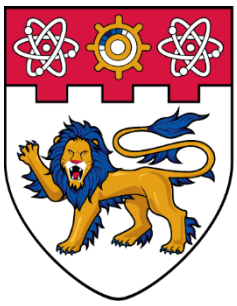


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

CZ4042: Neural Networks and Deep Learning

Assignment 1: Part A + Part B

Lin HuangJiayin

U1721208D

Contents

Introduction:

This paper consists of 2 section. The first focuses on classification, while the second focuses on regression.

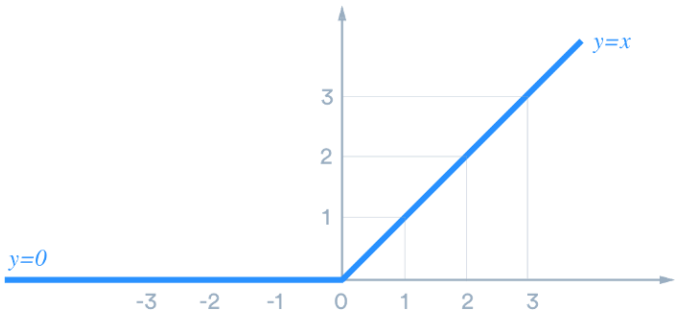
For the first section, the Cardiotocography dataset classified by expert obstetricians is being used. It contains 21 input attributes, and 2 class labels. This paper will make use of all 21 input attributes, and one of the class labels. The optimal value for batch size, number of neurons, decay parameter and layers will be determined through the experiment.

For the second section, the Graduate Admissions Prediction dataset is being used. It contains 8 input parameters, and 1 predicted parameter. This paper will make use of 7 input parameter, and the predicted parameter.

Methods:

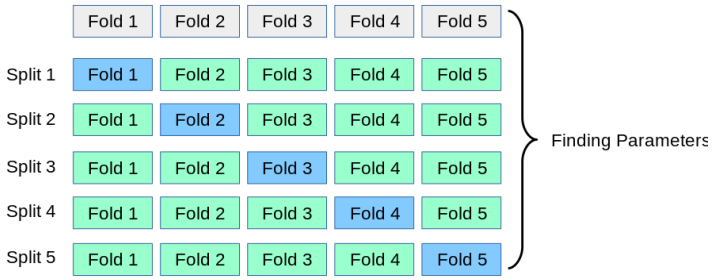
Activation functions:

A rectified-linear unit(ReLU) activation function will be applied to the hidden layer. ReLU is one of the most commonly used activation functions. It outputs the input directly and linearly when input>0. It can be written as f(u) = relu(u) = max {0,u} .



Model Selection:

The 5-fold cross-validation method will be applied on the training dataset. It is a commonly used resampling procedure for limited data sample. For this paper, the cross-validation method will be used on the training dataset, involving 70% of the total dataset. As such 70% of the dataset will be divided into 5. 1 set will be used for testing, and the other 4 will be used for training. Figure X illustrates the process of the training method.



Output layer:

A softmax layer will be used for classification. Softmax allows for a true probability distribution for a multi-class problem. The number off neurons corresponds to the output classes. As such, there are 3 neurons for this paper due to the 3 classes available: normal (N) ; suspect (S); pathologic (P)

A linear layer will be used for regression. The output provided will be scalar.

Data Pre-processing:

Part A

The initial dataset has been split into 2 parts, X and Y. X consists of the (21) input attributes, and Y consists of the class output of the respective input attributes in X. The data are then scaled and shuffled randomly.

Afterwards, the dataset is split into a 70:30 ratio for training and testing.

```
train_input = np.genfromtxt('ctg_data_cleaned.csv', delimiter= ',')

X, Y = train_input[1:, :21], train_input[1:,-1].astype(int)

X = scale(X, np.min(X, axis=0), np.max(X, axis=0))
Y = Y-1

no_data = len(X)

idx = np.arange(no_data)
np.random.shuffle(idx)
trainX, trainY= X[idx],Y[idx]

trainX = trainX[:1488]
trainY = trainY[:1488]
```

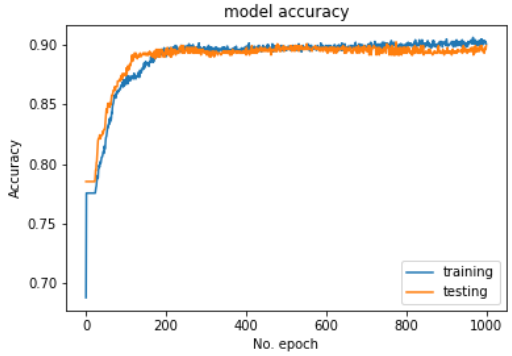
Part B

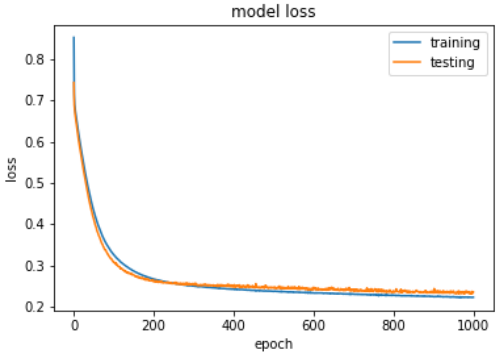
The initial dataset has been processed similarly to the dataset in part A. However, the dataset for part B underwent standard normal distribution, instead of scaling.

```
train_input = np.genfromtxt('admission_predict.csv', delimiter= ',')

X_data, Y_data = admit_data[1:,1:8], admit_data[1:,-1]
Y_data = Y_data.reshape(Y_data.shape[0], 1)

idx = np.arange(X_data.shape[0])
np.random.shuffle(idx)
X_data, Y_data = X_data[idx], Y_data[idx]

# X ~ N(0,1)
X_data = ((X_data - np.mean(X_data, axis = 0))/ np.std(X_data, axis =0))

trainX = X_data[:280]
trainY = Y_data[:280]

testX = X_data[280:]
testY = Y_data[280:]
```

Experiments, Results and Conclusions

Part A Question 1





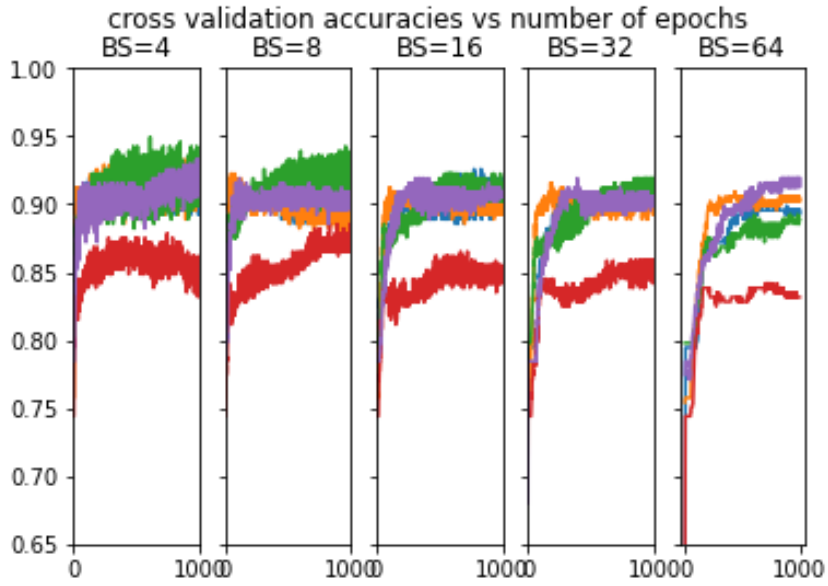
Training accuracy rate= 90.12%

Testing accuracy rate= 89.96%

The testing and training accuracies are consistent. **The test data converges around 210 epochs.**

