Homework 2

113030511 簡若伃

1. (20 pts) Select 2 hyper-parameters of the artificial neural network used in Lab 2 and set 3 different values for each. Perform experiments to compare the effects of varying these hyper-parameters on the loss and accuracy metrics across the training, validation, and test datasets. Present your findings with appropriate tables.

hidden_unit_options = [64, 128, 256]

learning_rate_options = [0.01, 0.001, 0.0001]

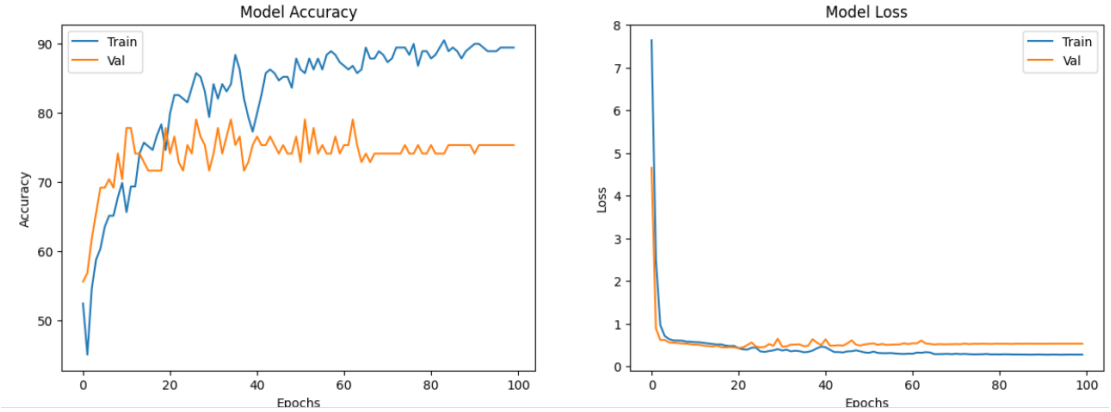
	hidden_units	learning_rate	train_loss	val_loss	test_loss	train_acc	val_acc	test_acc
0	64	0.0100	0.339175	0.564947	0.549461	85.714286	77.777778	70.967742
1	64	0.0010	0.412533	0.729248	0.683839	79.365079	71.604938	67.741935
2	64	0.0001	0.529077	0.671463	0.672859	74.603175	61.728395	64.516129
3	128	0.0100	0.319861	0.508123	0.525287	86.772487	82.716049	70.967742
4	128	0.0010	0.358284	0.788461	0.715884	85.714286	70.370370	70.967742
5	128	0.0001	0.515029	0.733691	0.690745	73.544974	64.197531	61.290323
6	256	0.0100	0.229676	0.717881	0.648972	89.947090	80.246914	74.193548
7	256	0.0010	0.336153	0.750539	0.634862	85.714286	72.839506	70.967742
8	256	0.0001	0.467068	0.771085	0.700017	76.719577	62.962963	64.516129

2. (20 pts) Based on your experiments in Question 1, analyze the outcomes. What differences do you observe with the changes in hyper-parameters? Discuss whether these adjustments contributed to improvements in model performance, you can use plots to support your points. (Approximately 100 words.)

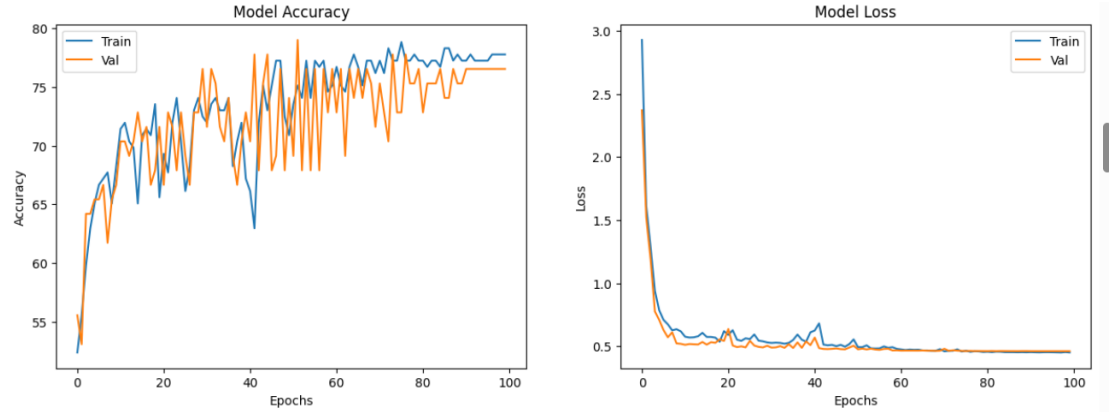
From the experiments, we observed that increasing the number of hidden units from 64 to 256 consistently improved train and test accuracy, especially when combined with a learning rate of 0.01. The best test accuracy 77.41% was achieved using 128 hidden units and a learning rate of 0.01. Meanwhile, using a learning rate of 0.001 or 0.0001 led to lower accuracy. Based on the experiments, both the learning rate and number of hidden units significantly affected the model's performance. Larger hidden units provided marginal improvements when combined with an appropriate learning

