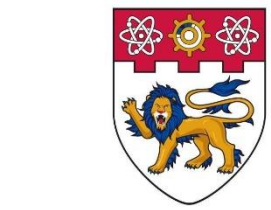


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

AY2021 Semester 1

CE4042

Neural Networks and Deep Learning

Assignment 2

Name: Raymond Toh

Matric Number: U1821599H

Tutor: Chen Change Loy

1 Table of Contents

2 Introduction

The goal of assignment two is to allow us to have a better understanding of how the hyperparameters affect Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN).

Part A of assignment two is to build a CNN for object recognition and obtain the optimal number channels for convolution layers by using a grid search method. The optimal number of channels will be used and compared with different optimizers. The dataset used to train the model is CIFAR-10 dataset with 10 labels and 32x32x3 input attributes.

Part B of assignment two is to compare RNN with CNN to classify the first paragraphs from Wikipage entries with 15 labels. Different types of RNN architectures will be used and compared against each other.

3 Methods

The following methods are used throughout this assignment.

3.1 Dropout Regularization

Dropouts allow us to set the probability of eliminating neurons during each training step. This will reduce the overfitting problem of the model.

3.2 Gradient Clipping

Gradient Clipping solves the problem of exploding and vanishing gradients from the backpropagation through time.

3.3 Grid Search

Grid search is an approach to tune hyperparameters and allows the model evaluation for a combination of specified values. For this assignment, it will be used on the number of channels.

4 Experiment & Results

4.1 Part A

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

1. 32x32x3 Input Layer
2. 1st Convolutional layer with 50 channels, valid padding, a ReLU activation and window size of 9x9.
3. 1st Max Pooling layer with VALID padding, window size of 2x2 and stride of 2.
4. 2nd Convolutional layer of 60 channels, VALID padding, ReLU activation and window size of 5x5.
5. 2nd Max Pooling layer with VALID padding, window size of 2x2.

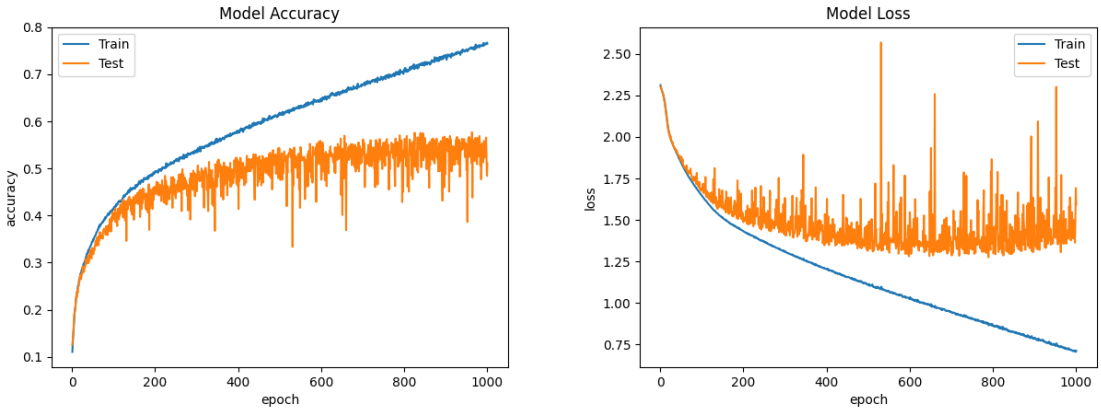
6. 1st fully connected layer of size 300 with no activation function.

7. 2nd fully connected layer of size 10 with SoftMax activation function.

Part A Q1 Train CNN Network

a

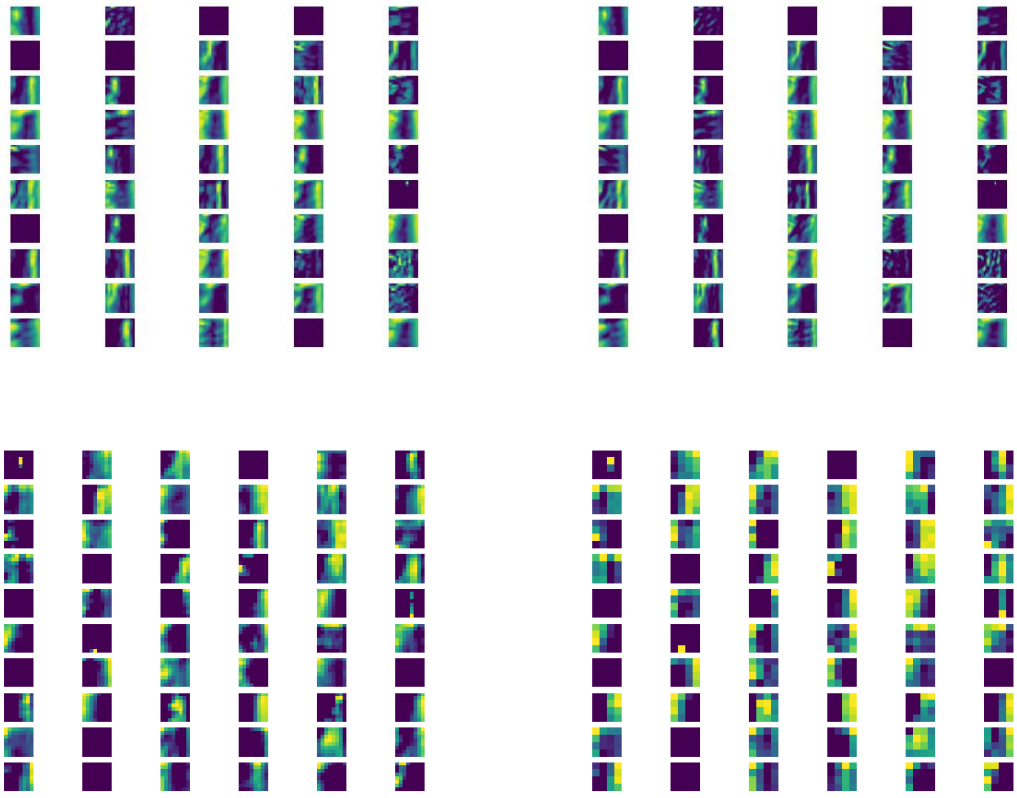
The network is trained with the above parameters over 1,000 epochs. The accuracies for training and test datasets are shown below along with their costs.