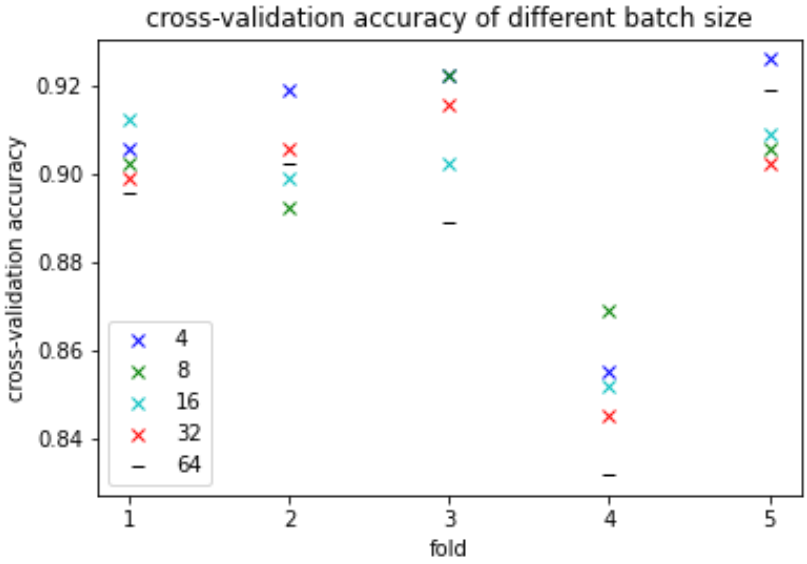
Part A Question 2

Cross-validation



The axes have been scaled such that all 5 graphs, of batch size4,8,16,32,64, are in scale to one another. Each line represents one-fold. It can generally be seen that the accuracies decrease with the increase of batch size. The vertical range for each fold decreases with the increase in batch size.

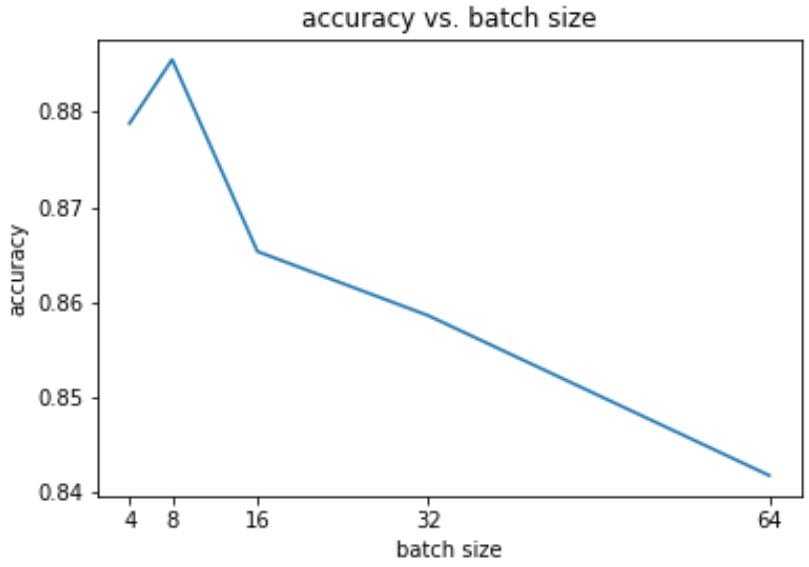
Optimal batch size



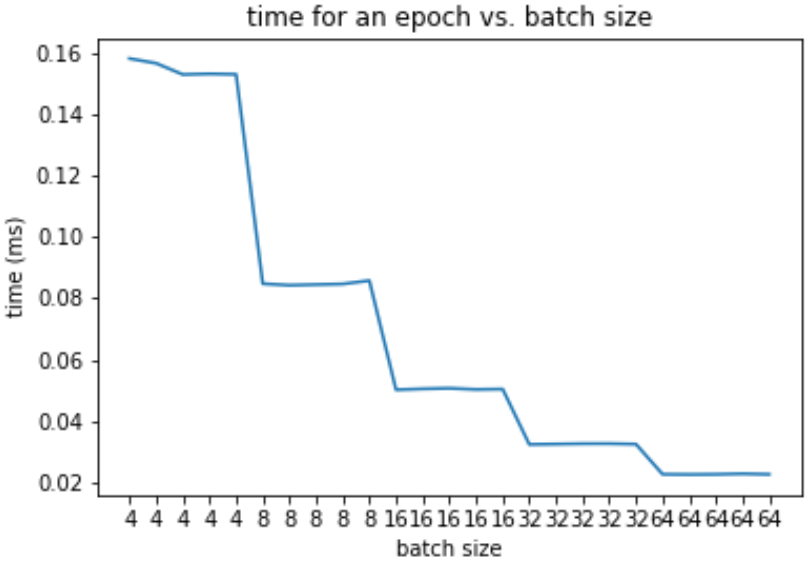
The cross-validation accuracy of each folds provides a more distinct visual of the accuracies across batch sizes. The table charts the accuracies from the graph.

	accuracies(most)				accuracies(least)

Fold1	16	4	8	32	64
Fold 2	4	32	64	16	8
Fold 3	4/8	4/8	32	16	64
Fold 4	8	4	16	32	64
Fold 5	4	64	16	8	32



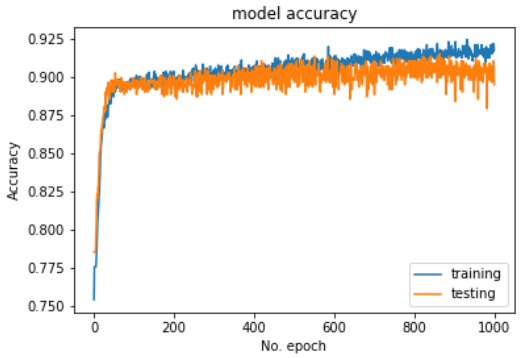
Even though the table shows favour to batch size 4 over batch size 8, the minimal accuracy graph shows that batch size 8 has the best accuracies.

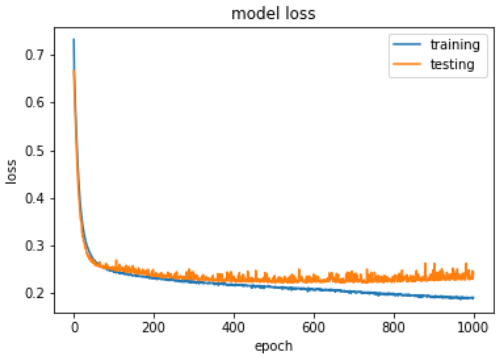


For each batch size, the network underwent 5 folds, as such the x axis contains all the different folds. There is no significant time difference between the folds, with the maximum of 0.15 for batch size of 4. The speed converges when batch size increases.

With the aid of the various graphs, **8 has been chosen has the optimal batch size.**

A batch size of 8 is thus used to replot the accuracies.





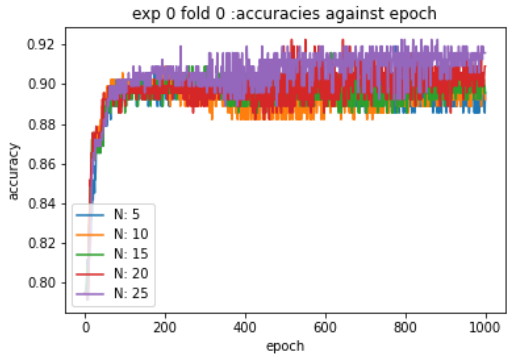
Training accuracy rate= 91.80%

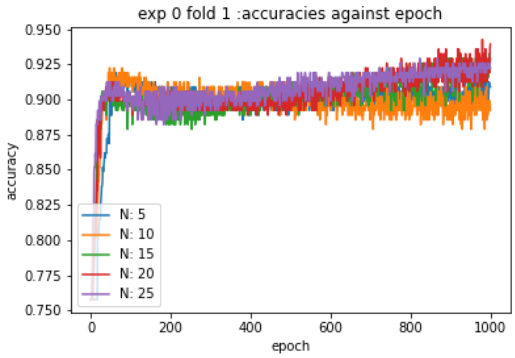
Testing accuracy rate= 89.49%

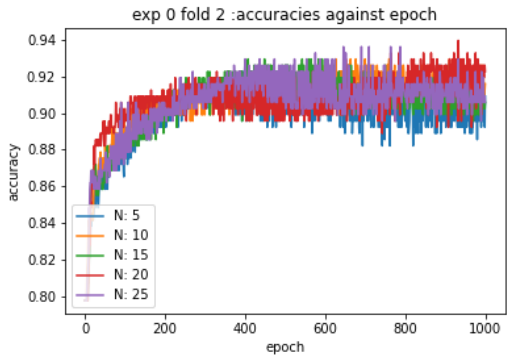
The testing results fluctuates more than the training result. There is an increase in testing loss as epoch increases. This means the model has overfit.

