113034507 龔良軒 HW2

Q1. (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.

A1: I conducted experiments by varying two types of hyperparameters: number of layers (included hidden layer and output layer) and learning rate.
There are 256 units of neurons in each hidden layer. I set 3 different values for Number of Layers: 2, 3, 4 and set 3 different values for Learning Rate: 0.01, 0.001, 0.0001. The final experiment result is below:

# of Layers	Learning Rate	Train Loss	Train Accuracy	Validation Loss	Validation Accuracy	Test Loss	Test Accuracy
2	0.01	0.38	80.95%	0.55	80.25%	0.5	67.74%
2	0.001	0.4	79.89%	0.53	76.54%	0.5	70.97%
2	0.0001	0.57	70.37%	0.58	72.84%	0.64	67.74%
3	0.01	0.37	81.48%	0.49	82.72%	0.44	70.97%
3	0.001	0.42	79.89%	0.58	75.31%	0.58	67.74%
3	0.0001	0.56	73.02%	0.59	72.84%	0.67	61.29%
4	0.01	0.34	84.66%	0.52	82.72%	0.48	70.97%
4	0.001	0.39	82.01%	0.56	79.01%	0.55	74.19%
4	0.0001	0.56	73.02%	0.59	69.14%	0.67	64.52%

 # The row with orange Background is the initial setting in Lab2.

Q2. (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.)

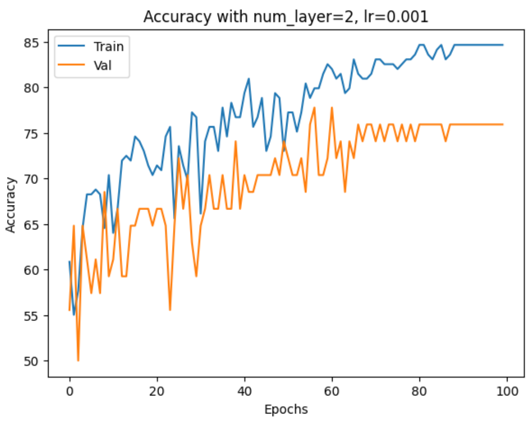
A2: First we fixed the learning rate and added more hidden layer in our ANN model. The result is below:

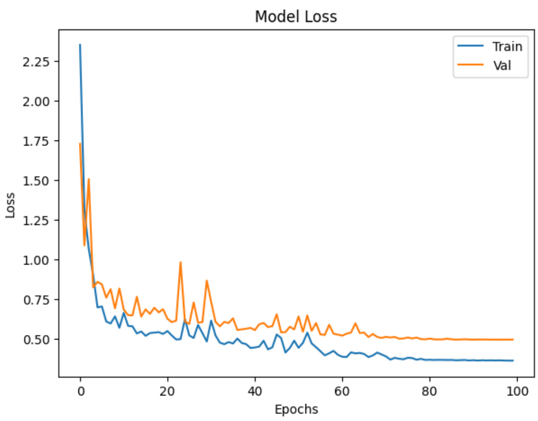
# of Layers	Learning Rate	Train Loss	Train Accuracy	Validation Loss	Validation Accuracy	Test Loss	Test Accuracy
2	0.001	0.4	79.89%	0.53	76.54%	0.5	70.97%
3	0.001	0.42	79.89%	0.58	75.31%	0.58	67.74%
4	0.001	0.39	82.01%	0.56	79.01%	0.55	74.19%

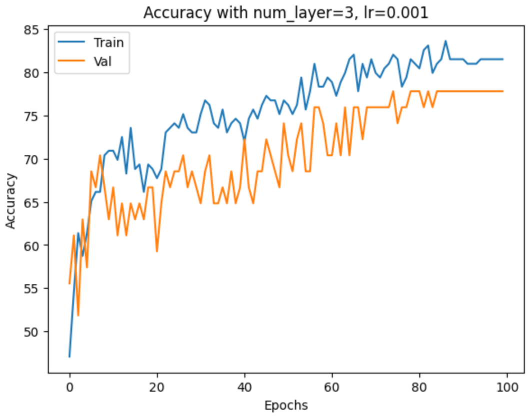
 # The row with orange Background is the initial setting in Lab2.