b

The first 2 images from the test data will be used to generate feature maps after the 2 convolutional and max-pooling layers.

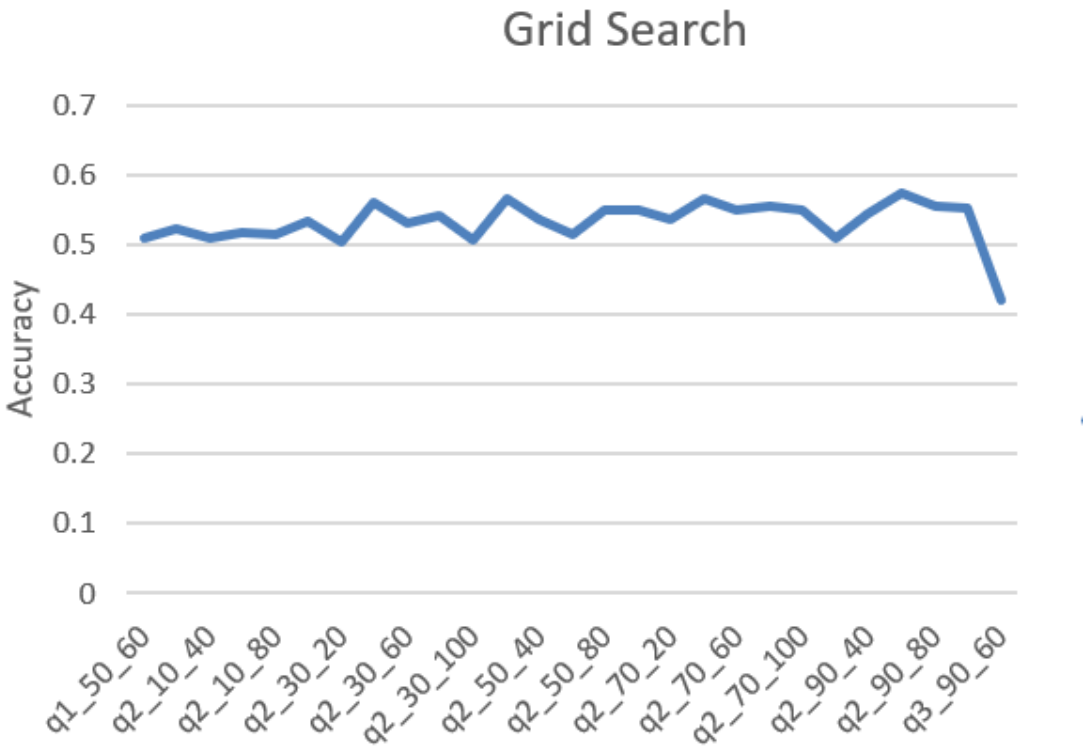

Test Image 1




Test Image 2

Part A Q2 Grid Search

Using a grid search method, the number of channels with the best performance will now be found. There will be a total of 25 combinations for both the first and second convolutional layer. A line graph of the 25 combinations are shown below.

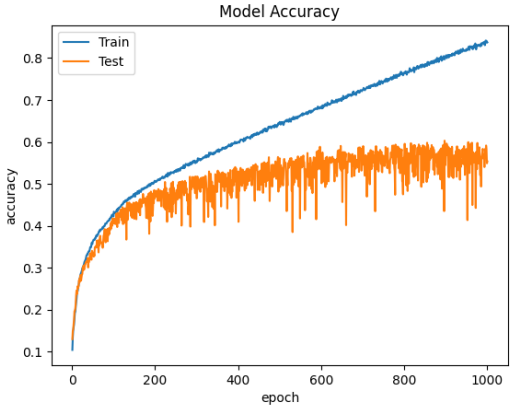


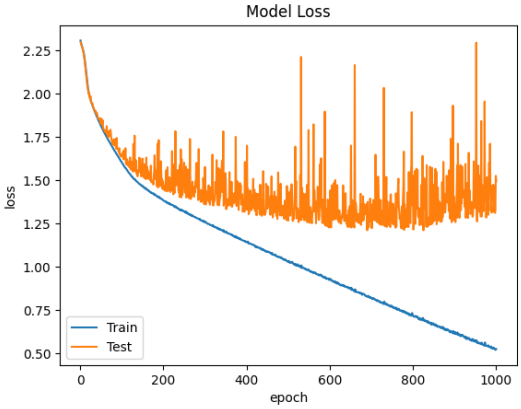
Part A Q3 Change in Optimizer

With the selected number of channels to be used. Different optimizers will be tested to see which one will work best to classify the data. The parameters will be tested independently.

