PRASANNA, LAKSITA
3/5/25

DEEP LEARNING HW 3

In this homework assignment, sentiment classification was implemented on the Twitter US Airline Sentiment data using Recurrent Neural Networks (RNNs). Since the OneHotDataGenerator method generated the dataset in batches, the entire dataset was used.

Problem 1 (simpleRNN)

Simple RNNs were used for sentiment classification. Models with various hyperparameters were experimented with to find the model with the best prediction accuracy.

Some parameters that were kept consistent among the models are the total number of epochs (20), batch_size(32), and max_length(30).

The graphs below show the Training and Validation accuracies for each model.

Different Learning Rates

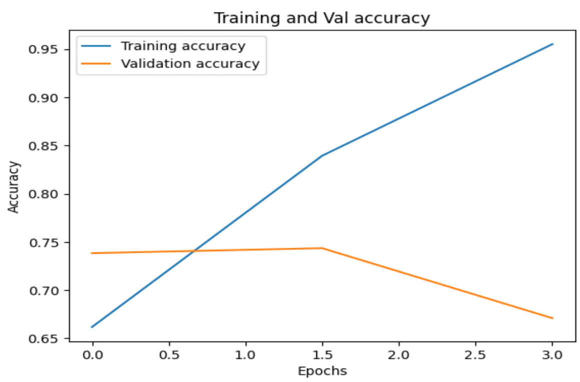


Figure 1a) Higher Learning rate (LR=0.001)

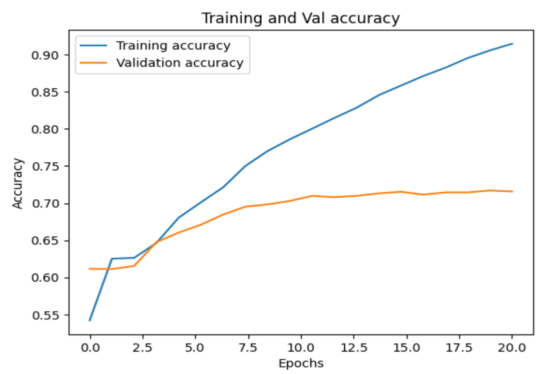


Figure 1b) Lower learning rate (LR = 0.00001)

Learning rates indicate the size of steps taken towards lowest gradient directing the loss function to a global minimum. The disadvantage of using a higher learning rate is that the model starts overfitting much earlier compared to the lower learning rate model.

Overfitting: Validation and test accuracies are less than training accuracies for both models which indicates overfitting. It is evident from Figure 1b that the lower learning rate model only overfits after 10 epochs whereas the higher learning rate model almost immediately starts overfitting in Figure 1a.

Generalizable: According to Table 1, the test accuracy of the lower learning rate model was 0.7254 which is significantly better than the higher learning rate. Thus, the lower learning rate model was more generalizable and was the better option out of these two models.

