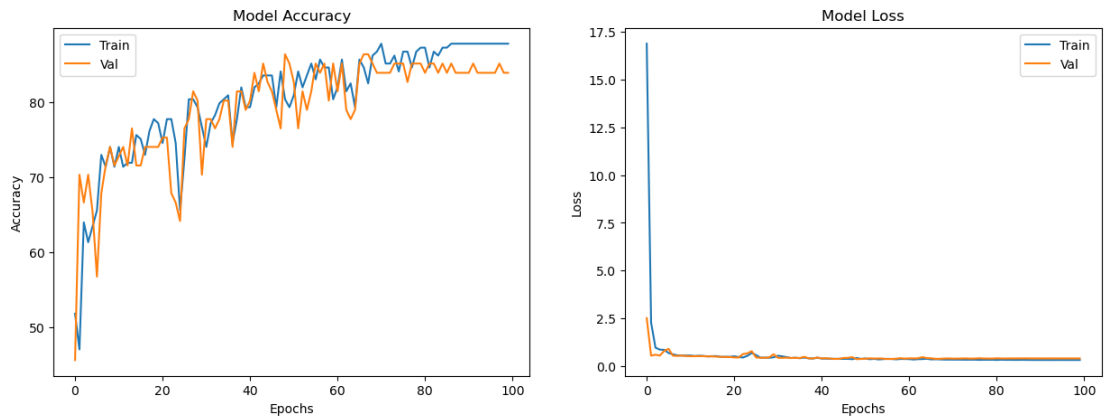


1. Learning rate: (10^{-2} , 10^{-3} , 10^{-4}) 、second layer's nodes: (128, 256, 512)

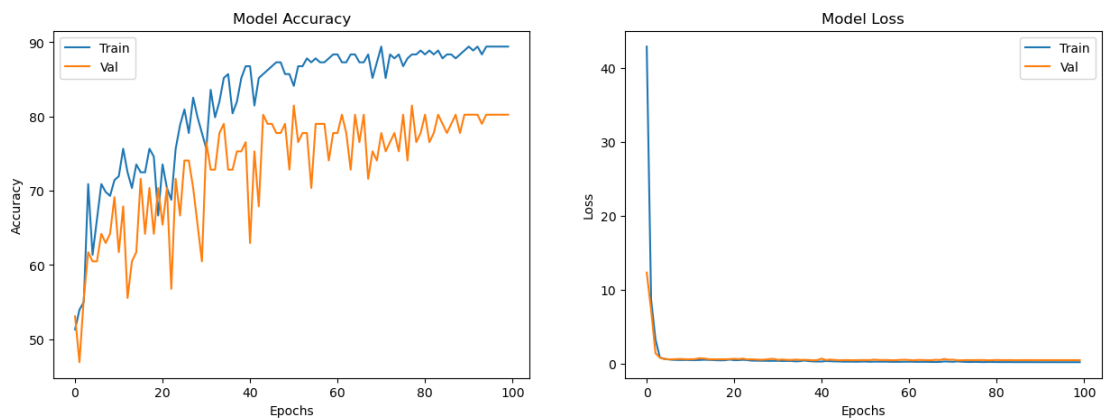
Lr: 10^{-2} , nodes: 128:



Best acc, loss

	train	val	test
Acc	87.83	86.41	77.42
loss	0.312	0.358	0.476

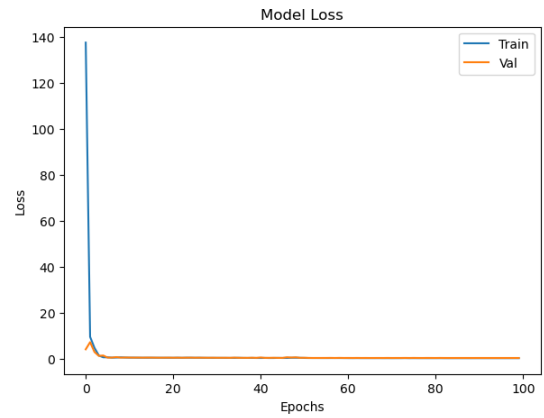
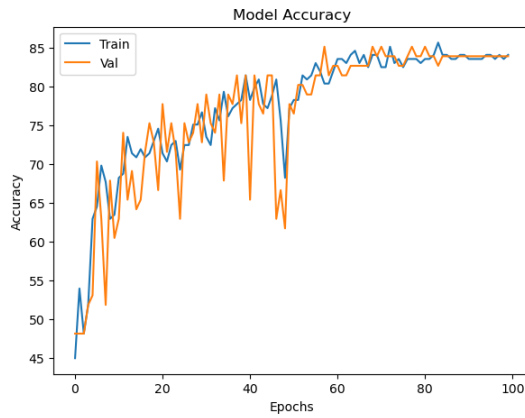
Lr: 10^{-2} , nodes: 256:



Best acc, loss

	train	val	test
Acc	89.41	81.48	77.42
loss	0.242	0.492	0.453

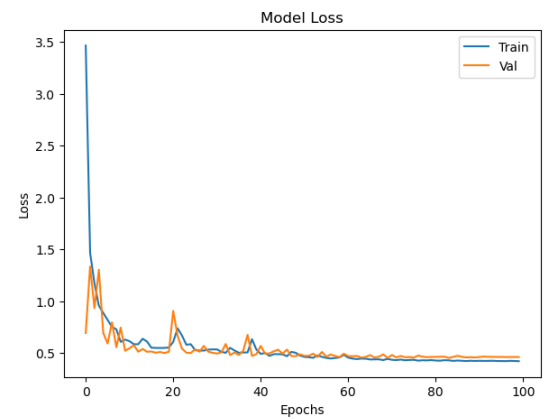
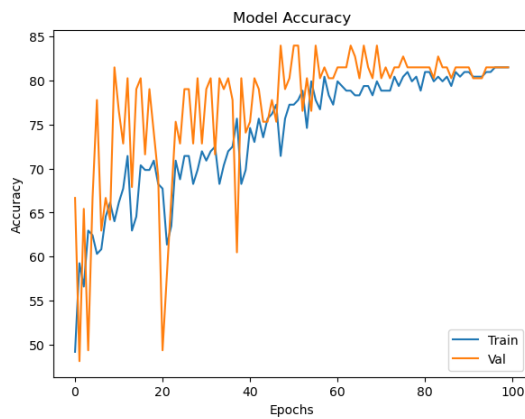
Lr: 10^{-2} , nodes: 512:



Best acc, loss

	train	val	test
Acc	85.71	85.18	67.74
loss	0.344	0.426	0.556

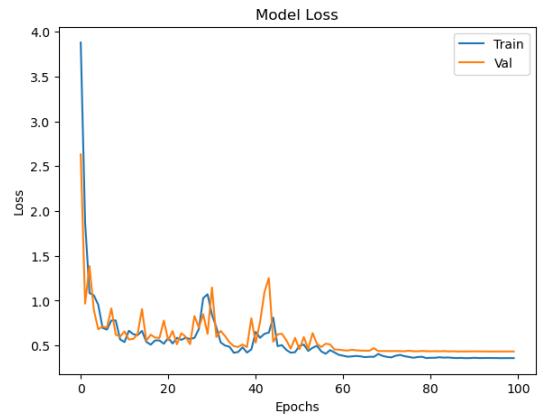
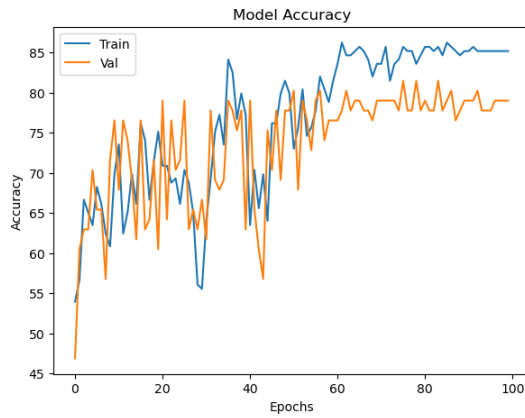
Lr: 10^{-3} , nodes: 128:



Best acc, loss

	train	val	test
Acc	81.48	83.95	64.52
loss	0.421	0.452	0.564

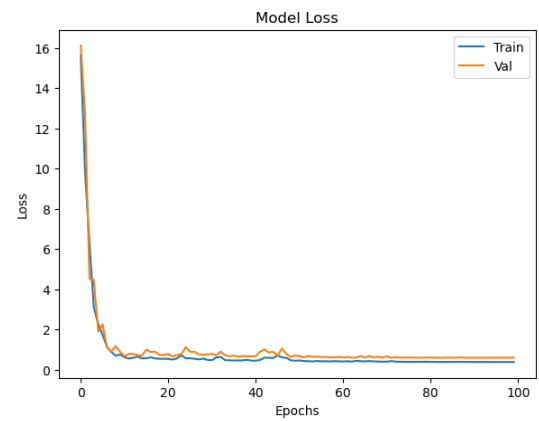
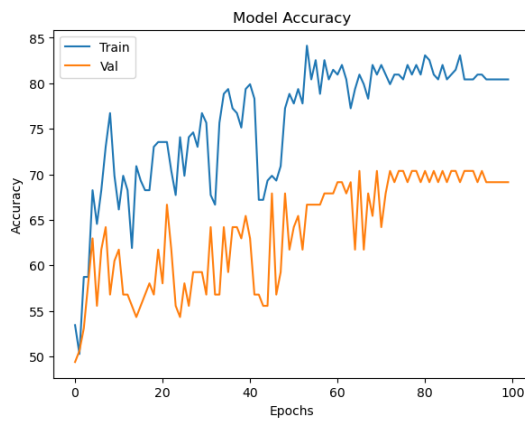
Lr: 10^{-3} , nodes: 256:



Best acc, loss

	train	val	test
Acc	86.24	81.48	77.42
loss	0.358	0.432	0.454

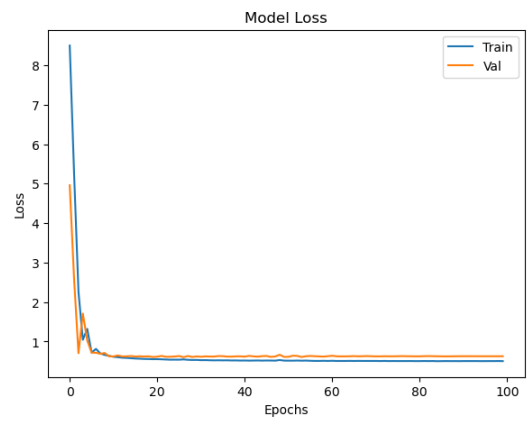
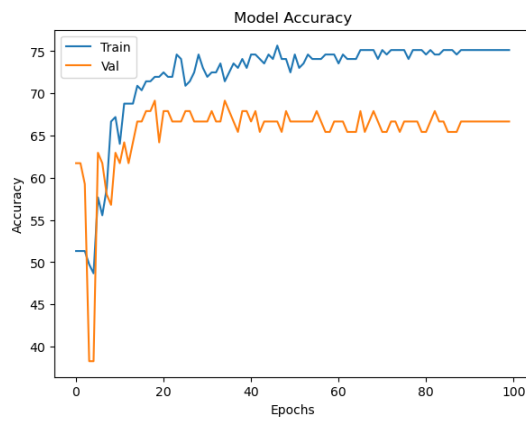
Lr: 10^{-3} , nodes: 512:



Best acc, loss

	train	val	test
Acc	84.12	70.37	58.06
loss	0.381	0.591	0.631

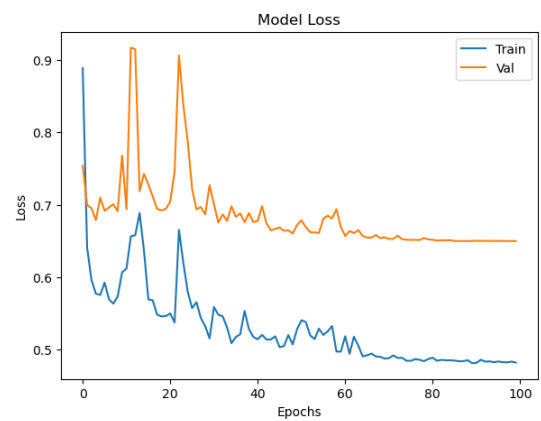
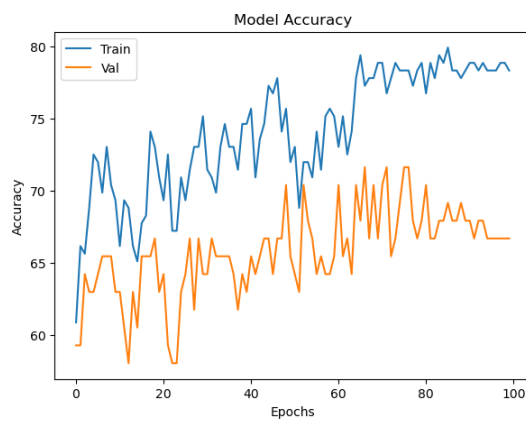
Lr: 10^{-4} , nodes: 128:



Best acc, loss

	train	val	test
Acc	75.66	69.14	67.74
loss	0.503	0.604	0.643

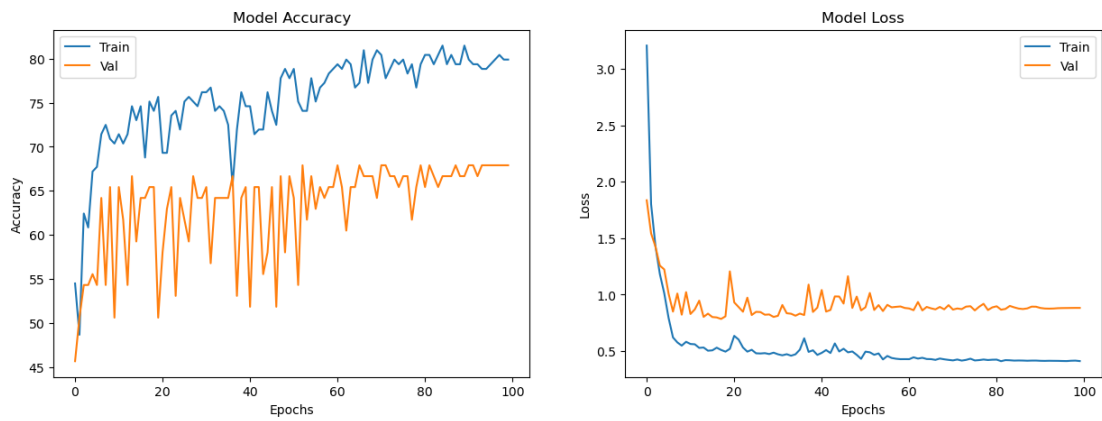
Lr: 10^{-4} , nodes: 256



Best acc, loss

	train	val	test
Acc	79.89,	71.60	67.74
loss	0.482	0.650	0.657

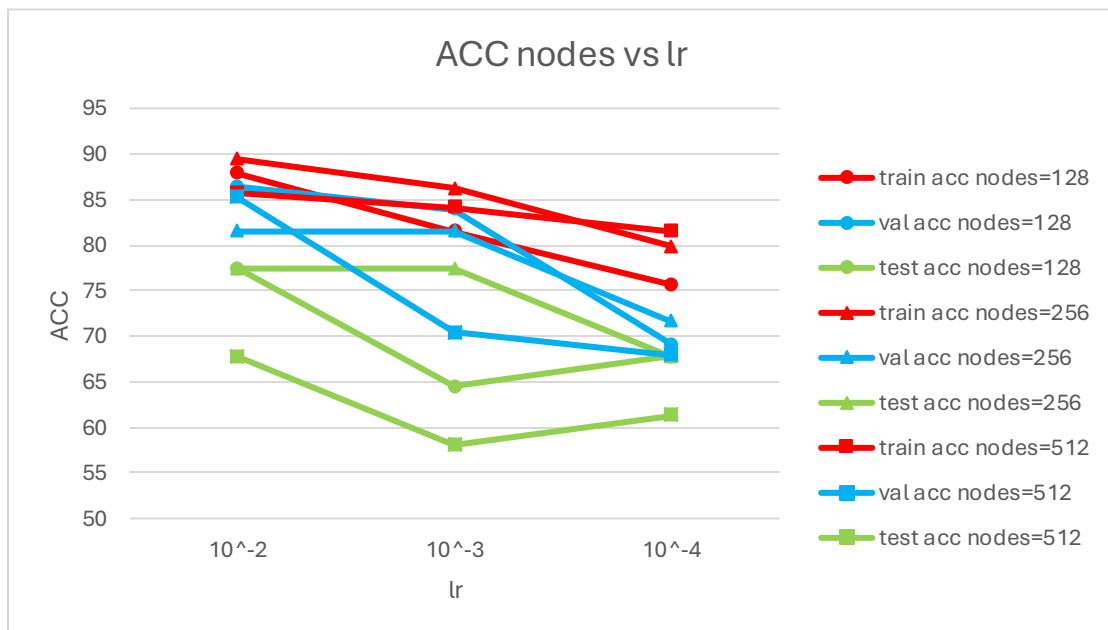
Lr: 10^{-4} , nodes: 512

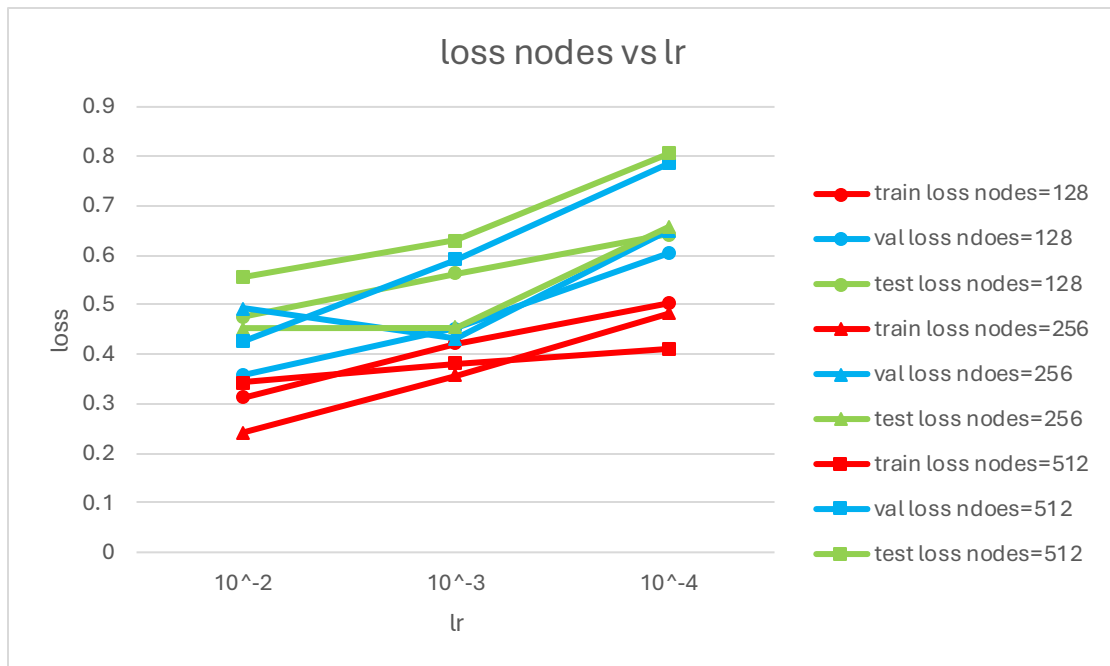


Best acc, loss

	train	val	test
Acc	81.48	67.90	61.29
loss	0.411	0.785	0.805

2.





在其他超參數以及模型架構不變的情況下，nodes 相同但 learning rate 下降在大多數的 case 下 ACC 都下降，且 loss 都提升，而在相同的 learning rate 條件下，node 數量等於 256 無論是 acc 還是 loss 的表現都最佳，接著表現是 node 數 128 再來是 512。因此最佳的組合是 learning rate = 10^{-2} , nodes = 256。

3. 在所有的 case 中 train 集的表現都比 test 集還要好，無論表現指標是 acc 還是 loss。有幾點可能原因。第一、模型 overfitting，模型過度注重在 train 數據集的表現，導致沒有學到真正的規律。第二、資料分佈不一致，測試集的樣本分佈跟訓練集有差別，例如：train 資料是正常狀況，但 test 資料包含極端案例。第三、資料量太少，訓練 loss 和 acc 都非常快下降，但測試集無法受益。