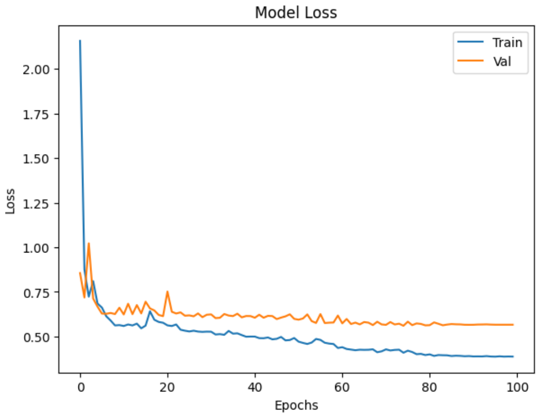
The loss between different groups seems to not change significantly. However, the accuracy of the training, validation, and test sets improves as the number of layers increases. In this case, adding more hidden layers to the ANN model appears to improve accuracy performance.

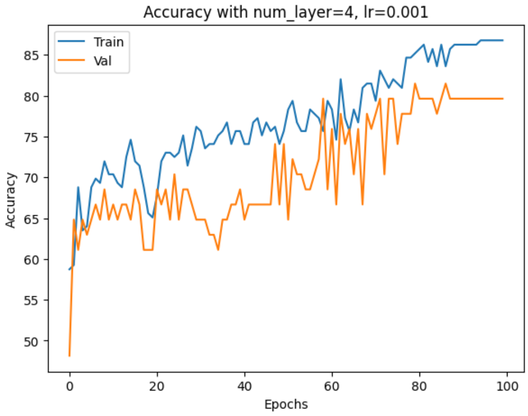
We can observe the learning curves under three different settings: As we increase the number of hidden layers, we also increase the model's complexity. This change causes the training loss to drop significantly during the training process, suggesting that we may achieve a better prediction model. However, one thing to note is that there could be an overfitting problem as the model becomes too complex. As a result of overfitting, the performance on the training data would be good, but the performance on the validation and test data would contrast negatively.