a Momentum

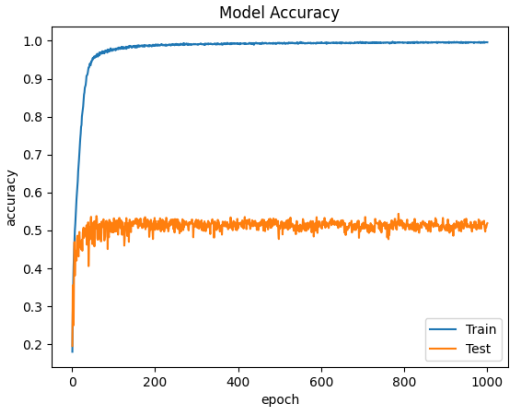
A momentum of 0.1 is used on the model and the result from the accuracy and loss function on the test data is shown below.

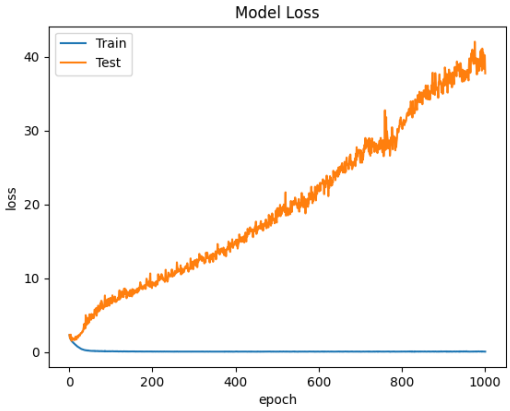




b RMSProp

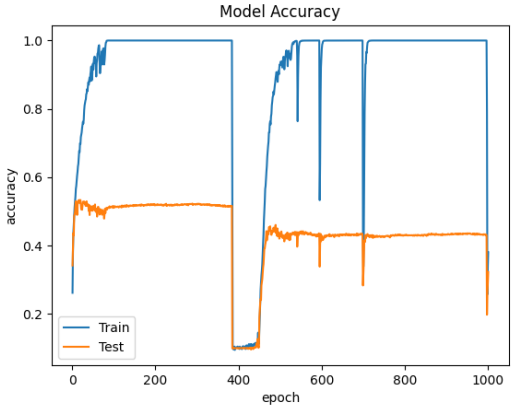
Next, the RMSProp optimizer is used in place of the SGD model. The respective accuracy and loss graphs are shown below.

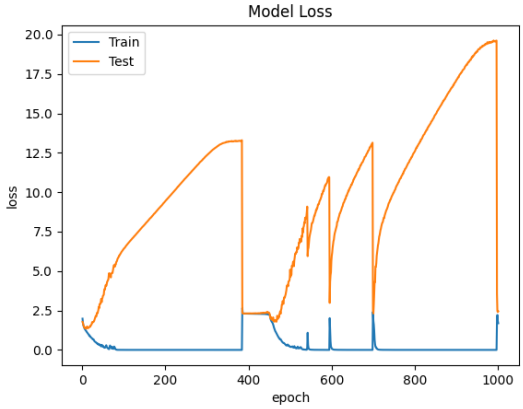




c Adam

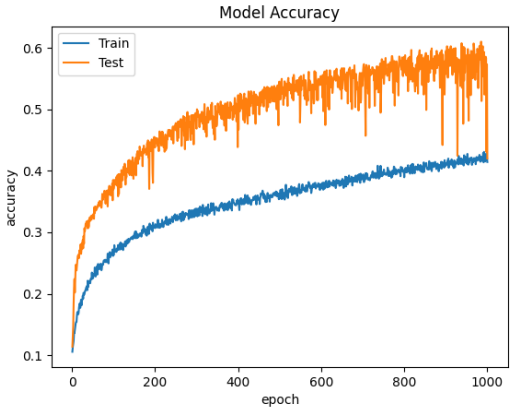
The Adam optimizer will now be used to test and see if the model performs better. Their respective graphs are shown below.

