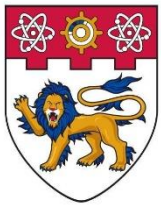

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

AY2021 Semester 1

CE4042

Neural Networks and Deep Learning

Assignment 1

Name: Raymond Toh

Matric Number: U1821599H

Tutor: Jagath Chandana Rajapakse

1 Table of Contents

2 Table of Figures

3 Table of Tables

4 Introduction

The goal of assignment 1 is to allow us to have a better understanding of how to obtain the optimal hyperparameters for model selection and to solve overfitting problems.

Part A of assignment 1 is to build a classification neural network and obtain optimal hyperparameters by using k-fold cross-validation. Optimal model will be retrained with optimal hyperparameters and used for the classification problem. The dataset uses to train the model is Cardiotocography dataset with 2 labels and 21 input attributes.

Part B of assignment 1 is to build a regression model to predict the admission rate for master programs. The Graduate admissions Predication dataset will be used to train the model. It consists of several parameters and a prediction rate of admission ranging from 0 to 1.

5 Methods

The following methods are used for this assignment.

5.1 K-fold cross-validation

It provides train and test indices to split data into train and test set. Dataset are split into k consecutive fold. Each fold will be used once as validation while the k-1 remaining folds will be the training set. This will allow us to evaluate the average accuracy for the model.

5.2 L2 Regularization

Regularizer allow us to add penalties on layers during optimization. These penalties will be summed into the loss function. This will reduce the chance of model getting overfitted.

5.3 Dropout Regularization

Dropout allow us to set the probability of eliminating neurons during each iteration of training steps. This will reduce the overfitting problem of the model.

5.4 Recursive Feature Elimination (RFE)

RFE is a feature selection method that eliminate the weakest input features that provide no better results.

5.5 Early Stopping

Early stopping allows us to stop the training process before the model get overfitted. It usually stops when the model start converging.

5.6 Batch Size

Batch size indicates the number of samples that will propagates through the network sequentially. This assignment used mini-batch gradient descent methodology. Hence, the learning of the network will fluctuate more than a full batch gradient decent. To reduce the fluctuation, Adam optimizer was used in this assignment.

5.7 Layers of Network

Numbers of Layers will determine the complexity of features that the neuron network will train. However, if there are not enough features to train and there are many hidden layers in the network, the model will tend to overfit.

6 Part A

In this classification problem, we will design a model with these few steps:

1. Design 3-Layer Neuron Network
2. Obtain optimal batch size using 5-fold cross-validation
3. Obtain optimal hidden neuron size using 5-fold cross-validation
4. Obtain optimal weight decay using 5-fold cross-validation
5. Comparison between 3-layer network and predefined 4-layer network