Experimenting with RNN layers with different units and more Dense layers

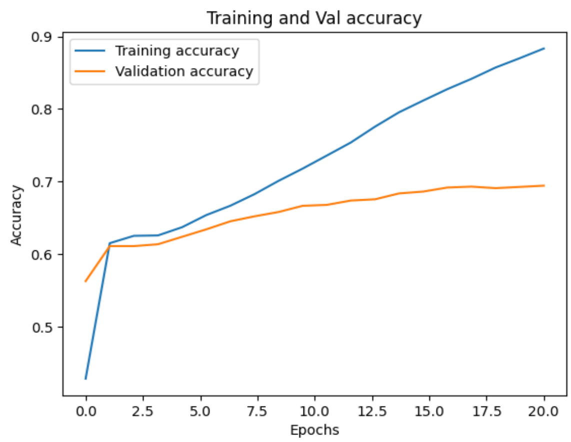


Figure 1c) RNN with fewer units

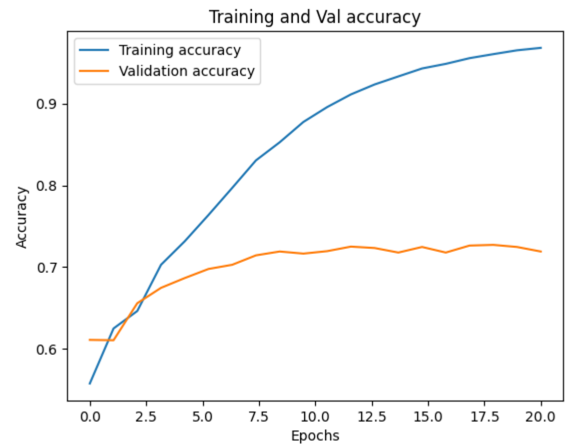


Figure 1d) RNN with more units

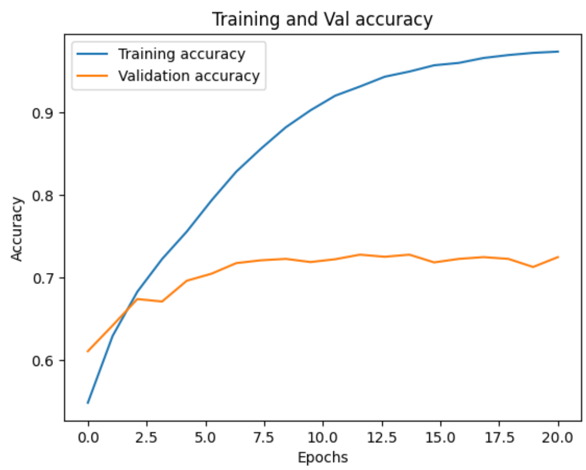


Figure 1e) Adding more Dense Layers

RNN with fewer units: 32, 32, 16, 16

RNN with more units: 128, 64, 32, 16

Dense layers: 256, 128, 64, 3 nodes

Overfitting: Validation and test accuracies are less than training accuracies for all 3 models which indicates overfitting. RNN with more units overfits at epoch 7, RNN with fewer units overfits at epoch 5, and 'More Dense layers' model overfits at epoch 7.5 as seen in Figures 1c, d, e. Increasing the number of units and nodes for the RNN layers and the Dense layers respectively increased the capacity of the models to learn more patterns. The difference in the training and validation accuracies for the two models was almost 0.1 or 0.2 as per Table 1 which indicates how similar they performed.

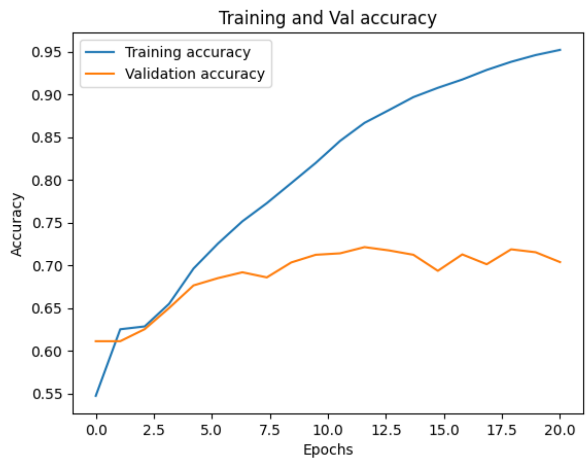
Generalizable: RNN with more units had the higher test accuracy 0.7551 of these three models therefore it is more generalizable. The runner up was the 'More Dense layers' models with a test accuracy of 0.7470.

PRASANNA, LAKSITA
3/5/25

DEEP LEARNING HW 3

Use of regularizers and dropouts was to minimize overfitting.