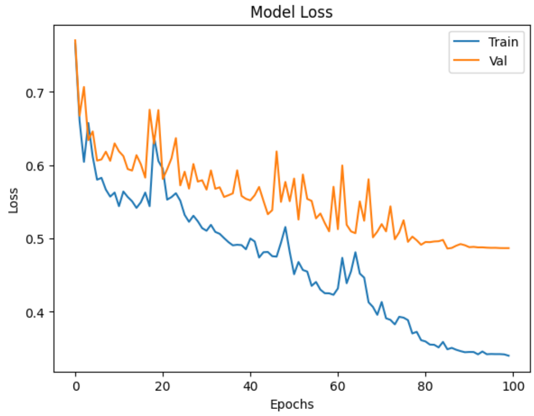










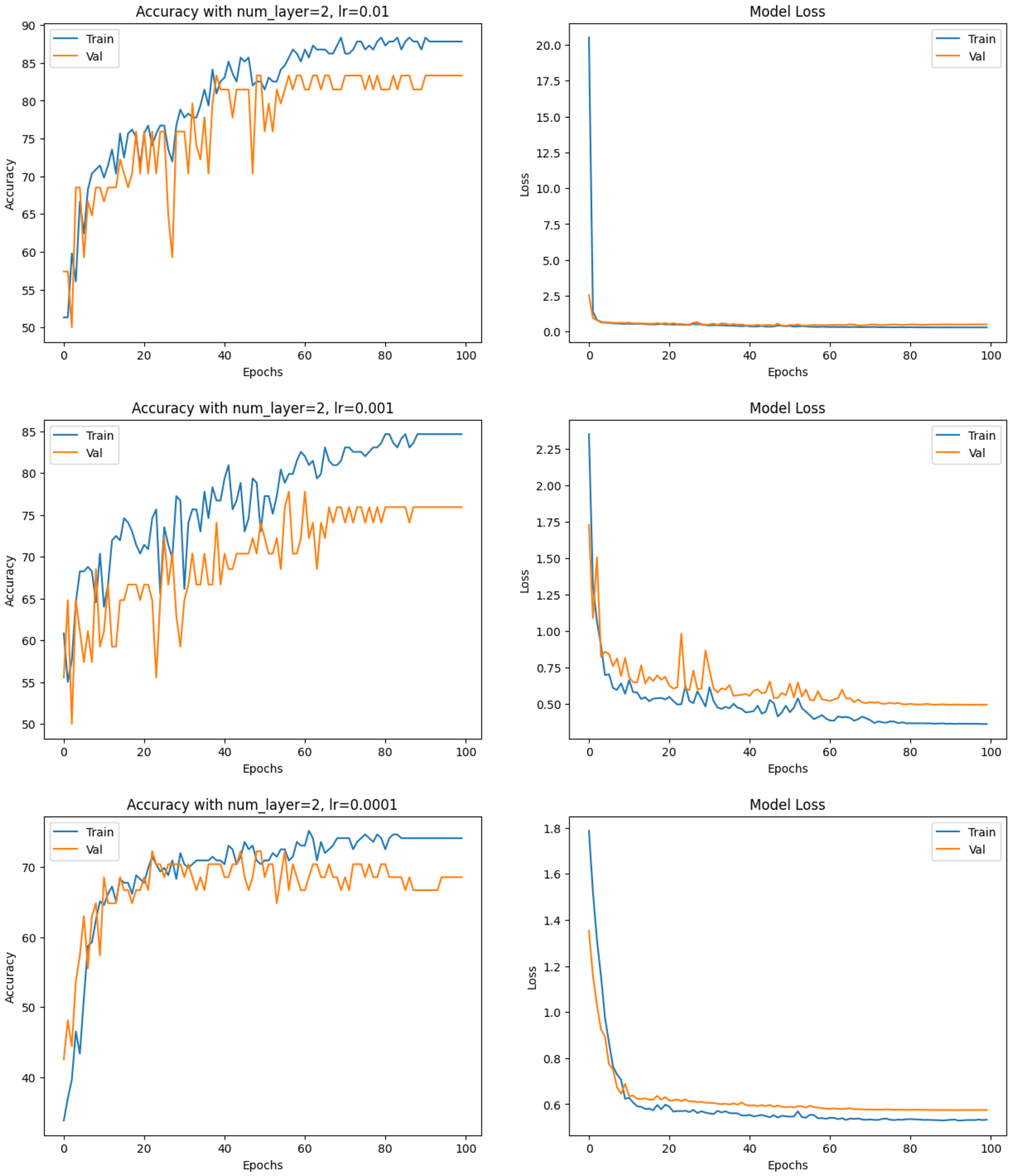


In the second part, we fixed the number of layers and decreased the learning rate in our ANN model. The result is below:

# of Layers	Learning Rate	Train Loss	Train Accuracy	Validation Loss	Validation Accuracy	Test Loss	Test Accuracy
2	0.01	0.38	80.95%	0.55	80.25%	0.5	67.74%
2	0.001	0.4	79.89%	0.53	76.54%	0.5	70.97%
2	0.0001	0.57	70.37%	0.58	72.84%	0.64	67.74%

The loss of train, validation and test rise while decreasing the learning rate. The accuracy of train, validation dropped and the accuracy of test roughly decreased while decreasing the learning rate. In this case, reducing the learning rate will improve the performance of accuracy.

Then, we can observe the learning curves with 3 different settings: If we set the learning rate to 0.0001, the training loss will decrease slowly. A possible way to improve model is to increase the epoch of training process.