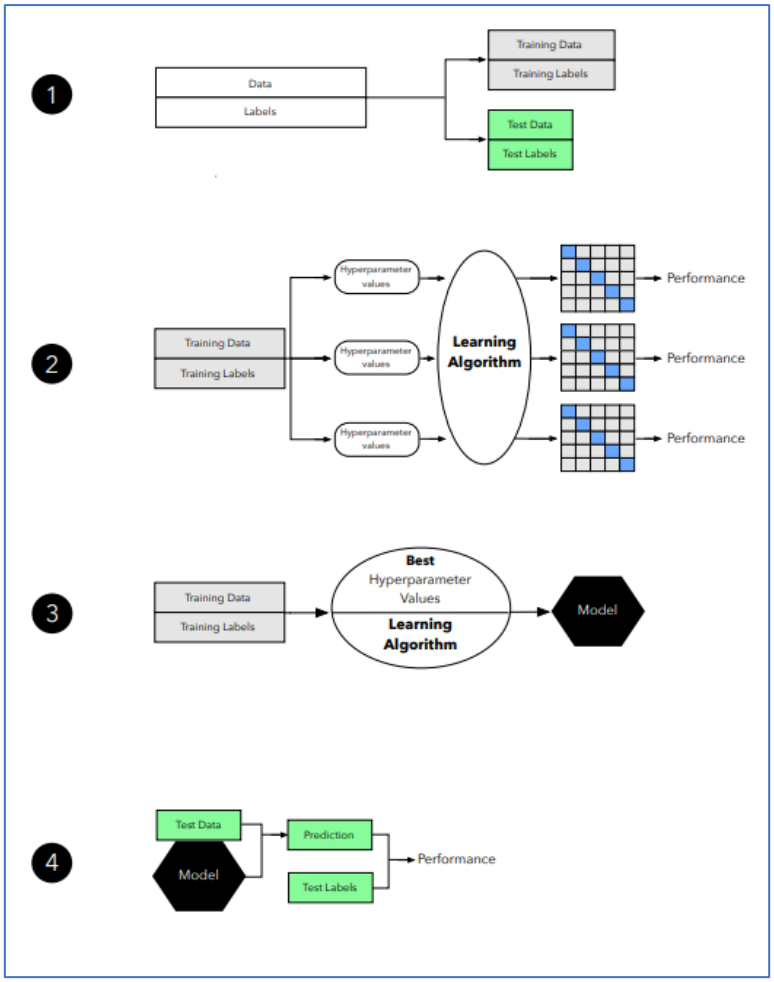

Figure 1: K-Fold Cross-validation for optima model [1]

Part A Q1 Design 3-Layer Neuron Network

A 3-layer Sequential Network was implemented with given specification:

- Learning rate – 0.01
- L2 weight decay – 10^{-6}
- Batch size – 32
- Neuron size (hidden layer) – 10
- Hidden layer activation function – ReLU
- Output activation function – Softmax

As model did not specify the optimizer, we will be using Adam in this assignment. Dataset was split into 70:30 ratio for training set and testing set. We will only be using testing set on Part A Q5. Training set will be used on Part A Q2 to Part A Q4 for 5-fold cross-validation.

Firstly, we sampled 1000 data from training set to evaluate the 3-layer network. The network was trained over 500 epochs and the train and test accuracies with given specification are shown below in Figure 2.

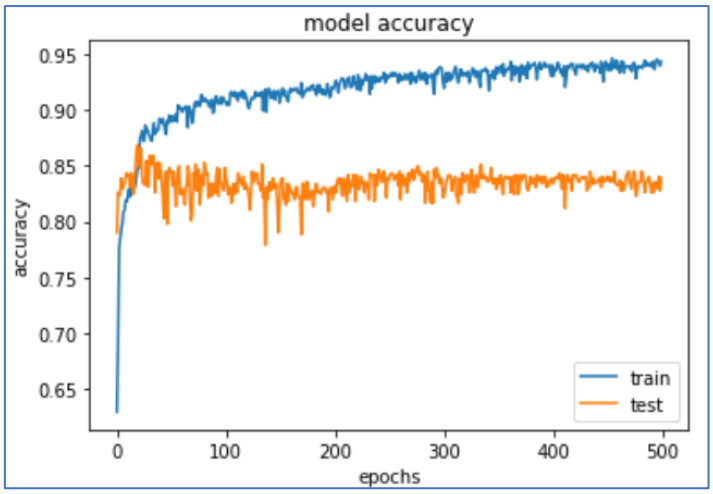

Figure 2: 3-Layer Network Train and Test Accuracies

3-Layer Network Train Accuracy	94.30%
3-Layer Network Test Accuracy	83.86%

Table 1: Model Evaluation

In Figure 2, test error begins to converge at estimated 50 epochs. Hence, we will be using 50 epochs for the cross-validation in the next few questions. Table 1 tell us that this model has overfitted as it does not generalise well with test set.

Part A Q2 Obtain Optimal Batch Size using 5-Fold Cross-Validation

Next, we tested the given batch size of {4,8,16,32,64} with the model and obtain the average of accuracy and loss for individual 5-fold cross-validation. The average of accuracy, loss and test error graph are plotted in Figure 3,4 and 5 respectively. The summary of results is shown in Table 2.