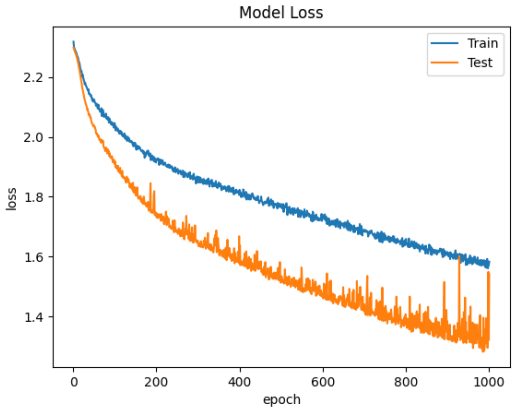




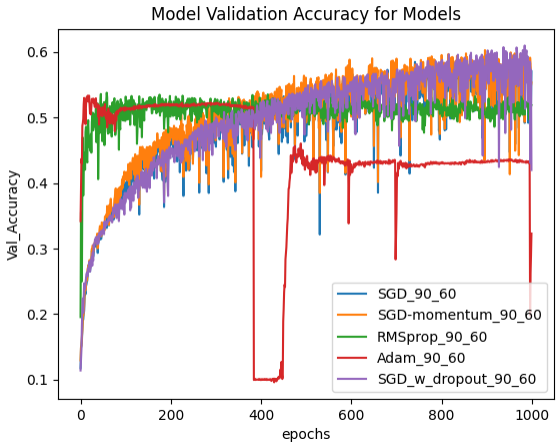
d Dropouts

Dropouts will now be enabled on the SGD model.





Part A Q4 Comparison of Algorithms



After changing all the parameters, their results are compared. The overall accuracy was not very high due to the fact that the image resolution is not high and also

having a limited number of data. If higher resolution images and increase the training dataset were given, the accuracy would improve.

4.2 Part B

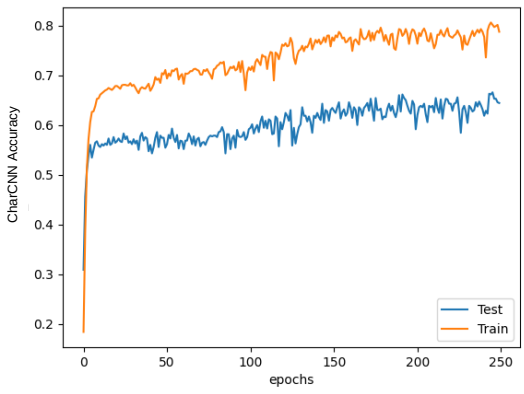
Text classification will now be performed. There are 15 classes for the text to be classified under. Both CNN and RNN networks will be tested.

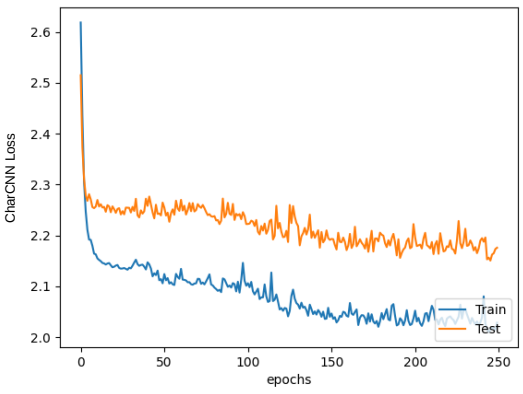
In this object recognition problem, we will design a model with these few steps:

1. 1st Convolutional layer with 10 filters, 20x256 window size, VALID padding and a ReLU activation function.
2. 1st Max Pooling layer with SAME padding, window size of 4x4 and stride of 2.
3. 2nd Convolutional layer of 10 filters, VALID padding, ReLU activation and window size of 20x1.
4. 2nd Max Pooling layer with SAME padding, window size of 4x4 and stride of 2.

Part B Q1 Character CNN Model

With the model design above, the model is now trained and tested. This model is designed to receive character ids for classification. The accuracy and loss graph is shown below.



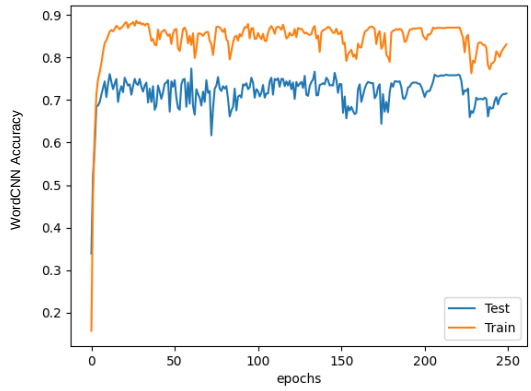


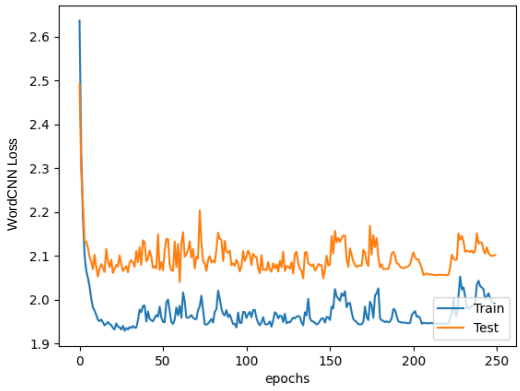
Part B Q2 Word CNN Model

A similar model from above is used with the model now taking in word ids to be classified. Their results are shown below.

1. 1st Convolutional layer with 10 filters, 20x20 window size, VALID padding and a ReLU activation function.
2. 1st Max Pooling layer with SAME padding, window size of 4x4 and stride of 2.
3. 2nd Convolutional layer of 10 filters, VALID padding, ReLU activation and window size of 20x1.
4. 2nd Max Pooling layer with SAME padding, window size of 4x4 and stride of 2.