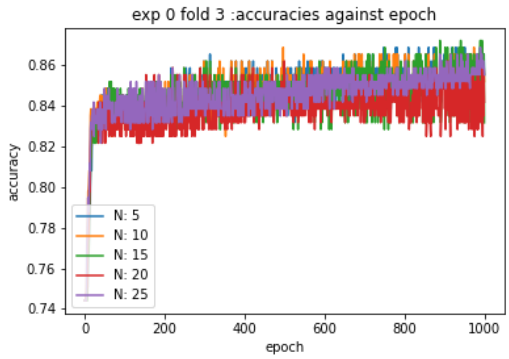
Part A Question 3

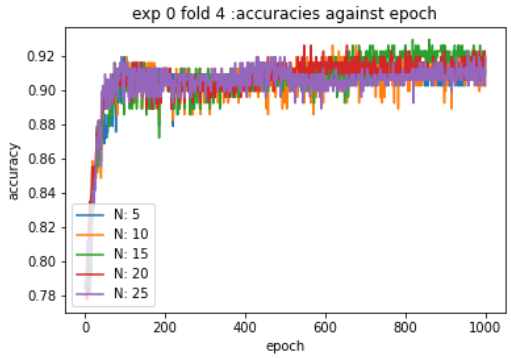
Cross-validation accuracies

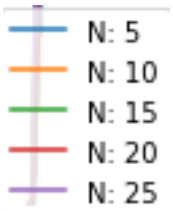






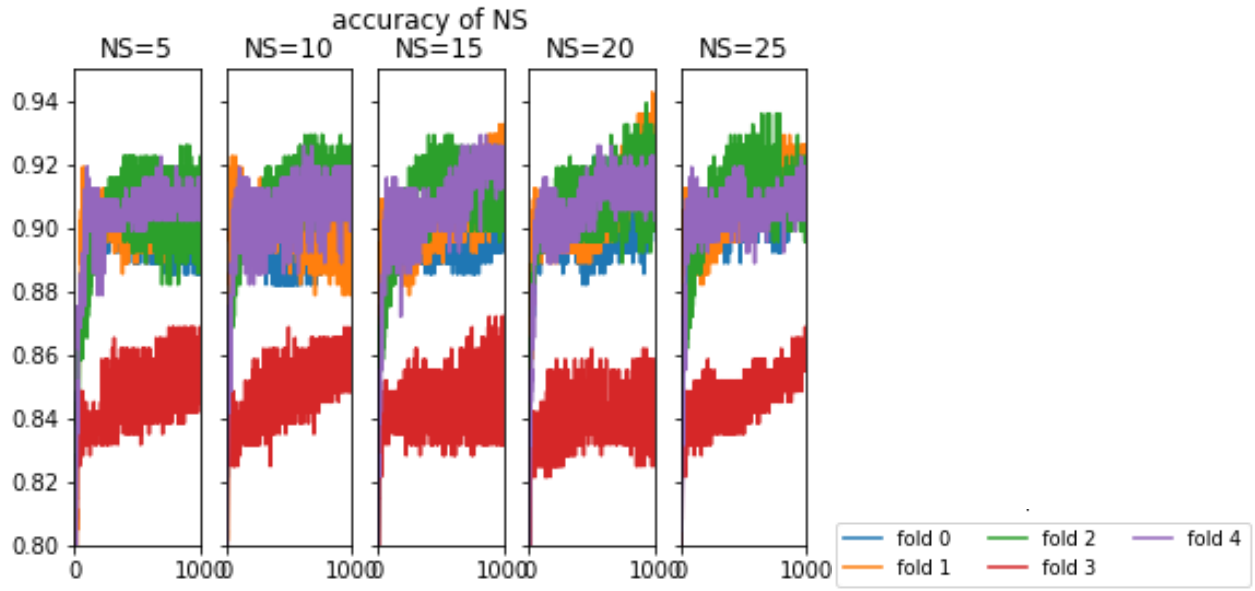




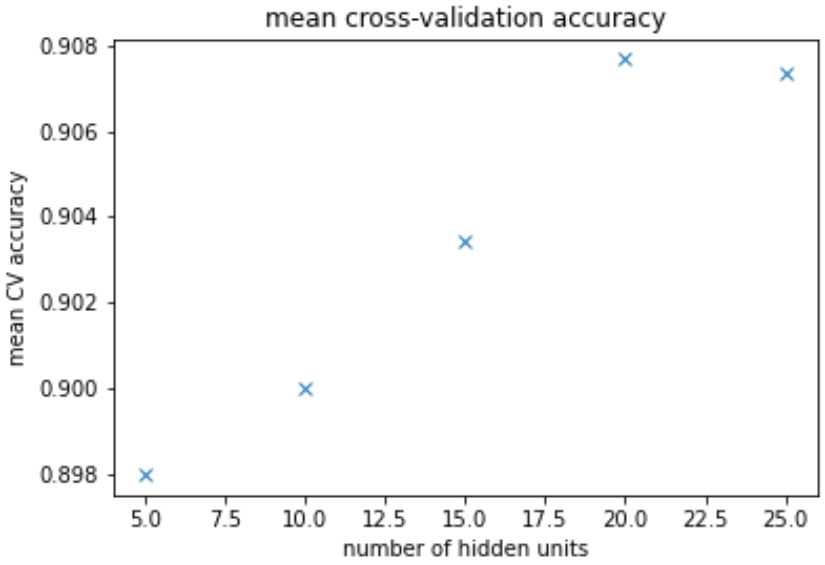


The cross-validation accuracies between different batch sizes' graphs are unable to provide any information. It is visually impossible to understand the relationship between the neuron size and the epoch.

Optimal number of neurons



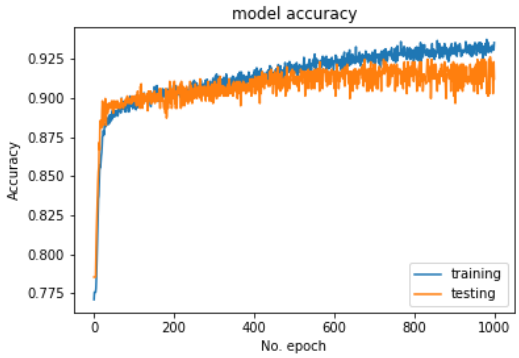
The cross-validation accuracies of different neuron size graph has been scaled to share the same y range. It provides a better representation of the accuracies provided by the different neuron size, as epochs increases.

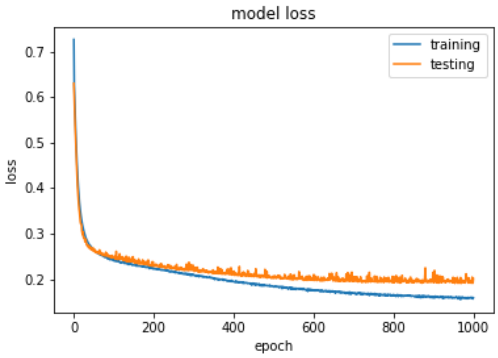


The cross-validation process repeats for 10 rounds, and the mean accuracy for each rounds and fold have been obtained and plotted again the number of neurons.