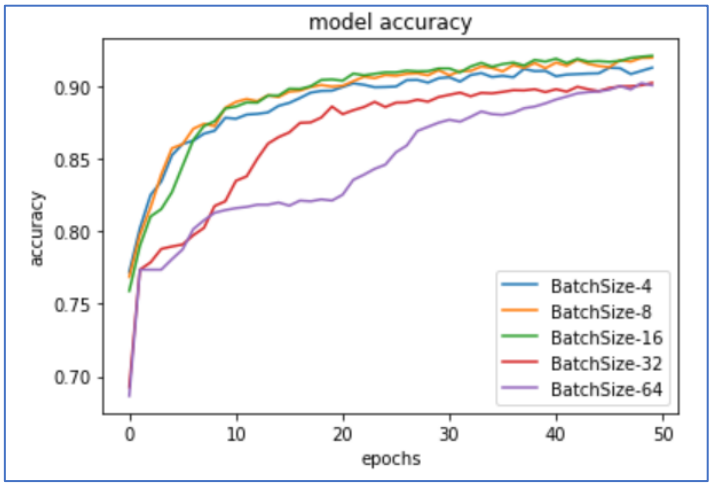

Figure 3: Average Accuracy results for different Batch Size

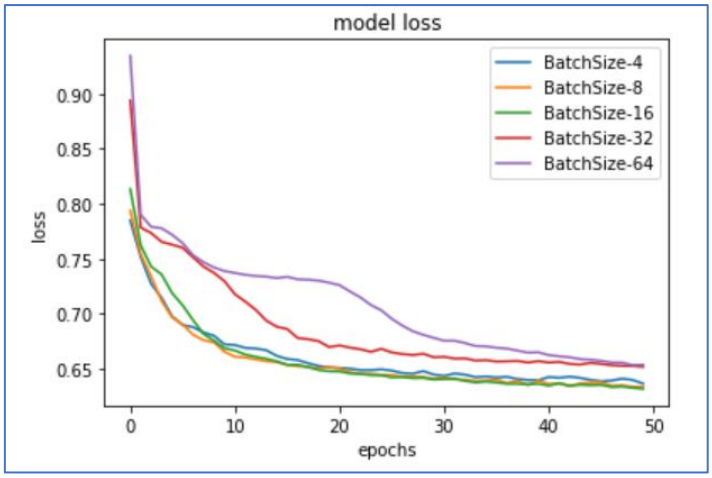

Figure 4: Average Loss results for different Batch Size

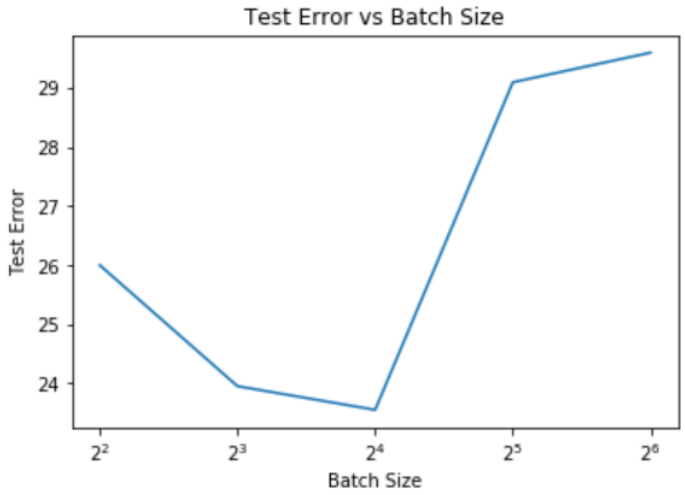

Figure 5: Graph of Average Test Error against Batch Size

Batch Size	Avg. Time Taken (s)	Avg. Accuracy (%)	Avg. Loss	Avg. Test Error
4	45.65s	91.25%	0.647	25.99
8	27.58s	91.94%	0.633	23.95
16	18.19s	92.07%	0.631	23.55

32	13.90s	90.21%	0.651	29.09
64	10.85s	90.04%	0.653	29.59

Table 2: Summary of results for different Batch Size

From the summary of results in Table 2, we can conclude that batch size 16 is optimal for this model as it has the highest accuracy and least loss with an average time taken. Also, we noticed that the increase of batch size, the shorter time it takes to train the network, and model converge at lower accuracy with increase in batch size. Batch size 16 will be used for the rest of the experiments.

Part A Q3 Obtain Optimal Hidden Neuron Size using 5-Fold Cross-Validation

Given 16 as the optimal batch size, we will now be testing neuron size of {5,10,15,20,25} in the hidden layer. 5-Fold Cross-Validation will be used and the average of accuracy and loss of each fold are recorded. The graph of the average for accuracy, loss and test error are plotted in Figure 6,7 and 8 respectively. The summary of results is shown in Table 3.