This CNN model will be reading the word ids and the test accuracy and entropy plots are shown below.

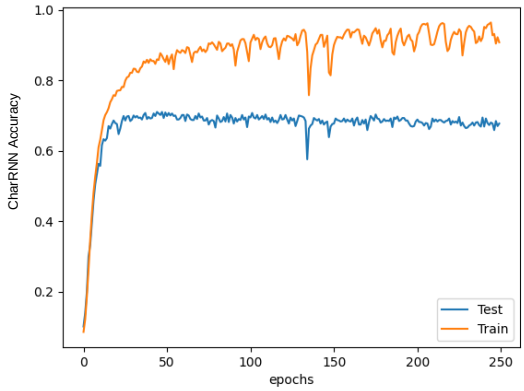


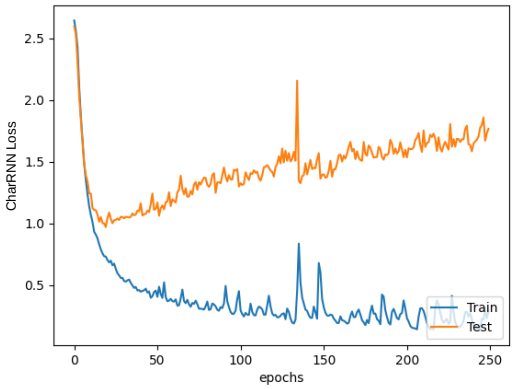


Part B Q3 Character RNN Model

The RNN model will now be designed that classifies text on a character level. The RNN will have a hidden layer of size 20.

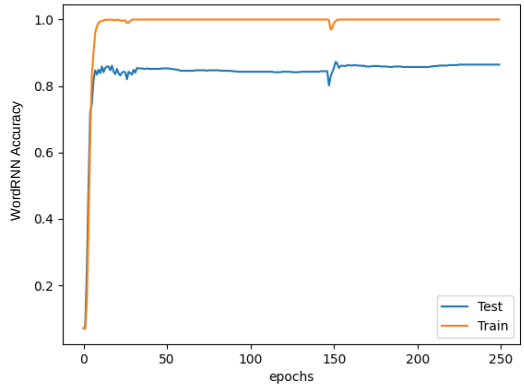
The test accuracy plots along with the entropy cost are shown below.

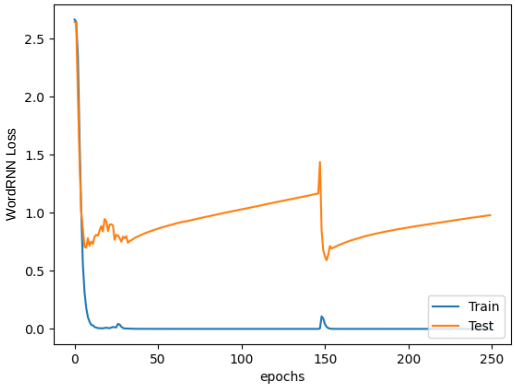




Part B Q4 Word RNN Model

A similar RNN is now designed to read on a word level. The RNN will have a hidden layer of size 20 and the inputs will be passed through an embedding layer of size 20 before being fed into the network. The test accuracy plots along with the entropy cost are shown below.