As such, the **optimal number of neurons is size 20. It is optimal since it has the highest accuracy out of all the other neuron sizes.**

A neuron size of 20 is thus used to replot the accuracies.





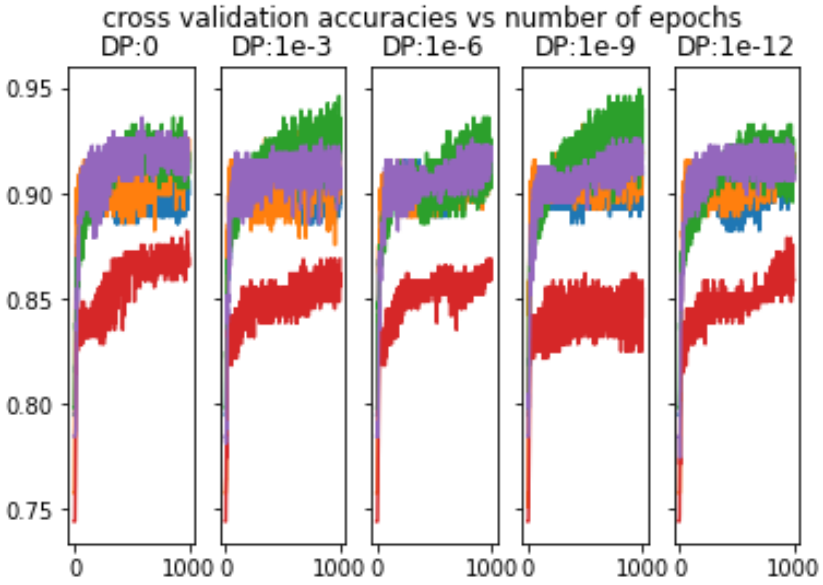
Testing accuracies: 91.22%

Training accuracies: 93.55%

The testing results fluctuates more than the training result. The difference between testing and training increases with the epochs. However, the testing loss is still on a decreasing slope, indicating that it is not overfitting

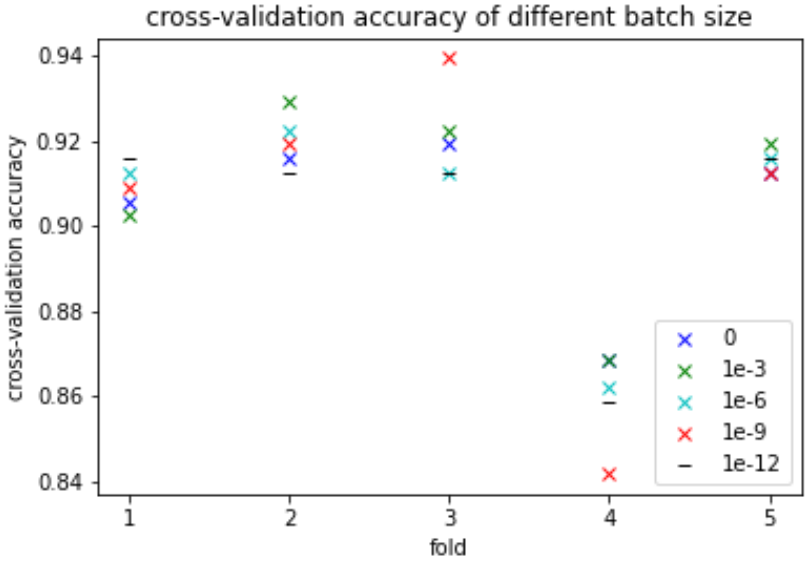
Part A Question 4

Cross-validation accuracies



The cross-validation accuracies have been scaled to the same y axis for a better comparison between the accuracies. The accuracies are generally consistent despite the changing decay parameter. Decay parameter of 0 provides the most stable accuracy, while decay parameter of 10*-9 has the most drastic range.

Select the optimal decay parameter

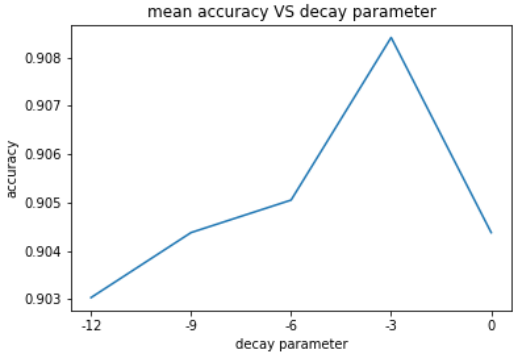


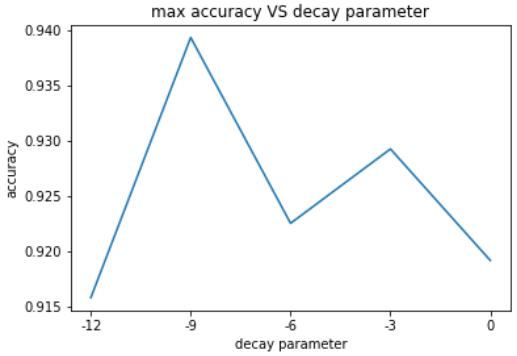
The cross-validation accuracy of each folds provides a more distinct visual of the accuracies across decay parameter. The table below charts the accuracies from the graph.

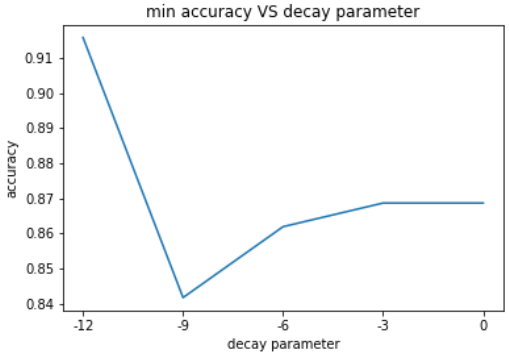
	accuracies(most)				accuracies(least)
Fold1	-12	-6	-9	0	-3

Fold 2	-3	-6	-9	0	-12
Fold 3	-9	-3	0	-12/-6	-12/-6
Fold 4	-3/0	-3/0	-6	-12	-9
Fold 5	-3	-6/-12	-6/-12	-9/0	-9/0

It is difficult to determine the accuracies from both the diagram, and the table, as the accuracies of each decay parameter is split for each fold.

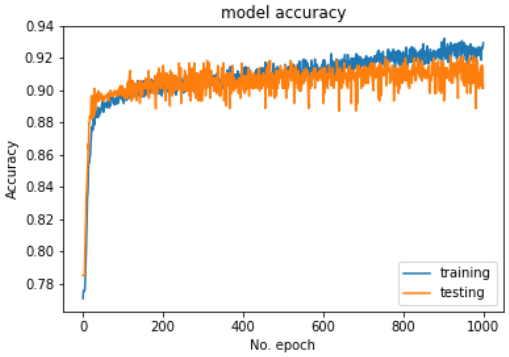


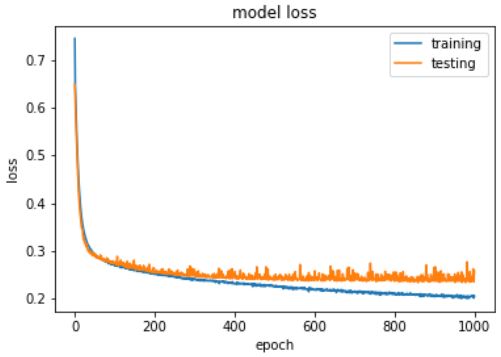




The mean, max and min accuracies for each decay parameter are not equal either. Hence starts the process of elimination. Firstly, -12 is removed as it has the lowest accuracies for mean and max. Secondly, 0 is removed as while it is constant, the accuracies are low. Next up, -9 is removed due to its sudden drop for min. This leaves -6 and -3. Since the accuracies for -3 is constantly higher than -6, **-3 is chosen as the optimal decay parameter.**

A decay parameter of -3 is thus used to replot the accuracies against the 5-fold cross validation method.