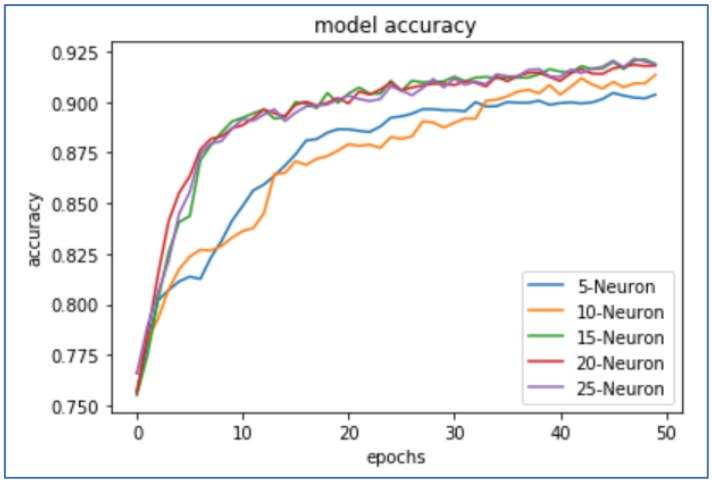

Figure 6: Average Accuracy results for different Neuron Size

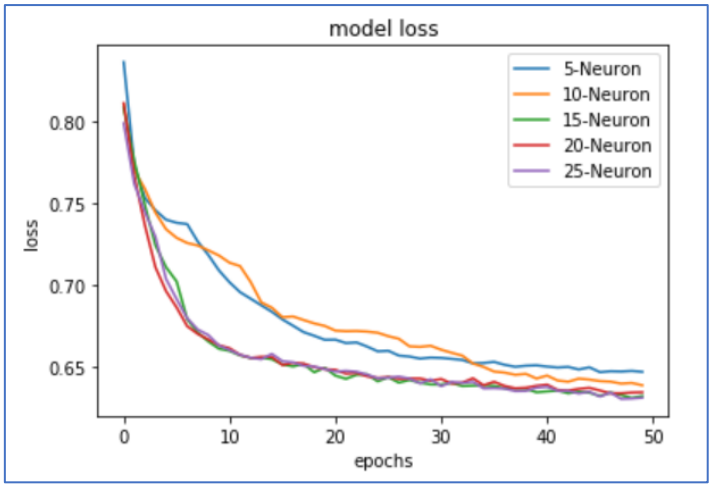

Figure 7: Average Loss results for different Neuron Size

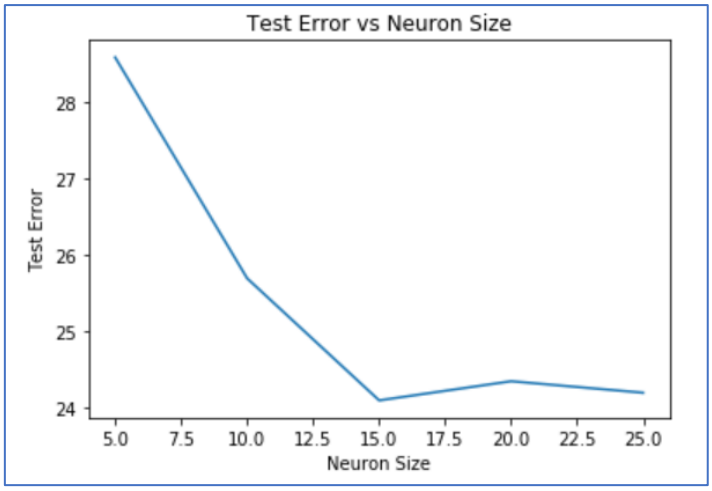

Figure 8: Graph of Average Test Error against Neuron Size

Neuron Size	Avg. Accuracy (%)	Avg. Loss	Avg. Test Error
5	90.37%	0.647	28.59
10	91.35%	0.639	25.70
15	91.89%	0.633	24.10
20	91.80%	0.635	24.35
25	91.85%	0.632	24.20

Table 3: Summary of results for different Neuron Size

From the summary of results in Table 3, we can see that neuron size of 15, 20 and 25 are comparable. However, in my opinion, neuron size of 15 is optimal as it has the highest accuracy and least test error with one of the lowest loss. Also, we noticed that the increase of neuron size, model converges with lesser epochs. Neuron size of 15 will be used for the rest of the experiments.