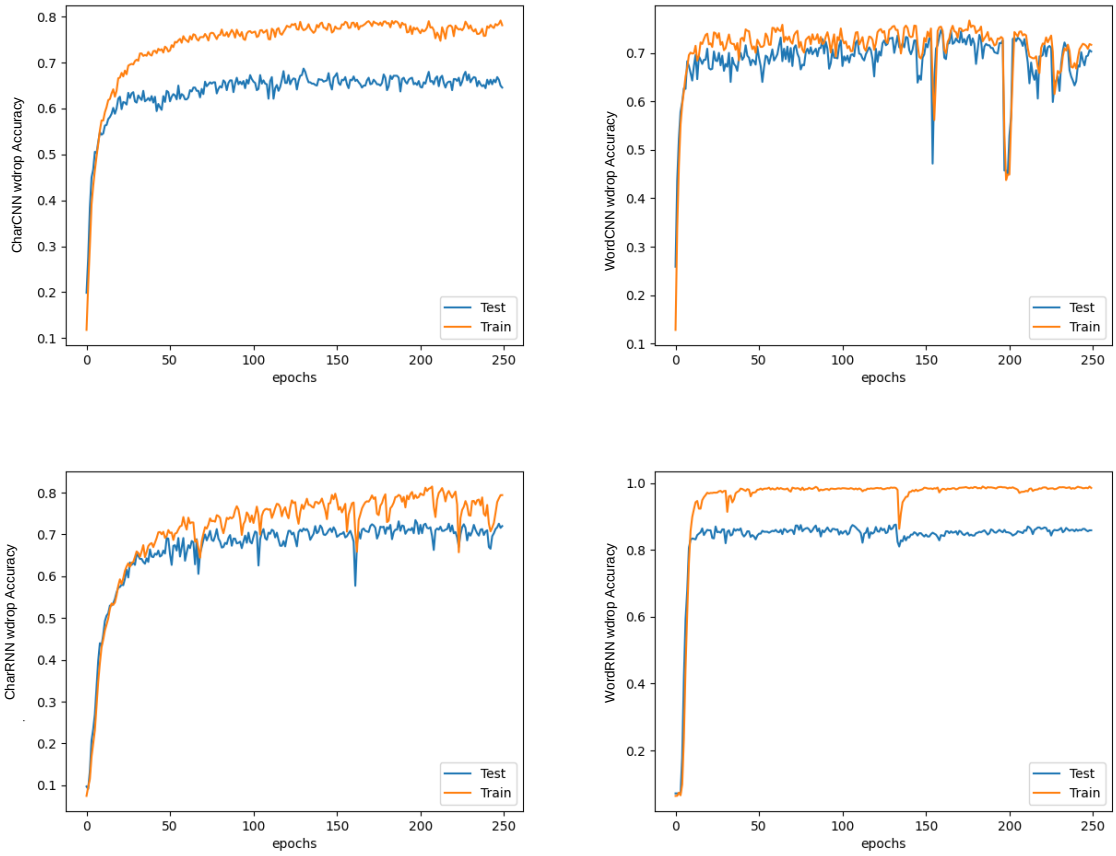




Part B Q5 Adding dropouts into the model

Dropouts will now be added into the model. Their performance will be tested on the 4 models above. The dropout version will be evaluated against the non-dropout version and their results shown below.

The network runtime was based on the parameters being trained. Hence, time needed would be increasing in the manner of CNNWord, CNNChar, RNNWord, RNNChar.



By adding dropout, we can see that as compared to part 1 to 4, the overfitting problem have been reduced. The most instinct model that can see this result is WordCNN. Also, in order to improve further from these results, we can get a bigger dataset to train so the model able to generalize better.

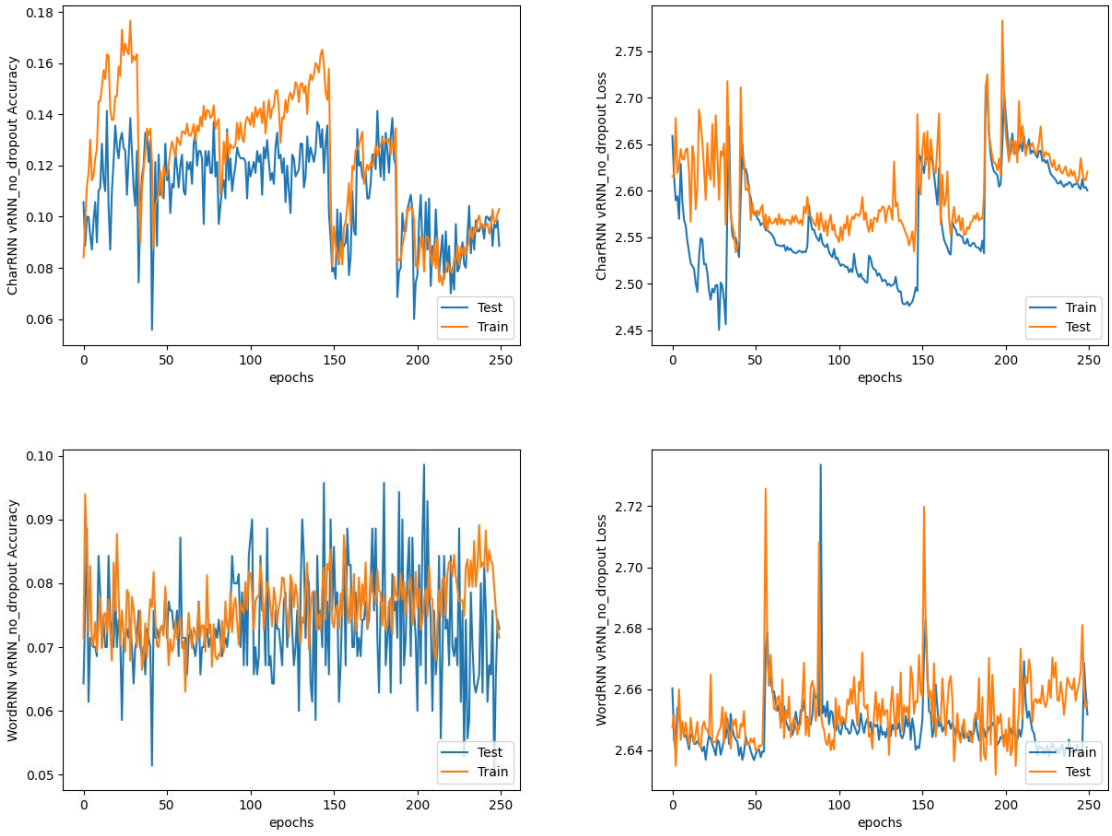
Part B Q6 RNN Improvements

The RNN will now be improved with the parameters and tested independently.