Different Regularizers



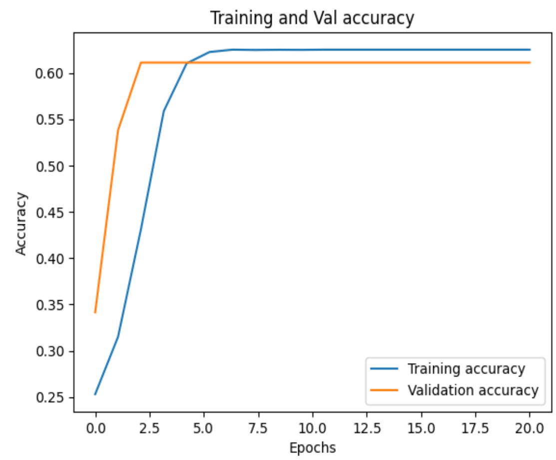


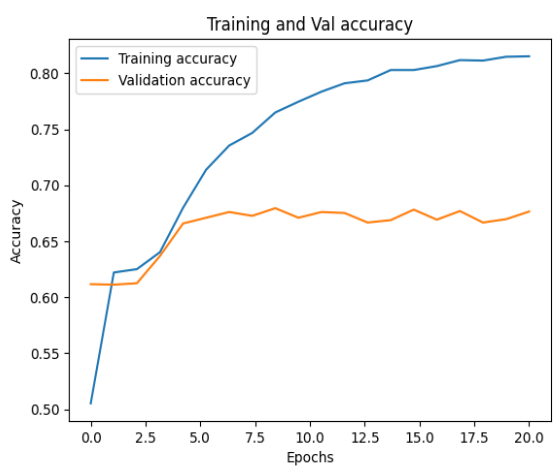
Figure 1f) Adding L2 Regularizer

Figure 1g) Using L1_L2 Regularizer

Overfitting: L2 regularizer model overfits (epoch 12.5) much later than the L1_l2 regularizer model (epoch 6) and all other models seen so far as seen in Figure 1f and 1g. The L1_L2 regularizer model significantly reduced overfitting as the training and validation accuracies only have a 0.1 difference.

Generalizable: The L2-regularizer model is more generalizable than the L1-L2 regularizer model as it had a greater test accuracy (0.7190) of the two.

Adding dropouts



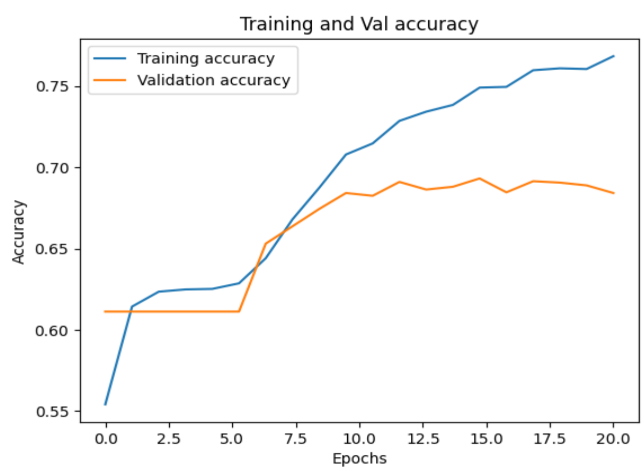


Figure 1h) Adding dropouts to Dense Layers

Figure 1i) Adding dropouts to RNN layers

Overfitting: The Dropouts to RNN layers model overfits at epoch 10 in Figure 1i which is later than the Dropouts to Dense Layers model, epoch 7.5, in Figure 1h.

DEEP LEARNING HW 3

Generalizable: The Dropouts to RNN layers model had the higher test accuracy of (0.6962) of the two models therefore it is more generalizable on unseen data as per Table 1.

Table 1: Comparing various simpleRNN models for Problem 1

Model	Epochs before overfitting	Train acc.	Val acc.	Test acc
Higher LR (0.001)	1.5	0.9613	0.6709	0.6948
Lower LR (0.00001)	10	0.9129	0.7157	0.7254
RNN (fewer units)	5	0.8880	0.6944	0.7056
RNN (more units)	**7**	**0.9561**	**0.7264**	**0.7551**
More Dense layers	7.5	0.9554	0.7187	0.7470
L2 regularizer	12.5	0.9560	0.7038	0.7190
L1_L2 regularizer	6	0.6250	0.6112	0.6291
Dropouts to Dense layers	7.5	0.8171	0.6765	0.6819
Dropouts to RNN layers	10	0.7742	0.6842	0.6962

Underfitting: None of the simpleRNN models are underfitting as all training and validation accuracies are above 0.5 as seen in Table 1.