Part A Q4 Obtain Optimal Weight Decay using 5-Fold Cross-Validation

With previous optimal hyperparameter of batch size and neuron size, we will now examine the weight decay of $\{0,10^{-3},10^{-6},10^{-9},10^{-12}\}$. As usual, 5-fold cross validation will be done for each fold and average of accuracy, loss and test error will be recorded and plotted in Figure 9,10 and 11. The summary of results shown in Table 4.

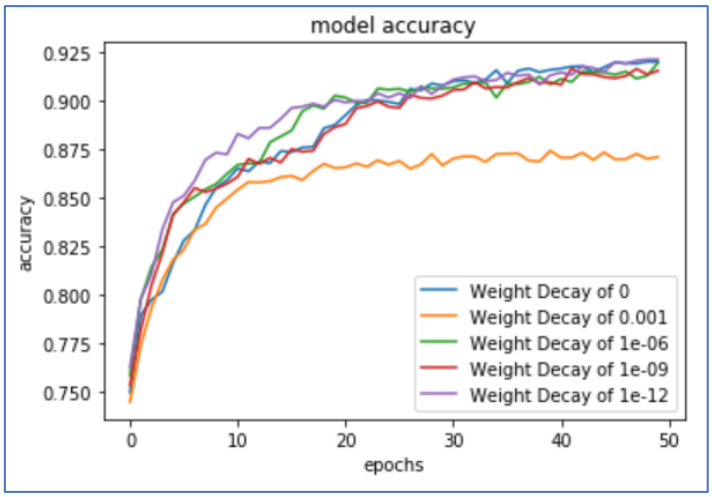

Figure 9: Average Accuracy results for different Weight Decay

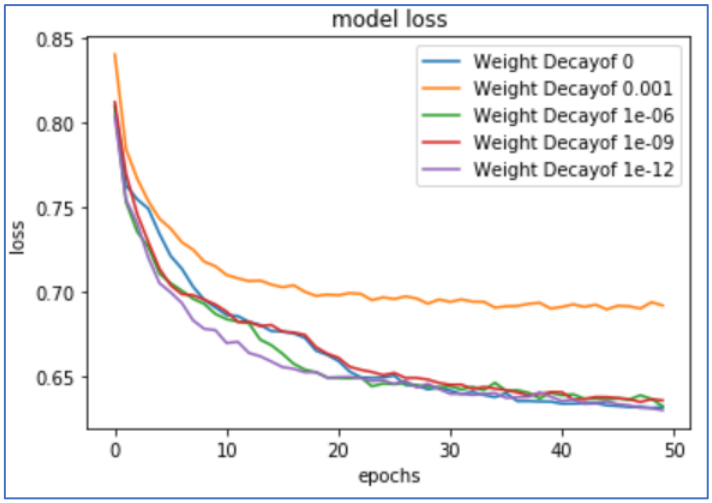

Figure 10: Average Loss results for different Weight Decay

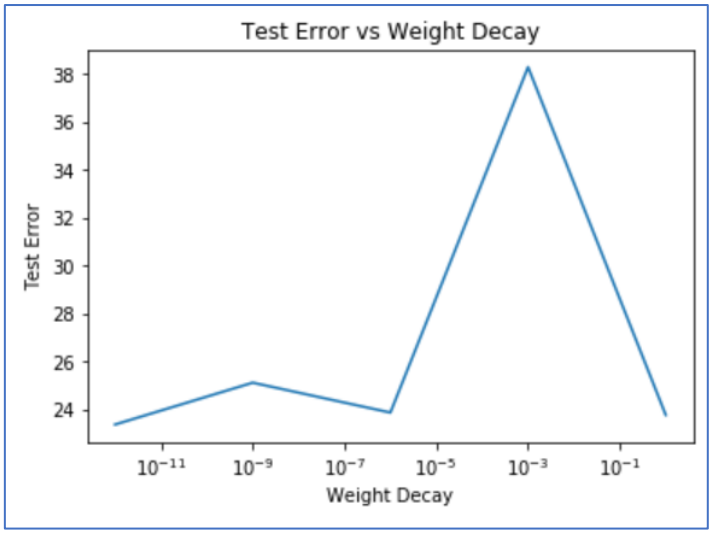

Figure 11: Graph of Average Test Error against Weight Decay

Weight Decay	**Avg. Accuracy (%)**	**Avg. Loss**	**Avg. Test Error**
0	92.00%	0.632	28.59
10^{-3}	87.11%	0.692	25.70
10^{-6}	91.97%	0.633	24.10
10^{-9}	91.55%	0.636	24.35
10^{-12}	92.14%	0.630	24.20

Table 4: Summary of results for different Decay Rate

From the summary of results in Table 4, we can see weight decay of 10^{-12} is optimal as it has the highest accuracy and least loss. Also, we noticed with higher weight decay, the likelihood of model to be underfitted increases. Weight decay of 0 means there are no penalties added to the cost function or individual loss functions, hence it converged similarly to a small weight decay value.