rate.

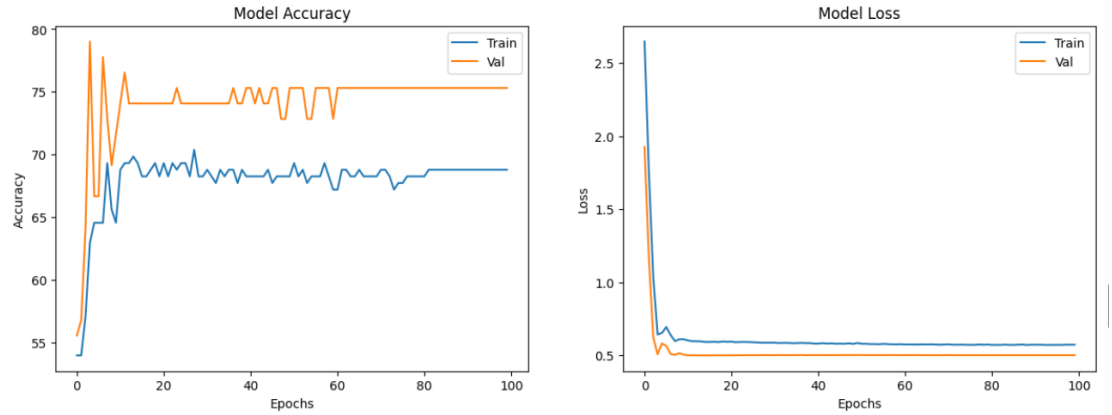
Training with hidden_units=64, learning_rate=0.01



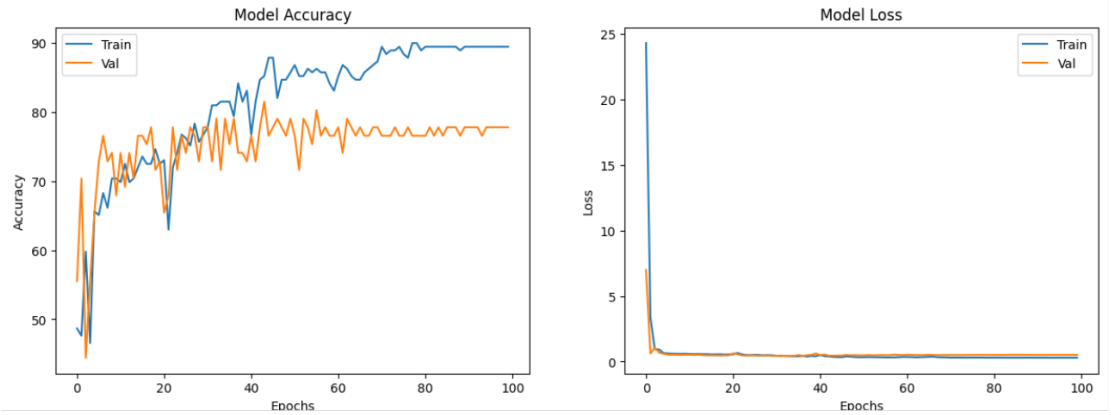
Training with hidden_units=64, learning_rate=0.001