4. 特徵選擇是特徵工程中的關鍵步驟，旨在從大量特徵中挑選出最具代表性的子集，以提升模型的準確性和運行效率。這一過程有助於剔除不相關或冗餘的特徵，從而減少特徵數量，提高模型精度，並縮短運行時間。常見的特徵選擇方法包括過濾法 (Filter)、包裝法 (Wrapper) 和嵌入法 (Embedded)。過濾法根據特徵的發散性或與目標變量的相關性進行評分；包裝法則根據預測效果評分來選擇特徵；嵌入法則在模型訓練過程中同時進行特徵選擇。Ref:

<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-ca11745db63c>

5. TabNet 是一種專為處理表格數據而設計的深度學習模型，結合了神經網絡

的表徵學習能力與決策樹的可解釋性。其核心架構包括特徵變換器（Feature Transformer）和注意力變換器（Attentive Transformer），以逐步選擇並處理重要特徵。這種設計使模型能夠針對每個樣本動態選擇特徵，提升模型的解釋性和性能。與傳統的梯度提升模型相比，TabNet 在處理表格數據時表現出色，並且能夠直接使用原始數據進行訓練，減少了對特徵工程的依賴。Ref: <https://medium.com/@doris2913/%E8%AB%96%E6%96%87-tabnet-attentive-interpretable-tabular-learning-52abeb9d7a>