9012% Val loss: 0.5888 Test acc: 58.0645% Test loss: 0.6807	Train acc: 77.2487% Train loss: 0.4865 Val acc: 67.9012% Val loss: 0.5851 Test acc: 67.7419% Test loss: 0.6647	Train acc: 75.6614% Train loss: 0.5047 Val acc: 66.6667% Val loss: 0.5988 Test acc: 64.5161% Test loss: 0.6679

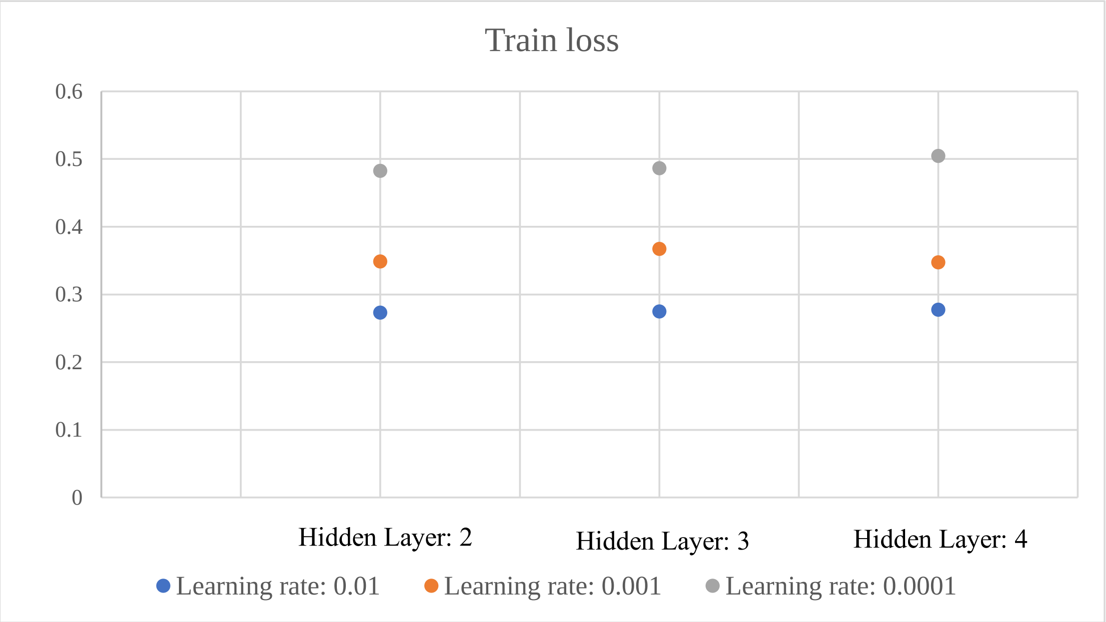
2.

根據以上實驗結果，隱藏層數為三層時相較於兩層及四層結果最佳，越多層可能會使模型表現更加，但也可能導致 Overfitting 的情況發生；Learning rate 則是 0.01 時較 0.001、0.0001 時佳，對於簡單的數據 Learning rate 越大模型表現結果更好。以此數據集來說，隱藏層數為三層且 Learning rate 為 0.01 時模型表現結果最佳。

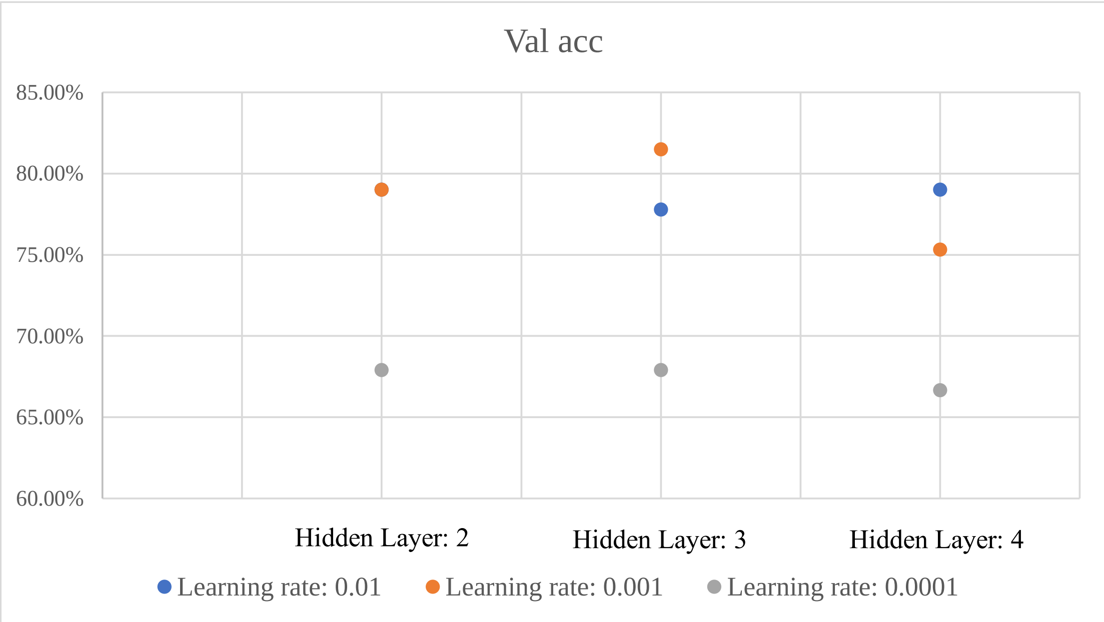
Train acc:



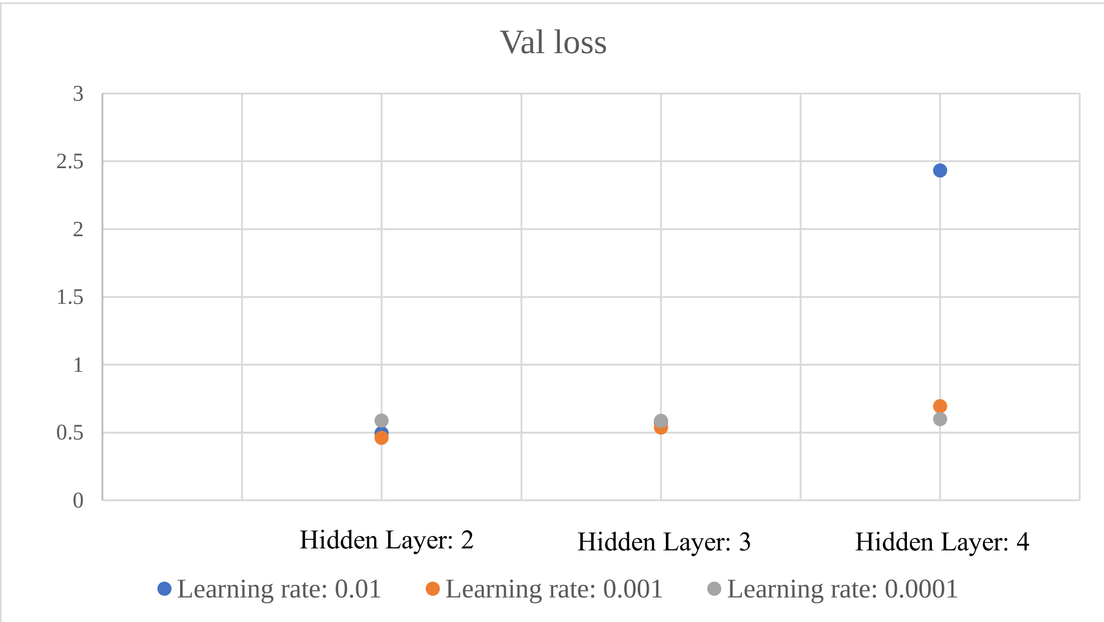
Train loss:



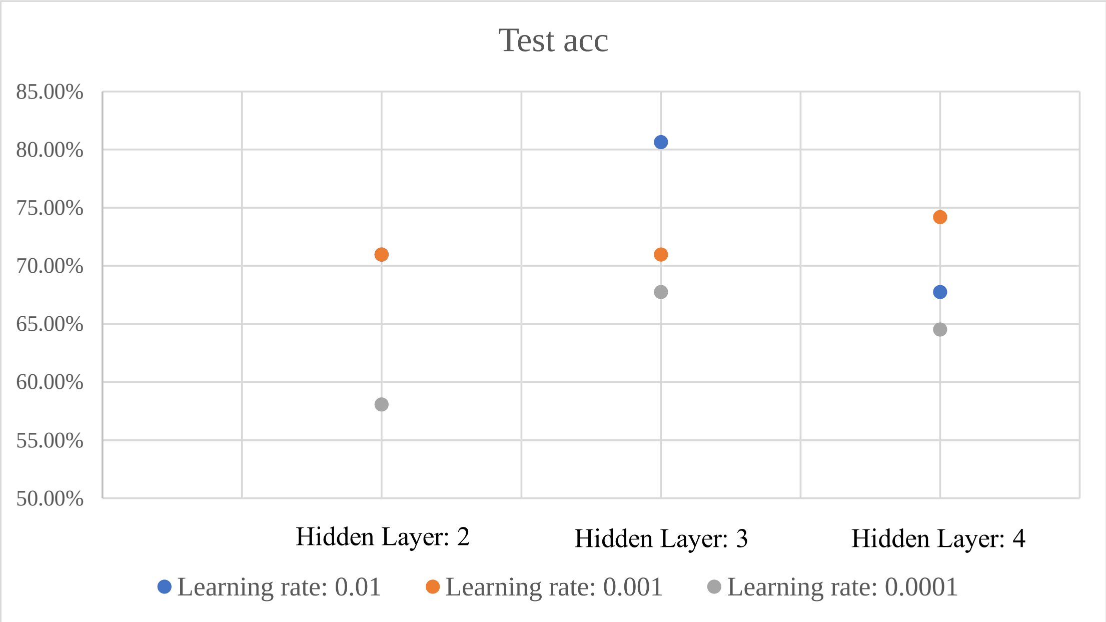
Val acc:



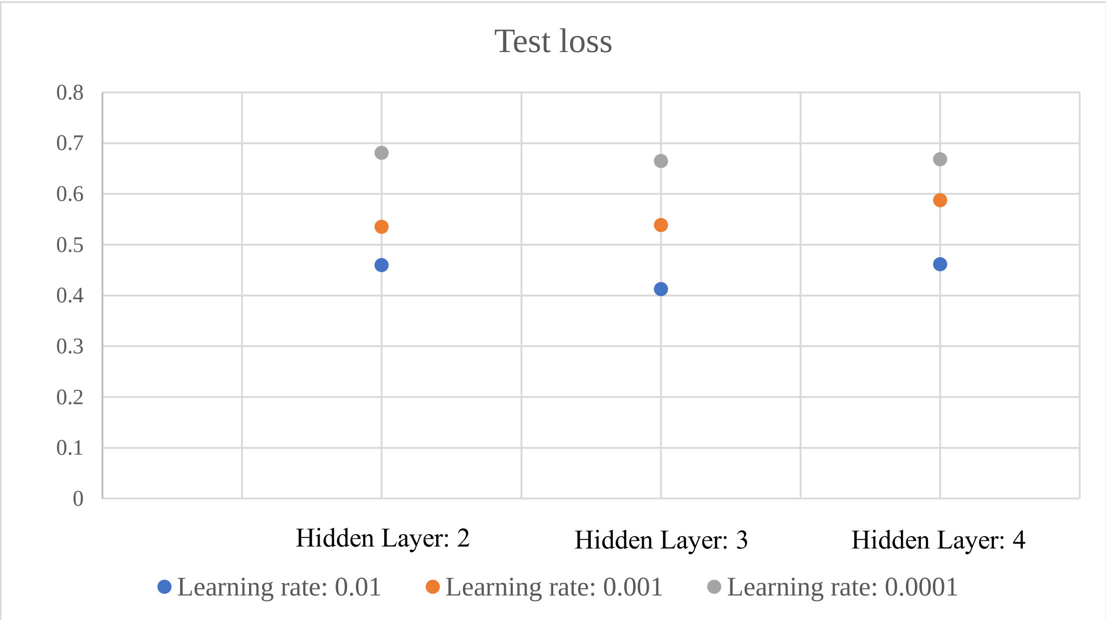
Val loss:



Test acc:



Test loss:



3.

Training acc、Test acc 不同的原因可能為 Testing Data 太少只有 31 筆，預測正確差 1 筆資料就會讓 accuracy 差大約 3%，且在調整不同參數時模型會學習太多或太少導致 Overfitting、Underfitting 的情況發生，讓 Test acc 較低，也有可能是 Training Data、Test Data 的資料分佈不同，讓模型預測準確率下降。

4.

特徵選擇的方法包含過濾法、包裝法、嵌入法
過濾法：根據特徵的發散性或相關性進行評分，並設置一個閾值或選擇待選特徵的數量，從而篩選出重要特徵
包裝法：根據目標函數（通常是預測效果的評分），每次根據特徵對模型性能的貢獻，選擇一些特徵進行保留或排除。
嵌入法：先訓練一個機器學習模型，通過該模型計算各個特徵的權重或重要性，然後根據這些權重從大到小進行特徵選擇。

https://zhuanlan.zhihu.com/p/74198735

5.

TabNet 更適合處理 tabular datasets，TabNet 是一種端到端的神經網絡，專為直接處理原始表格數據而設計，並利用注意力機制來選擇性地解釋每個特徵，不需要手動特徵工程，可以直接輸入原始數據並透過梯度下降（Gradient Descent）進行最佳化，也能夠提供 Local Interpretability 與 Global Interpretability，讓使用者可以理解模型如何做出決策，在許多 tabular datasets 的基準測試中（e.g., XGBoost、LightGBM 等），TabNet 的表現可相仿或超越這些傳統樹模型，在大數據場景下更有優勢。

https://medium.com/@doris2913/%E8%AB%96%E6%96%87-tabnet-attentive-interpretable-tabular-learning-52abeb9d7a

TabNet: Attentive Interpretable Tabular Learning - SÖ Arik, T Pfister