Part A Q5 Comparison between 3-Layer Network and Predefined 4-Layer Network

We will retrain 3-Layer network with the few optimal hyperparameter we obtained previously and compare with the predefined 4-layer network. We will use 250 epochs for this question as we are unsure of the estimated epochs that 4-layer network start to converge. Both networks are trained with 70% of dataset and tested with 30% of the dataset that we split in Part A Q1. The graph plotted of the train/test accuracy and train/test loss is shown in Figure 12 and Figure 13. Table 5 shown the summary of the overall results.

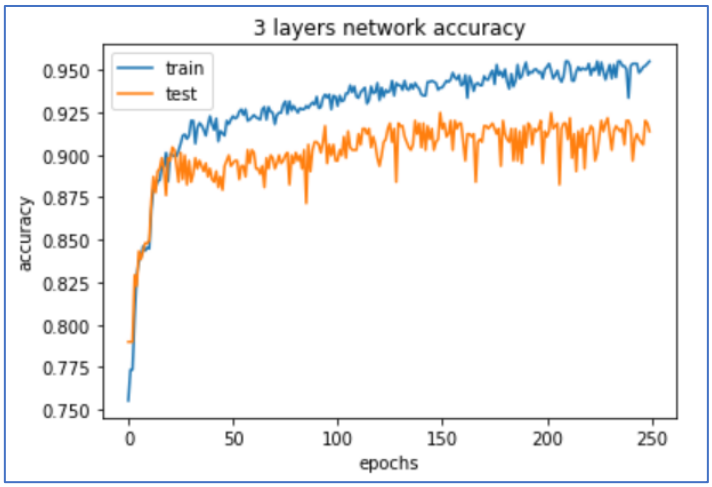

Figure 12: 3-Layer Network Train/Test Accuracy

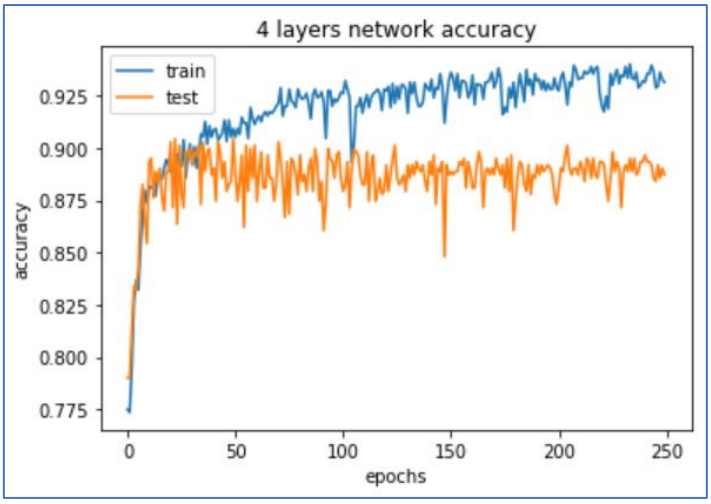

Figure 13: 4-Layer Network Train/Test Accuracy

3-Layer NN Train Accuracy	95.50%	**3-Layer NN Test Error**	55
3-Layer NN Test Accuracy	91.38%		
4-Layer NN Train Accuracy	93.15%	**4-Layer NN Test Error**	72
4-Layer NN Test Accuracy	88.71%		

Table 5: Summary of 3-Layer NN and 4-Layer NN Results

Firstly, the 3-layer optimal model have reduced the overfitting problem as compared to Part A Q1. Secondly, the 3-layer optimal model still having a minor overfitting problem as it required more data to train, hence, adding addition layers implies that more parameters to train, which also mean more data is needed. As we can see, 4-layer network have converged at around 25 epochs. In contrast, 3-layer network is still slowly increasing its test accuracy.