1. a. Replacing GRU layer with Vanilla layer
   b. Replacing GRU layer with LTSM layer
2. Increasing number of RNN layers to 2
3. Introducing gradient clipping with threshold of 2

Part B Q6ai Vanilla layer

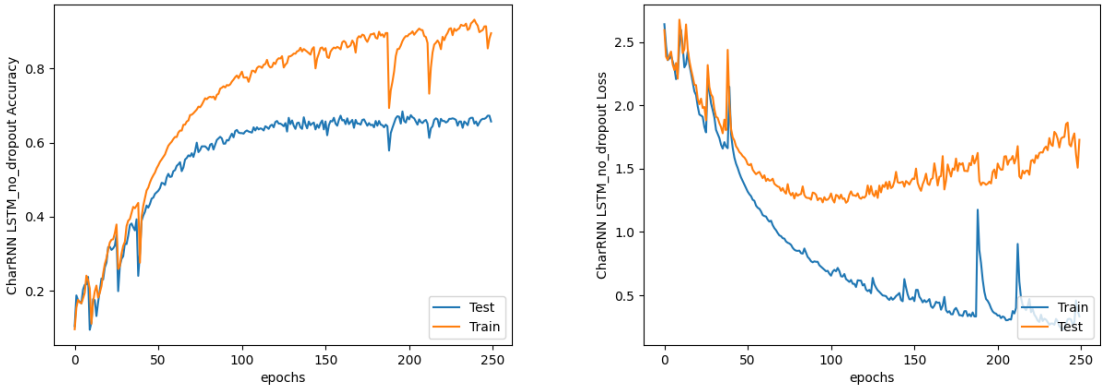
Both character and word models will now have their GRU layer replaced with a pure vanilla layer and their test accuracy and loss graphs are shown below.

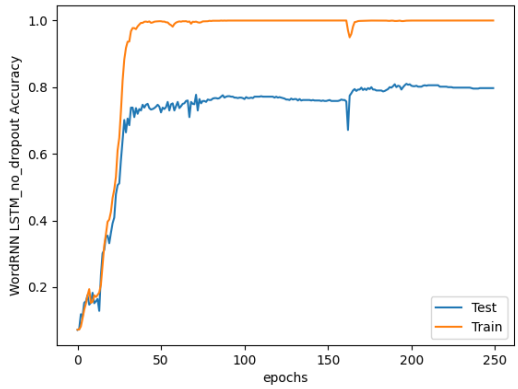


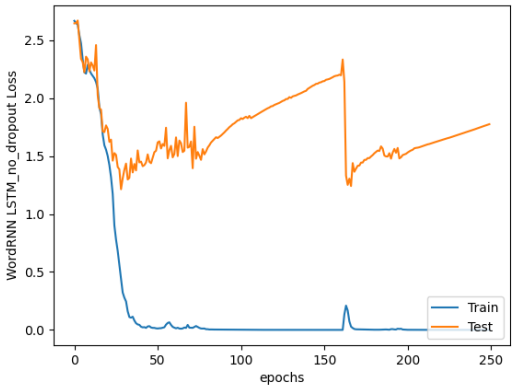
By replacing GRU with vanilla RNN, the model validation accuracy drop from 70% to estimated 15%. We can see that by using vanilla RNN, the model will not generalize well.

Part B Q6aii LTSM layer

Both the character and word RNN models will now have a LTSM layer and their test accuracy and loss results are shown below.



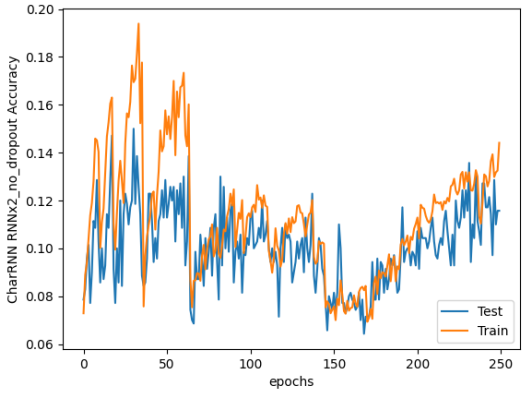


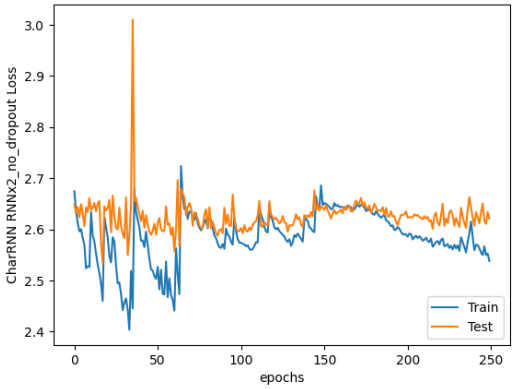


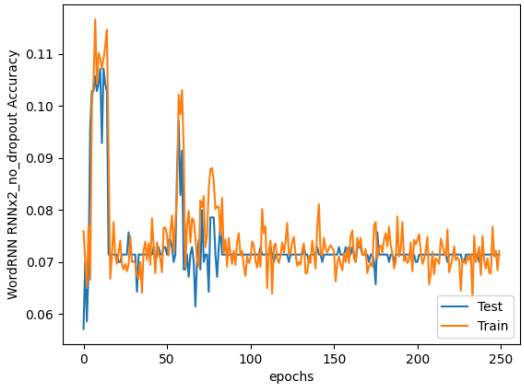
Althought GRU and LSTM are similar to each other, the accuracy for GRU is higher as compared to LSTM model.