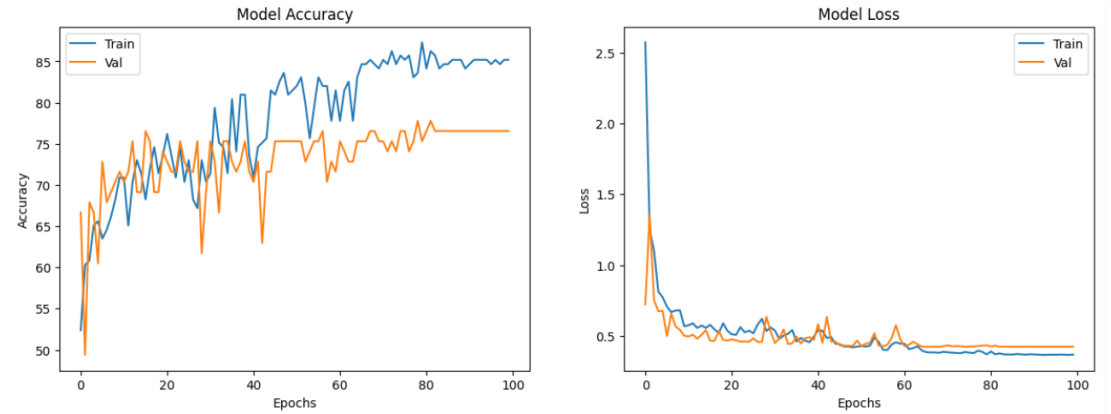
Training with hidden_units=64, learning_rate=0.0001



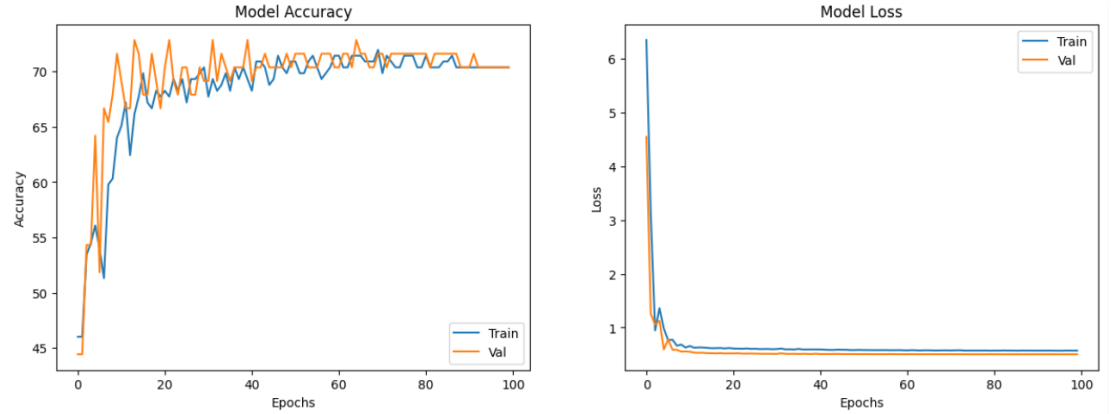
Training with hidden_units=128, learning_rate=0.01



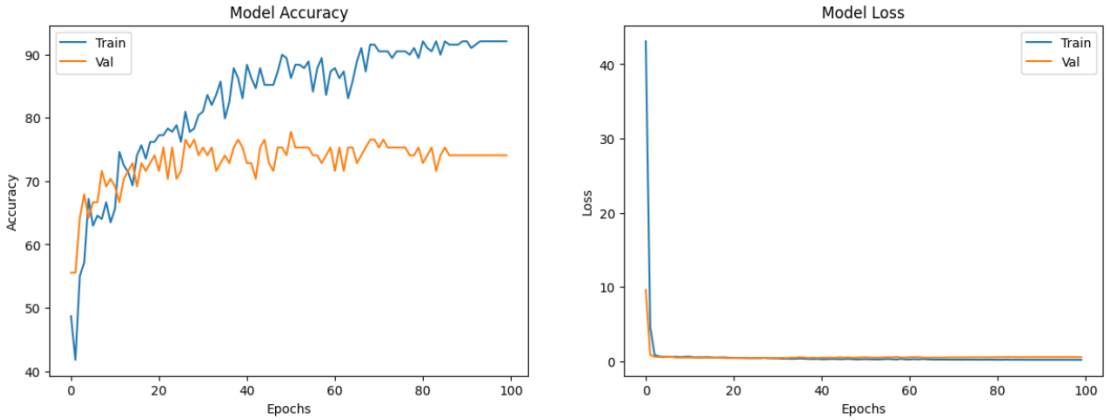
Training with hidden_units=128, learning_rate=0.001