7 Part B

In this regression problem, we will design a model with these few steps:

1. Design 3-Layer Neuron Network
2. Remove unnecessary input features using RFE
3. Comparison between Neuron networks

Part B Q1 Design 3-Layer Neuron Network

A 3-layer Sequential Network was implemented with given specification:

- Learning rate – 0.001
- L2 weight decay – 10^{-3}
- Batch size – 8
- Neuron size (hidden layer) – 10
- Hidden layer activation function – ReLU
- Output activation function – Sigmoid (ranging from 0 to 1)

As model did not specify the optimizer, we will be using Adam in this assignment. Dataset was split into 70:30 ratio for training set and testing set.

Firstly, we will use the dataset to evaluate the 3-layer network as it is a small dataset. The network was trained over 300 epochs. However, early stopping stopped the training as it converged at estimated 104 epochs. The train and test accuracies with given specification are shown below in Figure 14.

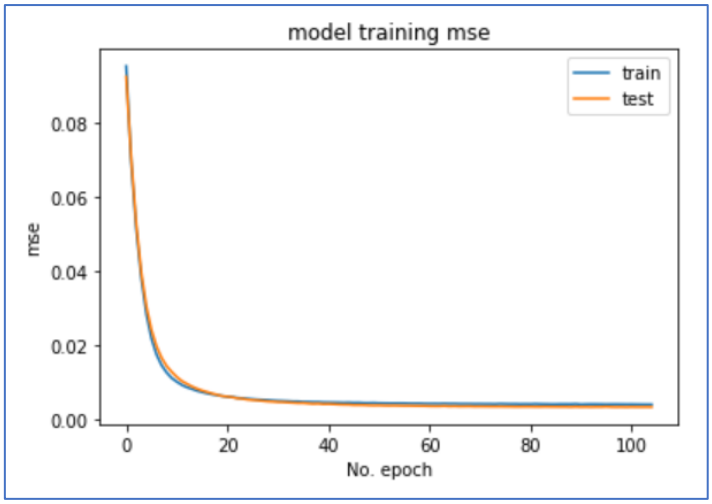

Figure 14: 3-Layer Network Train and Test Accuracies

We pick 50 random sample from dataset and do prediction on it. The results on Figure 15 shown the prediction outcome of admission rate.

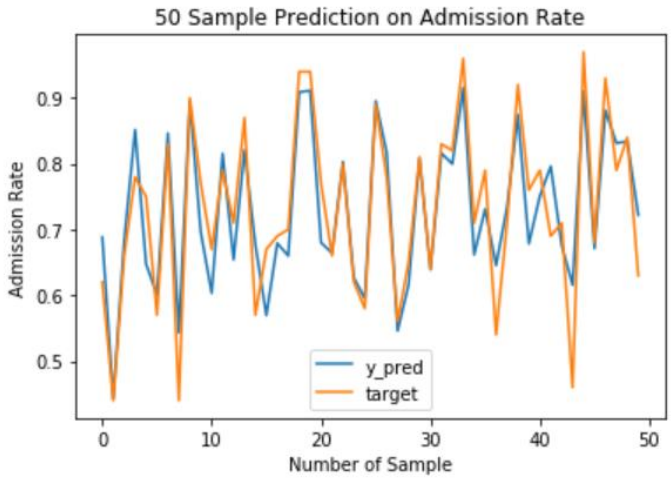

Figure 15: 50 Sample Prediction on Admission Rate

Part B Q2 Remove unnecessary input features using RFE

As current dataset has 7 input features, we will be applying RFE to remove redundant input features and obtain the optimal number of features for our network. Firstly, we will be removing features one at a time and plots its validation mean square error against number of epochs. The results shown in Figure 16. The final validation error is shown in Table 6.

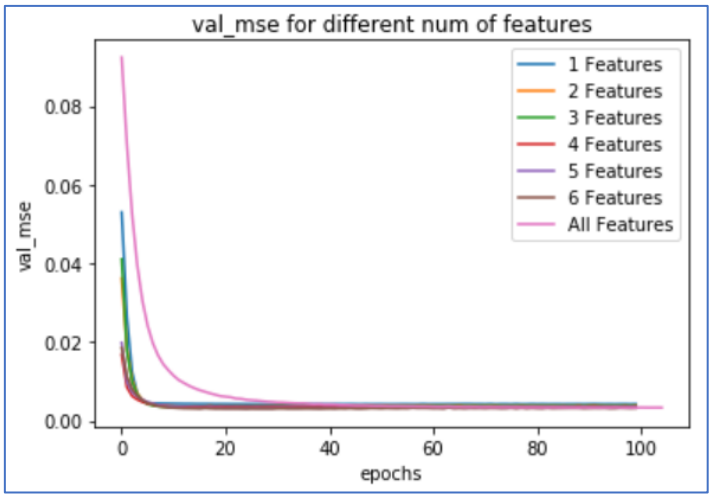

Figure 16: Validation Error for different Number of Features

Number of Features	Validation Error
1	0.004254
2	0.003796
3	0.003831
4	0.003673
5	0.003473
6	0.003233
7	0.003349