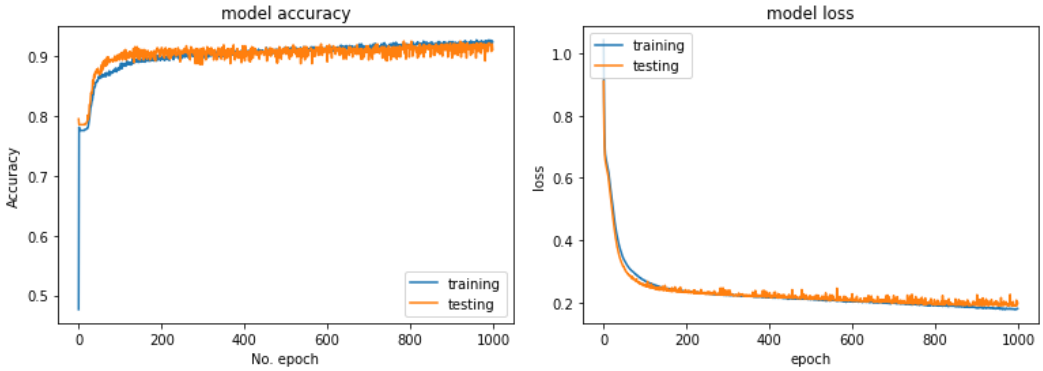
Training accuracy rate= 92.94%

Testing accuracy rate= 90.12%

The testing results fluctuates more than the training result. The difference between testing and training increases with the epochs. However, the testing loss is still on a decreasing slope, indicating that it is not overfitting

Part A Question 5

4-layer network



Training accuracy rate= 92.33%

Testing accuracy rate= 91.06%

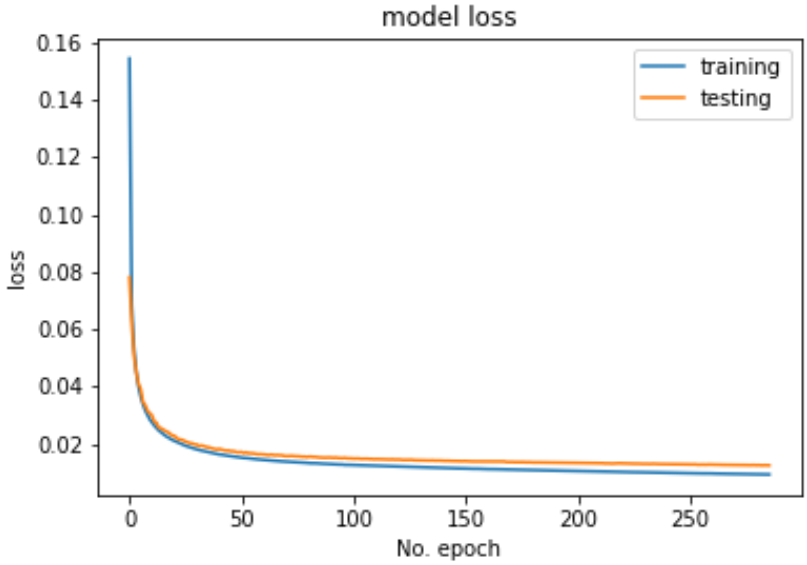
Compare and comment on the performances of the optimal 3-layer and 4-layer networks

The network provides similar accuracies. Training accuracy for the 3-layer network is higher, while testing accuracy for 4-layer network is higher. The testing accuracies for the 4-layer network fluctuates less than the 3-layer network.

However, both networks have similar shapes. For the accuracy, the testing accuracies overtake the training accuracies at the start and drops below it. For the loss, the testing accuracies was lower than the testing accuracies at the start, and then it overtook the training accuracies.

Part B Question 1

Training dataset



The train errors converge around 100 epochs.

To see when the test error is minimum, an early stopping function has been implemented, with a patience of 10. **The training has stopped at epoch 286, which when the test error is minimum**

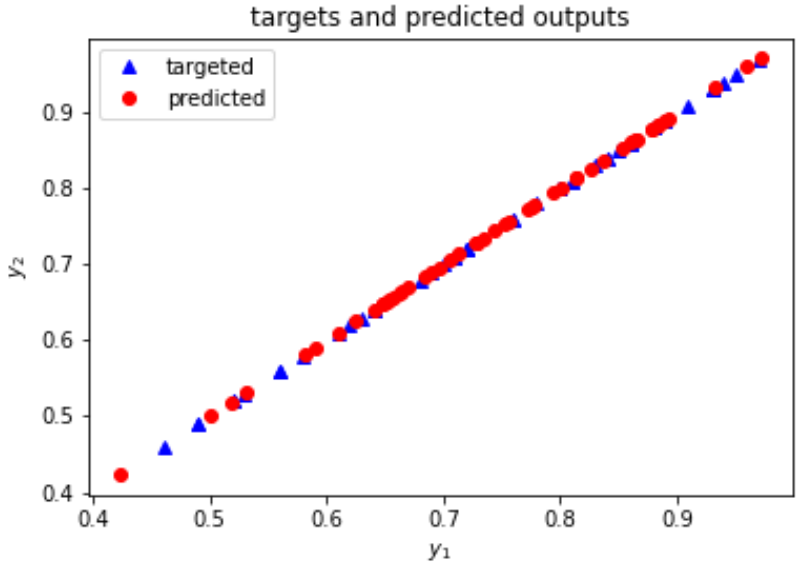
Train m.s.e. = 0.9%

Test m.s.e. = 1.3%

Plot the predicted values

Target Y value	Predicted Y value
[[0.94] [0.85] [0.63] [0.7] [0.52] [0.8] [0.8] [0.68] [0.86] [0.93]	[[0.932215] [0.7765496] [0.728181] [0.6537486] [0.5907607] [0.8250681] [0.778924] [0.6562734] [0.8593117] [0.8914718]

The first 50 test data have been used to plot the predicted values against the target values. The table shows the first 10 values, the target, and its predicted value. The accuracies of the target and predicted varies from 0.01% to 0.09%, at least for the first 10 values.

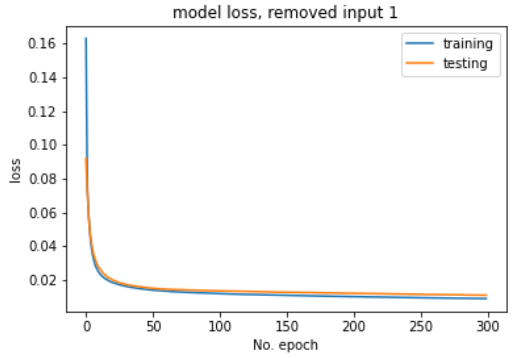


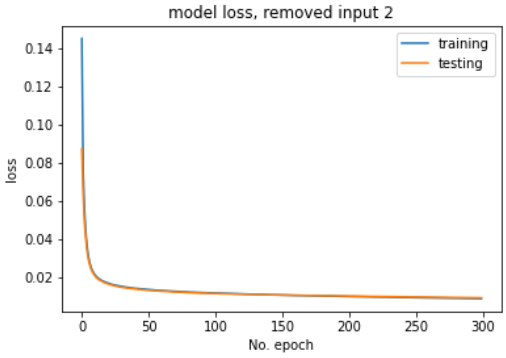
This graph shows that there is not a significant change in the value.