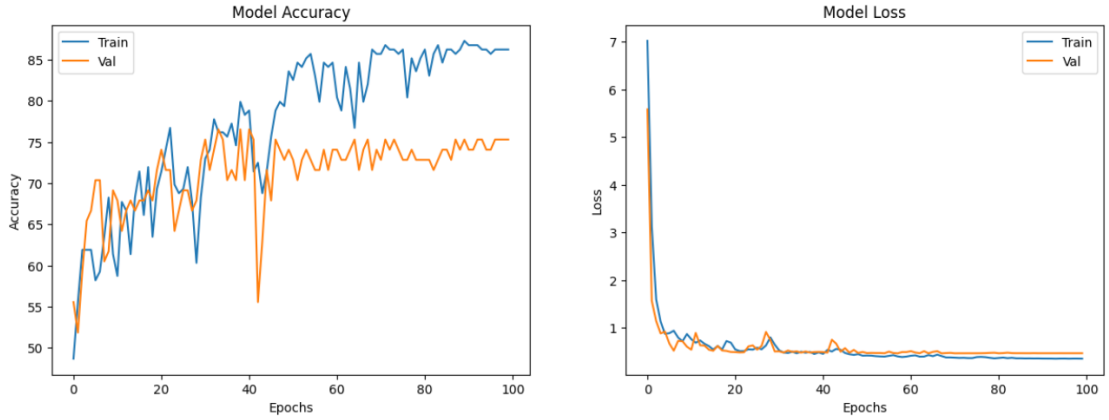
Training with hidden_units=128, learning_rate=0.0001



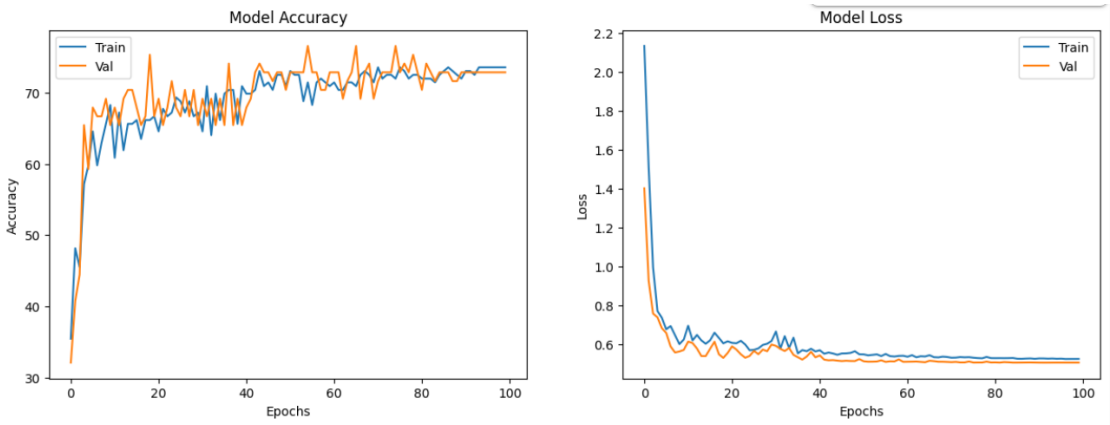
Training with hidden_units=256, learning_rate=0.01