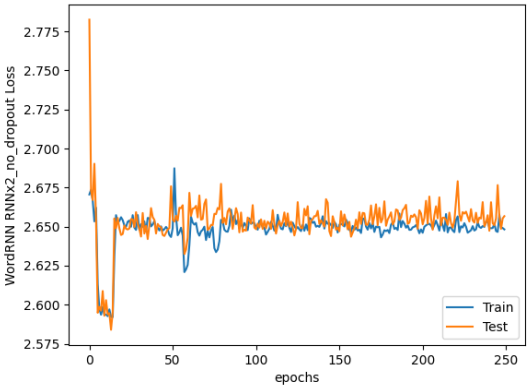
Part B Q6b Increasing RNN layers

The RNN layers will now be increased to 2 and their test accuracy and loss results are shown below.





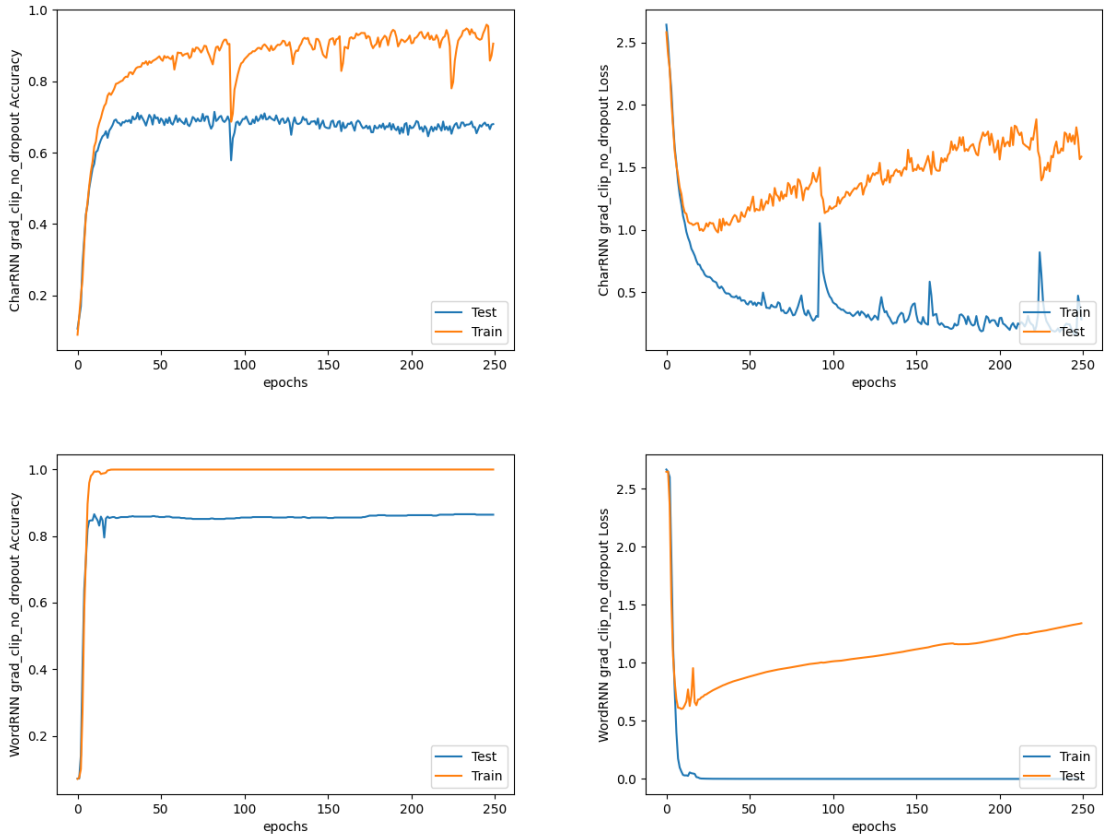




The results of 2 vanilla RNN and one vanilla RNN are the similar. However, looking at the loss of WordRNN, we notice that the model was not learning. Hence, the accuracy will not increase.

Part B Q6c Gradient Clipping

Gradient clipping is introduced to the model. A clipping threshold of 2 is used on the model. The test accuracy and loss results are shown below.



We know that gradient clipping is often used to solve the exploding or vanishing gradient due to backpropagation through time. However, in this part, it does not differ from original GRU model as the gradient is always within the range. Hence, the output accuracy of the model is similar.

5 Conclusion

After the assignment, a better knowledge of CNN and RNN was learnt. The use of parameters such as dropouts, gradient clipping and the use of different optimizers showed how a Neural Network can be tweaked to perform better. It is good to tweak a network based on the desired result and data provided.

In Part A, the use of a grid search is implemented to find the ideal number of channels in the 2 convolutional layers. It is important to use this as this can test every possible scenario. Different optimizers were also tested as some models perform better given the dataset. Test accuracies were not high, this could be because the resolution of the images was very low.

For Part B, the text being classified were trained on CNN and RNN. Their performance was evaluated, and CNN performed as good as RNN. However, the preprocessing of data is different. CNN uses embedded layers to convert text into values, while RNN uses one-hot vector and encode its vocabs. Lastly, to aim of improving performances, few hyperparameters have used for evaluation which is dropouts, use of a vanilla layer, increasing RNN layers and use of gradient clipping was used. Gradient clipping provided the best results.