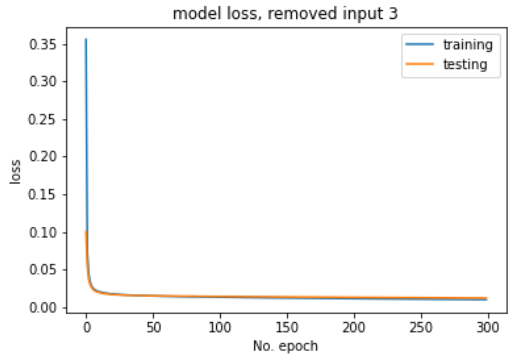
Part B Question 2

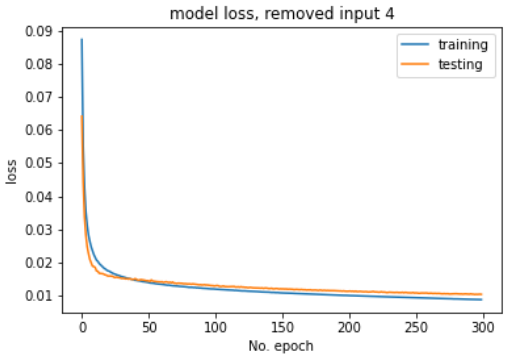
6 Input features

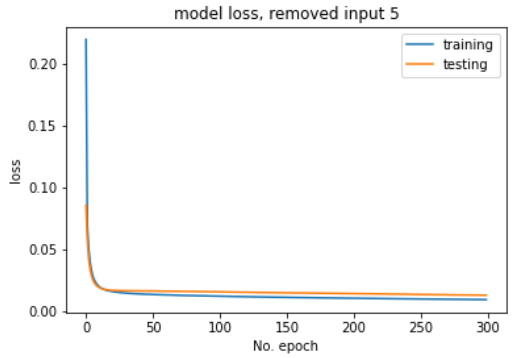
6 models have been trained, each with a different input deleted from the initial dataset.

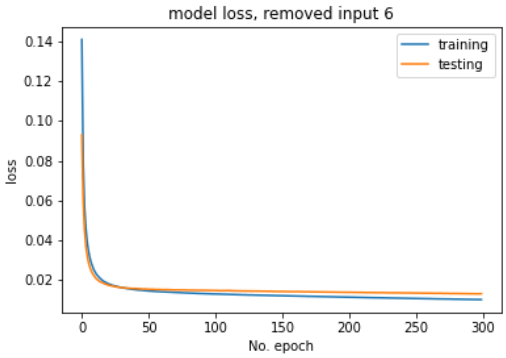


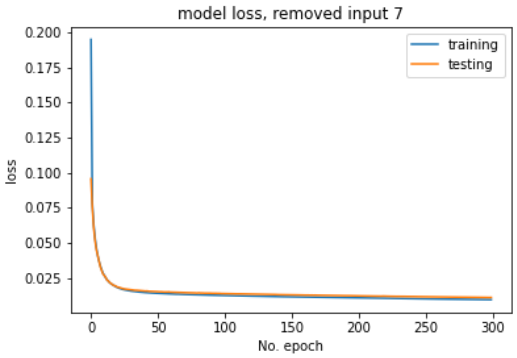












input 1 removed: train m.s.e. = 0.9%, test m.s.e. = 1.1%

input 2 removed: train m.s.e. = 0.9%, test m.s.e. = 1.0%

input 3 removed: train m.s.e. = 1%, test m.s.e. = 1.2%

input 4 removed: train m.s.e. = 0.9%, test m.s.e. = 1.0%

input 5 removed: train m.s.e. = 0.9%, test m.s.e. = 1.3%

input 6 removed: train m.s.e. = 1.0%, test m.s.e. = 1.3%

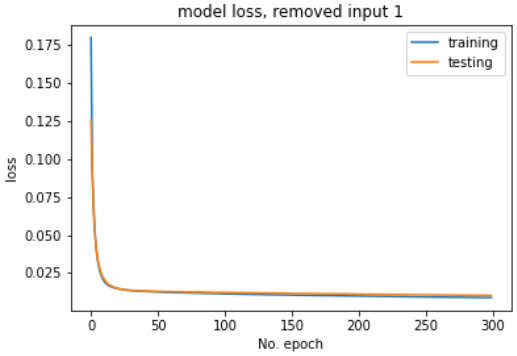
input 7 removed: train m.s.e. = 1.0%, test m.s.e. = 1.1%

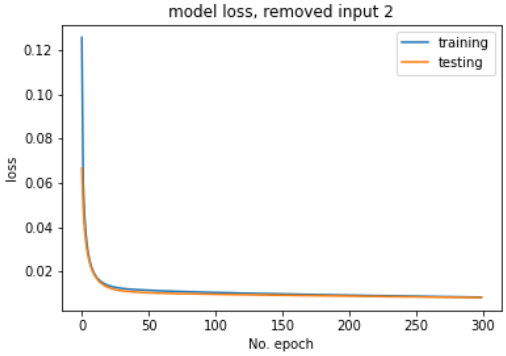
Input 2 and 4 provide the least train and test m.s.e, train m.s.e. = 0.9%, test m.s.e. = 1.0%.

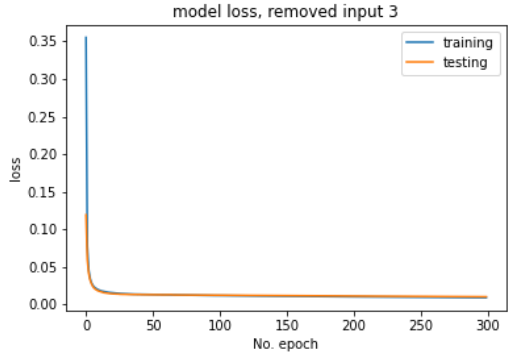
Input 4 is chosen to be removed as its loss are less consistent than input 1's. The Statement of Purpose has been removed from the dataset

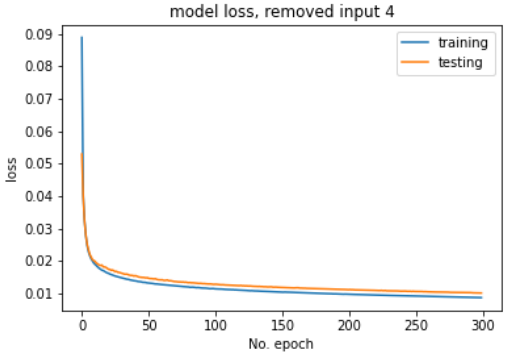
5 input features

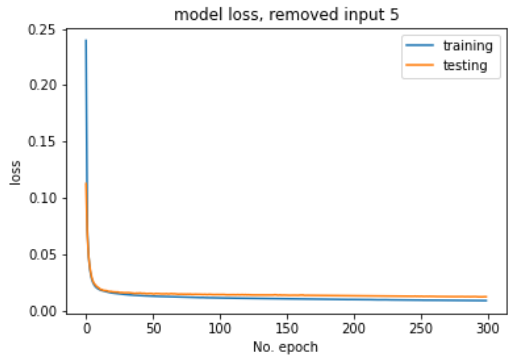
5 models have been trained, each with a different input deleted from the (initial dataset without the 4th input).