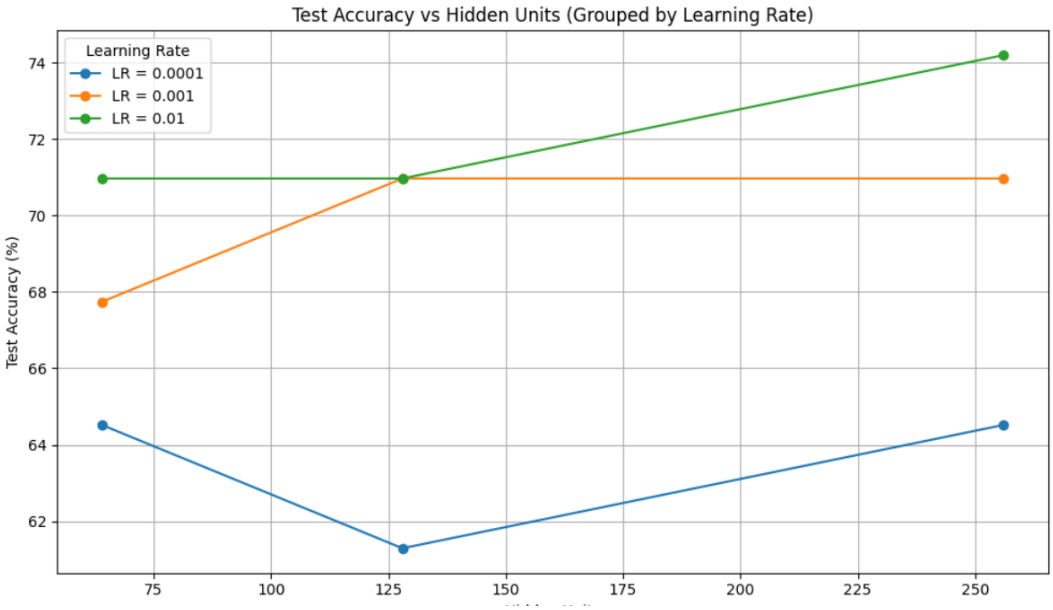


Training with hidden_units=256, learning_rate=0.001

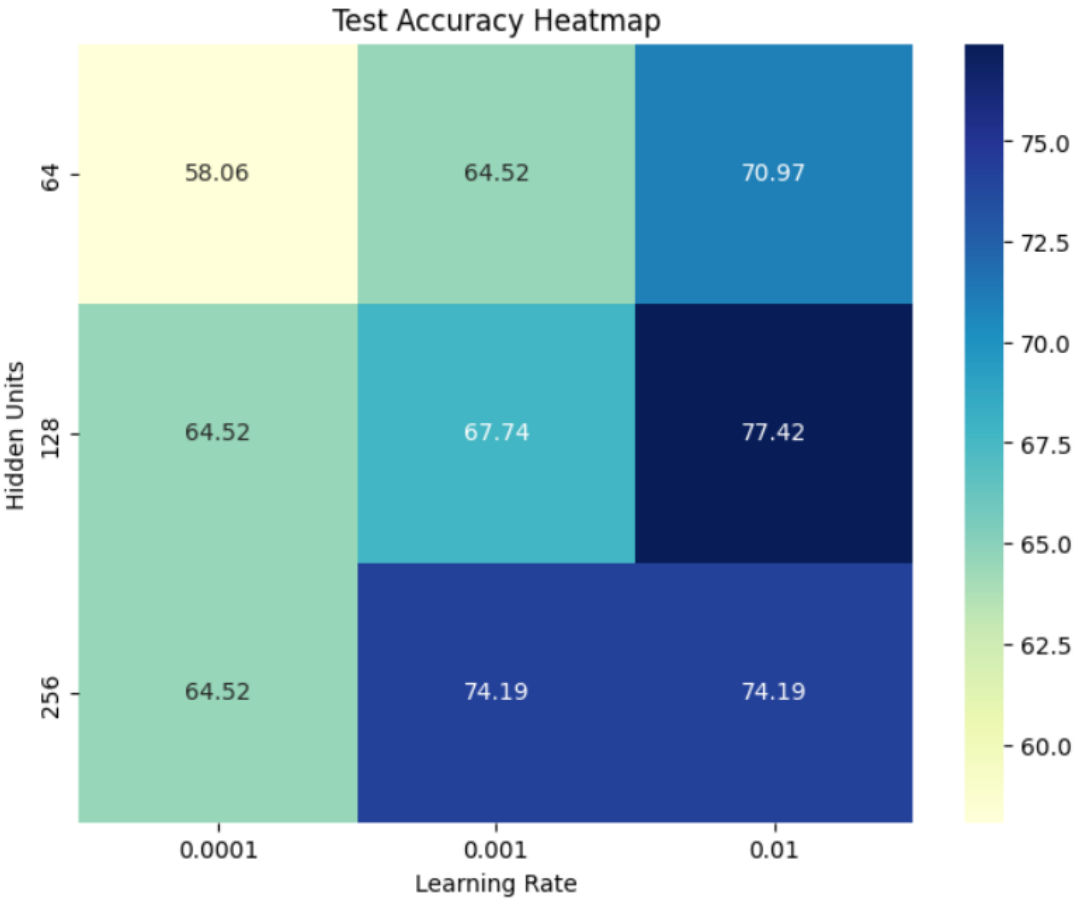


Training with hidden_units=256, learning_rate=0.0001





We can see higher learning rate perform better. Low learning rates may prevent the model from learning complex patterns effectively within a fixed number of epochs.