Table 6: Summary of Validation Error with different Number of Features

In Table 6, we noticed that as the features decreases, the validation error tend to increase. However, from 7 features (all input features) to 6 features, the error decreases, and 6 features to 5 features, the error increases. This shows that 6

features are the optimal number for input features. We will be using 6 features for further experiment in Part B.

Part B Q3 Comparison between Neuron networks

In this question, we will be comparing 3-layer network, 4-layer network, 4-layer network with dropout, 5-layer network and 5-layer network with dropout. All these networks will use the optimal number of input features we discover in Part B Q2. We will be using epochs of 100, batch size of 8, optimizer to be Adam and the given specification in the question to design the network. The result of mean square error and validation error is show in Figure 17 and Figure 18 respectively.

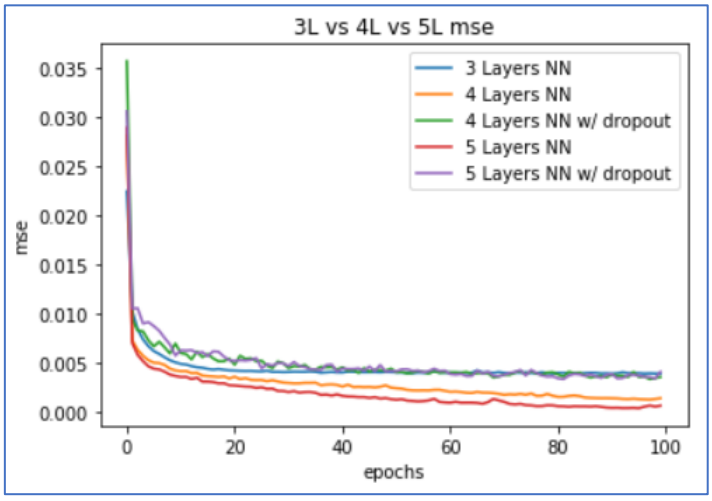

Figure 17: Mean Square Error results for different network

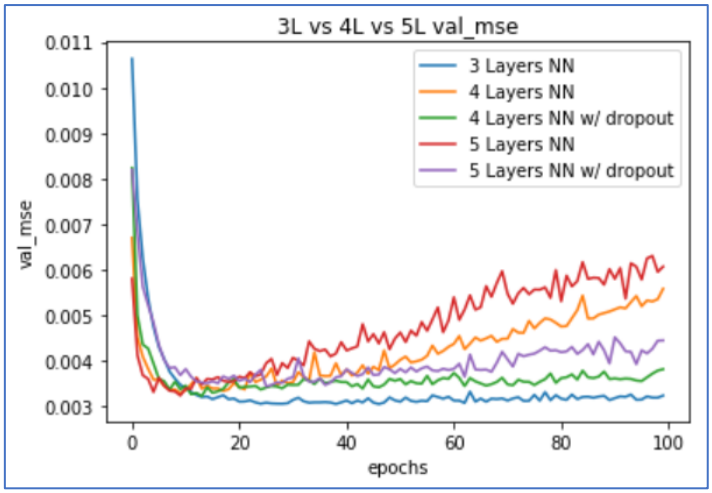

Figure 18: Validation Error results for different network

In Figure 17, we can see that 4-layer and 5-layer network without dropout have less error as compared to other network. However, in Figure 18, 4-layer and 5-layer network overfitted as the error increases over epochs. The 3-layers network converges and by adding more layer, it will overfit as discussed in Part A Q5. Although it was overfitted, layers with dropout regularization implemented helps to reduce the overfitting problem by turning off neurons for each iteration.

8 References

[1] S. Raschka, "Model evaluation, model selection, and algorithm selection in machine learning," *arXiv preprint arXiv:1811.12808,* 2018.