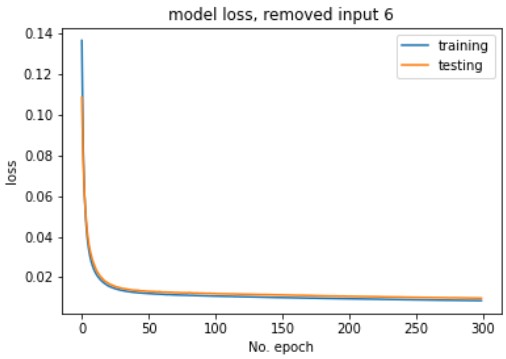












input 1 removed: train m.s.e. = 0.9%, test m.s.e. = 1.0%

input 2 removed: train m.s.e. = 0.8%, test m.s.e. = 0.8%

input 3 removed: train m.s.e. = 0.9%, test m.s.e. = 1.0%

input 4 removed: train m.s.e. = 0.9%, test m.s.e. = 1.0%

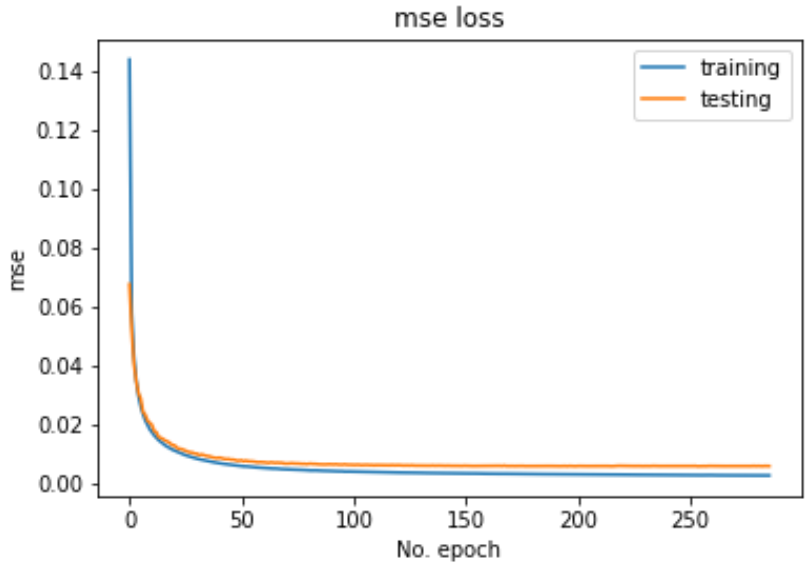
input 5 removed: train m.s.e. = 0.9%, test m.s.e. = 1.2%

input 6 removed: train m.s.e. = 0.9%, test m.s.e. = 1.0%

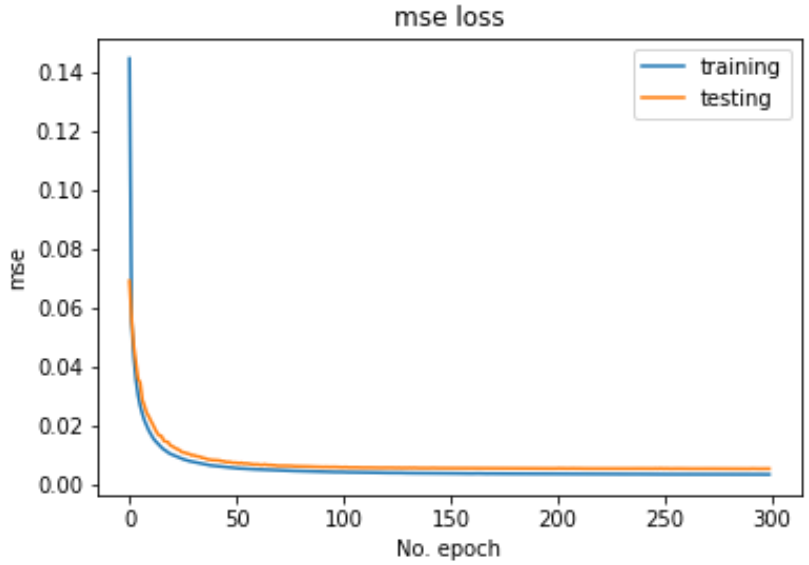
Input 2 provides the least train and test m.s.e, train m.s.e. = 0.8%, test m.s.e. = 0.8%.

Input 2 is chosen to be removed. The TOEFL score has been removed.

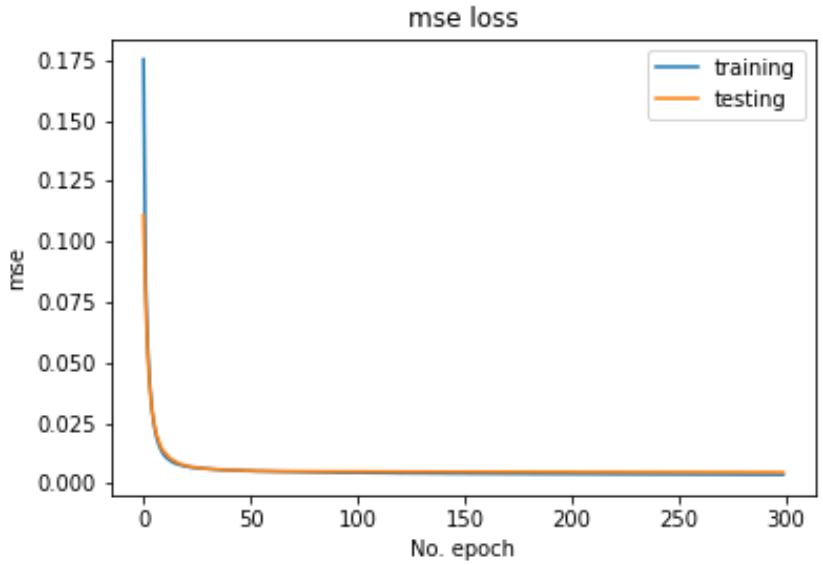
Compare the accuracy of the model with all input features, with models using 6 input features, and 5 input features



m.s.e. loss for 7 input features: train m.s.e. = 0.009, test m.s.e. = 0.013



m.s.e. loss for 6 input features: train m.s.e. = 0.009, test m.s.e. = 0.011



m.s.e loss for 5 input features: train m.s.e. = 0.009, test m.s.e. = 0.010

The test m.s.e decreases with each loss of input features, while the train m.s.e. increases. This is fair to say that **the accuracy has increased with each decrease in input features**.

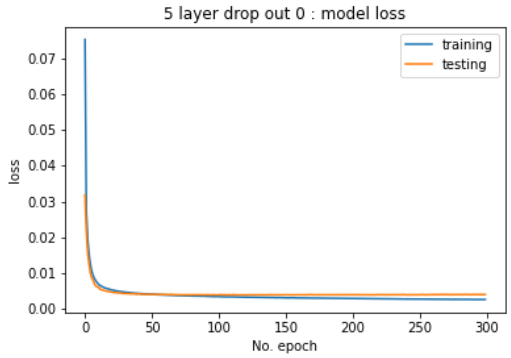
When a prediction of the test data is made, the following value have been returned.

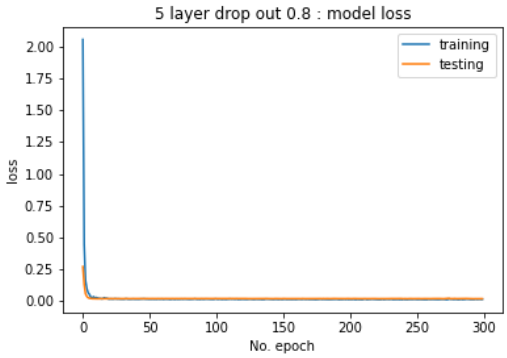
Target	Predicted: All	Predicted: 6	Predicted: 5
[[0.94] [0.85] [0.63] [0.7] [0.52] [0.8] [0.8] [0.68] [0.86] [0.93]]	[[0.932215] [0.7765496] [0.728181] [0.6537486] [0.5907607] [0.8250681] [0.778924] [0.6562734] [0.8593117] [0.8914718]]	[[0.93290126] [0.88395953] [0.700374] [0.73419595] [0.55875516] [0.8299297] [0.7113184] [0.7097521] [0.8550215] [0.9091114]]	[[0.92436653] [0.82377774] [0.7045614] [0.6591988] [0.56267935] [0.824414] [0.7490077] [0.629107] [0.80707645] [0.88673824]]
Mean square error	0.0061261719663334 39	0.0051002633193 89378	0.0046928478912 10916

Due to the short sample of data in the table, the increase in accuracies in the prediction cannot be seen. However, from the mean squared error of the prediction and targeted, **the accuracies have increased when the input features drop.**

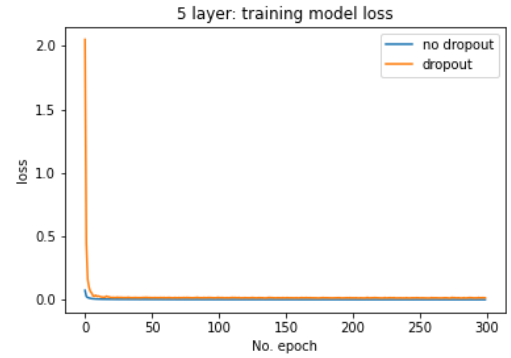
Part B Question 3

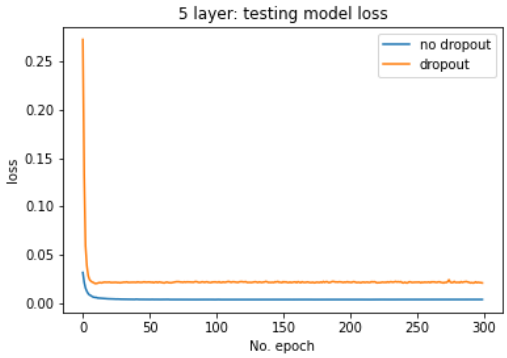
5 Layer network:





The network with dropout provides a more coherent error between its' testing and training result. Both networks converge. The network with dropout converges earlier, around 25, while the network without converges around 100.