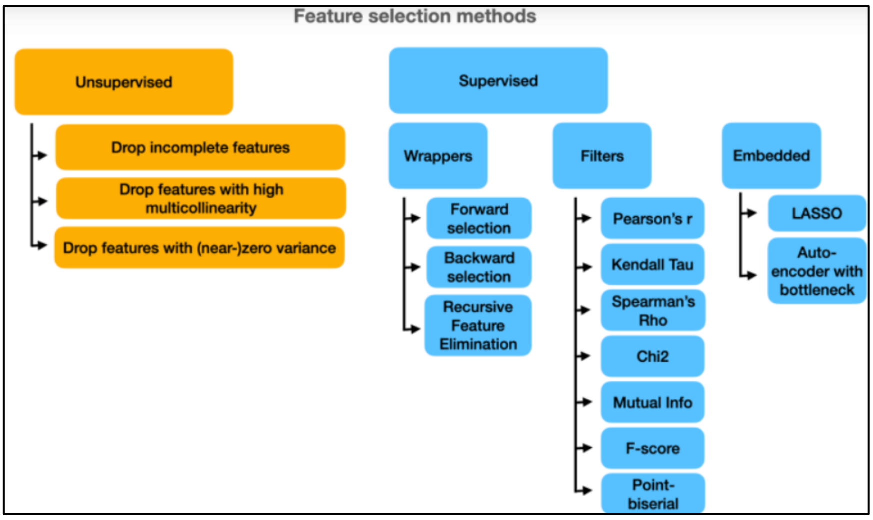
Q3. (20 pts) In Lab 2, you may have noticed a discrepancy in accuracy between the training and test datasets. What do you think causes this occurrence? Discuss potential reasons for the gap in accuracy. (Approximately 100 words.)

A3: There are several potential reasons for the gap in accuracy between the training and testing (or validation) datasets. One common reason is overfitting, which occurs when the model is too complex and learns patterns specific to the training data, including noise. As a result, it performs well on the training set but poorly on unseen data. Another possible reason is a distribution shift between the training and testing sets, meaning the input features (X) do not follow the same distribution. To address this, it is important to apply standardization to ensure consistency. Additionally, class imbalance in the dataset—where some classes are overrepresented—can bias the model toward predicting the majority class, reducing its generalization ability. Furthermore, because the number of test samples is very small, the performance can vary significantly, making the results less stable and reliable.

Q4. (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.)

A4: There are many feature selection methods for tabular data, such as filter methods, wrapper methods, embedded methods, and other unsupervised algorithms.

- Filter method evaluates the relevance of features independently of any machine learning model.
- Wrapper method uses a predictive model to evaluate combinations of features and select the best subset based on model performance.
- Embedded Method: Feature selection happens during model training, based on how important the model deems each feature.
- Others: such as PCA、ICA、NMF…and so on.



By performing feature selection, we can reduce irrelevant and redundant features to improve model performance while also decreasing the dimensionality of the feature space, thus avoiding the curse of dimensionality. Additionally, with fewer features, the training time can be significantly reduced. Moreover, when there are too many features, the model's interpretability tends to decrease, which is an important aspect alongside performance. Feature selection not only helps in making the model more efficient but also improve the performance in practical applications.

Reference:
https://neptune.ai/blog/feature-selection-methods
https://www.stratascratch.com/blog/feature-selection-techniques-in-machine-learning/

Q5. 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.)

A5: TabNet is a deep learning architecture specifically designed for tabular data. It leverages a sequential attention mechanism to dynamically select the most relevant features at each decision step. TabNet has demonstrated superior performance compared to traditional neural networks and decision tree-based models across various tabular benchmarks.

The key features of TabNet is "Attentive transformer", which uses sequential attention mechanism to select the most important features for each decision step. The attentive transformer utilizes sparse matrix multiplication, making it highly efficient for processing tabular data.

TabNet has several advantages for processing tabular data below:

- High Interpretability: TabNet's feature selection mechanism directly pointed out which features are most important for model prediction.
- Efficiency: TabNet focuses on learning from the most relevant features at each decision step, making it more efficient than other models that process all features simultaneously. This approach is particularly beneficial for large tabular datasets with many features.
- Great Performance: TabNet has been shown to outperform other neural network and decision tree variants on a wide range of tabular datasets due to its ability to catch more complex interaction between features.

Reference:
https://www.machinelearningexpedition.com/tabnet-tabular-neural-network/
https://github.com/dreamquark-ai/tabnet?ref=machinelearningexpedition.com