The heatmap shows that the learning rate has a strong impact on model performance. A learning rate of 0.01 consistently produced higher test accuracy across all model

sizes. The best result was achieved with 128 hidden units and a learning rate of 0.01, reaching 77.42% accuracy. While increasing hidden units does help, the gains plateau after 128 units.

3. (20 pts) In Lab 2, you may have noticed a discrepancy in accuracy between the training and test datasets. What do you think causes this occurrence? Discuss potential reasons for the gap in accuracy. (Approximately 100 words.)

There are a few reasons I think for the discrepancy in accuracy between the training and test datasets. First, about overfitting, the model learned the training data better which includes noise and cool patterns that are bad for generalizing. We can see when using 256 hidden units, train accuracy is high and test accuracy is low usually represents overfitting. And higher learning rate converges fast and performs well but may "memorize" training data if unchecked. Second, the model is lack of regularization, there are no dropout, weight decay, or early stopping observed. It is likely to be overfitting when parameters are more. Third, I would like to talk about data imbalance, it is common in medical datasets just like heart disease prediction. Training dataset distribution is not so like the test dataset, so the model might not generalize well. Last, small dataset also leads to the high risk of overfitting, we need more data.

4. (20 pts) Discuss methodologies for selecting relevant features in a tabular dataset for machine learning models. Highlight the importance of feature selection and how it can impact model performance. You are encouraged to consult external resources to support your arguments. Please cite any sources you refer to. (Approximately 100 words, excluding reference.)

Methodologies for selecting relevant features in a tabular dataset:

A. Filter Methods: Use statistical techniques independent of any model. Like correlation matrix, chi-square test and ANOVA F-test. Correlation matrix will drop highly correlated features, for our heart disease dataset, 'Serum cholesterol in mg/dl' and 'Resting blood pressure' would be select for

example. Chi-square test is for categorical variables. And ANOVA F-test is for numerical vs categorical target.

B. Wrapper Methods: Use the model's performance as a guide. Like Recursive Feature Elimination (RFE), which iteratively removes least important features. And forward/backward feature selection, start with none or all, add or remove one feature at a time.
   These are slower but usually more accurate than filter methods.

C. Embedded Methods: Feature selection is built into the model training process. Like L1 regularization (Lasso Regression), it shrinks irrelevant features' weights to 0. And Tree-based models (Random Forest, XGBoost) which provide feature importance scores. It will be useful in our dataset.

Importance of feature selection and how it can impact model performance:

Feature selection can improve model accuracy since it removes irrelevant or noisy data that may confuse the model, focus on the most informative inputs. Effectively reduces overfitting, making fewer complex models generalize better, they are less likely to memorize training data. Also, simpler models are easier to understand and explain, which is essential in domains like healthcare or finance. Intuitively fewer features mean fewer computations, algorithms like decision trees or SVMs train much faster when input dimensionality is reduced. It is worth noting that sacrifice some training accuracy to gain better generalization on unseen data.

External resources:

An Introduction to Variable and Feature Selection
https://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf

5. (20 pts) While artificial neural networks (ANNs) are versatile, they may not always be the most efficient choice for handling tabular data. Identify and describe an alternative deep learning model that is better suited for tabular datasets. Explain the rationale behind its design specifically for tabular data, including its key features and advantages. Ensure you to reference any external sources you consult. (Approximately 150 words, excluding reference.)

Identify and describe an alternative deep learning model that is better suited for tabular datasets:

TabNet, it incorporates attention mechanisms to select the most relevant features at each decision step, mimicking how tree-based models, like XGBoost make splits. It process raw tabular data and trains using gradient descent-based optimization. At each decision step, it applies sequential attention to dynamically select relevant features. This mechanism improves interpretability by directing learning capacity toward the most informative inputs. TabNet performs both feature selection and extraction within a unified deep learning framework using a method known as soft feature selection.

Explain the rationale behind its design specifically for tabular data, including its key features and advantages:

- Sequential Attention: At each decision step, TabNet learns which features to focus on, allowing dynamic feature selection.
- Sparse Feature Masking: Only use a subset of features, improving interpretability and regularization.
- Interpretable: Visualize which features the model attends to.
- Backprop to improve decisions and weights, which gives more control.
- LR reduction uses fine-tuning approaches that work for all deep learning principles such as special loss.

External sources:

TabNet: Attentive Interpretable Tabular Learning

https://arxiv.org/abs/1908.07442

https://medium.com/@turkishtechnology/deep-learning-with-tabnet-b881236e28c1