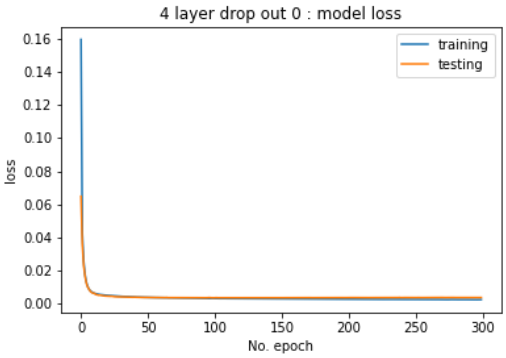
Without dropout: train m.s.e. = 0.3%, test m.s.e = 0.4%

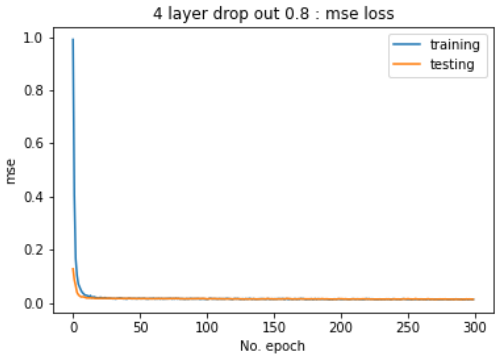
With dropout: train m.s.e. = 1.7%, test m.s.e = 2.1%

The network with **dropout has a significant increase in test error, resulting in lower accuracy.**

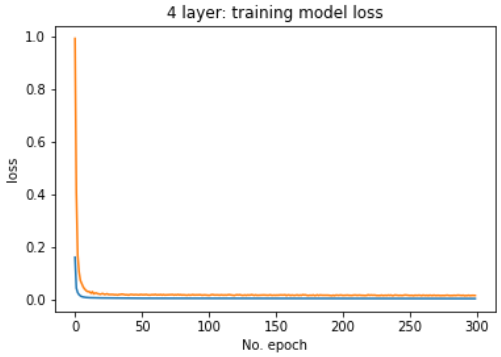
The networks both converges around the same epoch with and without dropout.

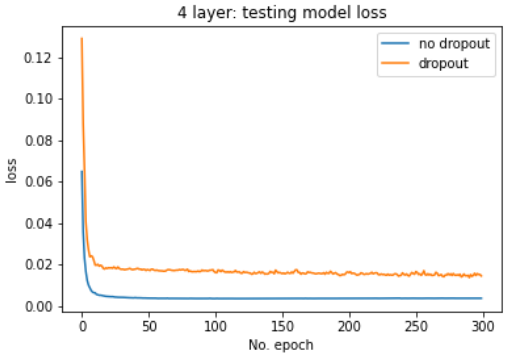
4 Layer network:





Both networks have a coherent error between its' testing and training result. Both networks converge. The network with dropout converges earlier, around 25, while the network without converges around 75.





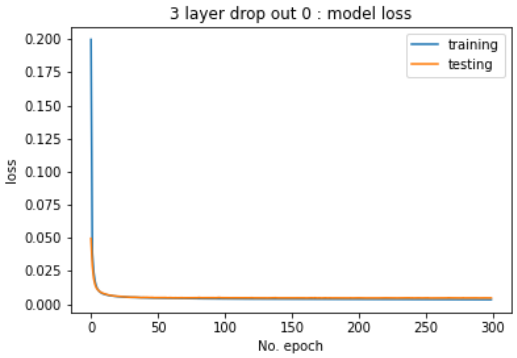
Without dropout: train m.s.e. = 0.3%, test m.s.e = 0.4%

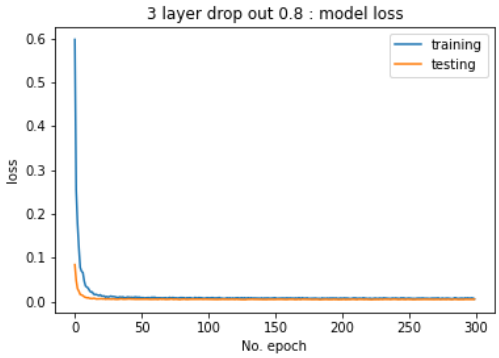
With dropout: train m.s.e. = 0.7%, test m.s.e = 1.5%

The layer with the **dropout has a higher error, resulting in lower accuracy**.

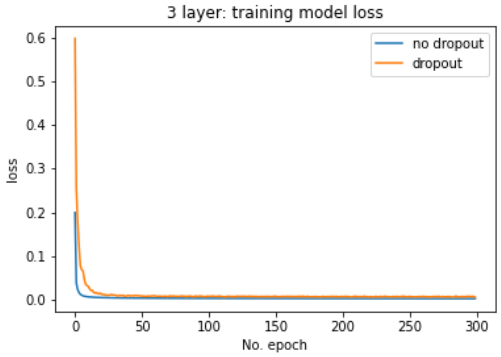
The networks both converges around the same epoch with and without dropout. However, for the testing error, the dropout network decreases at a faster rate than the network without dropout

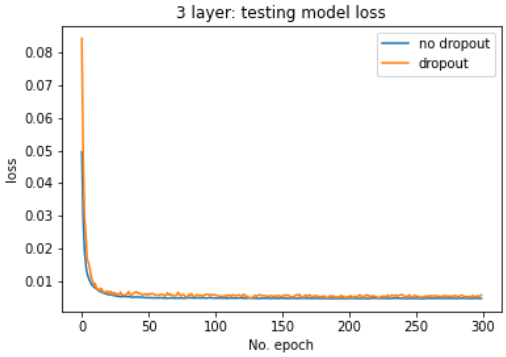
3 Layer network:





Both networks have a coherent error between its' testing and training result. Both networks converge. The network with dropout converges earlier, around 25, while the network without converges around 50.





The networks both converges around the same epoch with and without dropout. The networks are of similar errors, with and without dropout.

Without dropout: train m.s.e. = 0.3%, test m.s.e = 0.5%

With dropout: train m.s.e. = 0.7%, test m.s.e = 0.6%

The layer with the **dropout has a higher error, resulting in lower accuracy.**

Compare the performances of all the network

	5-layer		4-layer		3-layer	
	no dropout	dropout	no dropout	dropout	no dropout	dropout
testing error	converges around 100	converges around 25	converges around 75	converges around 25	converges around 25	converges around 25
train m.s.e	0.3	1.7	0.3	0.7	0.3	0.7
test m.s.e	0.4	2.1	0.4	1.7	0.5	0.6

When there is no dropout, there is minimum changes between the different layers.

When there is dropout, the dropout's error decreases with the reduction of layers. A smaller network has a higher accuracy than a more complex layer.

The error is lower when there is no dropout factor introduced. Hence, a no dropout network layer is preferred.

The testing error is the minimum at a 5-layer/4-layer network with no dropout.

However, the difference between the error (0.1) is too small to say for certainty that a 5-layer/4-layer no dropout network is better than a 3-layer no dropout network.