Overfitting: All the models are overfitting as there's a drop between the Training and Validation accuracies. Overfitting has been heavily minimized for the 'L1_L2 regularizer' and 'Dropouts to RNN layers' models as there's only a 0.1 and 0.10 difference between training and validation accuracy, respectively. Overfitting could have been reduced by using fewer RNN layers in all the later models.

Best model: RNN models with more units was the best model as it had the highest test accuracy of 0.7551. Since it had the highest test accuracy, it is also more generalizable on unseen data compared to the other models. The 'More Dense layers' model was also equally as good as the RNN model with more units as can be seen by its test accuracy of 0.7470 in Table 1.

Problem 2 (LSTM)

Based on the tweets, the sentiment of the texts was predicted using LSTM layers, in this problem. LSTMs excel over SimpleRNNs by efficiently managing long-term dependencies through a gated system that preserves crucial information while eliminating unnecessary data. Moreover, they enhance training stability by addressing problems such as exploding gradients, leading to improved performance on longer and more intricate data sequences.

Models with various hyperparameters were experimented with to find the model with the best prediction accuracy. Some parameters that were kept consistent among the models are the total number of epochs (20), batch_size(32), and max_length(30).

Different Learning Rates

DEEP LEARNING HW 3

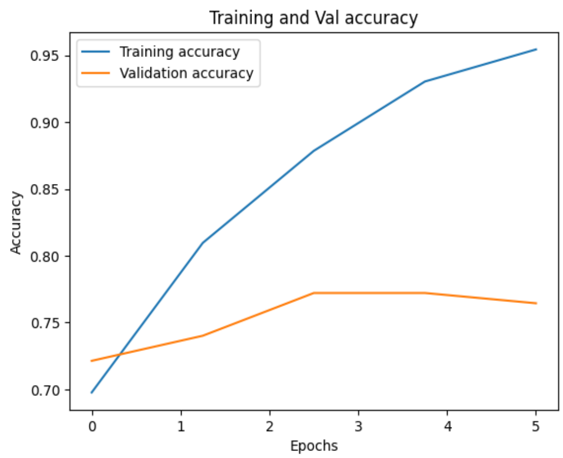


Figure 2a) Higher Learning rate (LR=0.001)

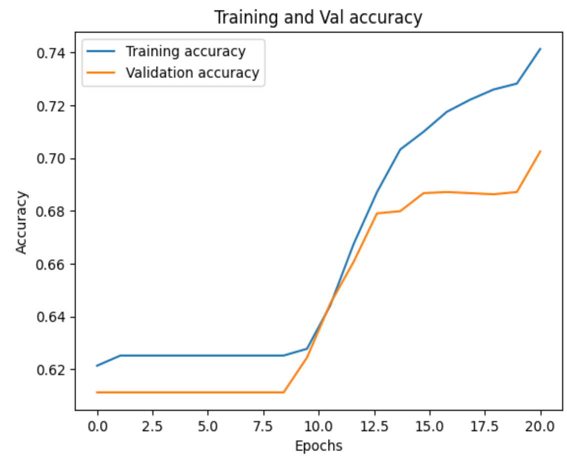


Figure 2b) Lower learning rate (LR = 0.00001)

LSTM units for both learning rates: 32, 32

Learning rates indicate the size of steps taken towards lowest gradient directing the loss function to a global minimum. The disadvantage of using a higher learning rate is that the model starts overfitting much earlier compared to the lower learning rate model.

Overfitting: Validation and test accuracies are less than training accuracies for both models which indicates overfitting. It is evident from Table1 that the lower learning rate model only overfits after 10 epochs whereas the higher learning rate model almost immediately starts overfitting.

Generalizable: The higher learning rate model has the higher test accuracy of 0.7832 as per Table2 which suggests that it is more generalizable in unseen data.

Experimenting with LSTM layers with different units and more Dense layers

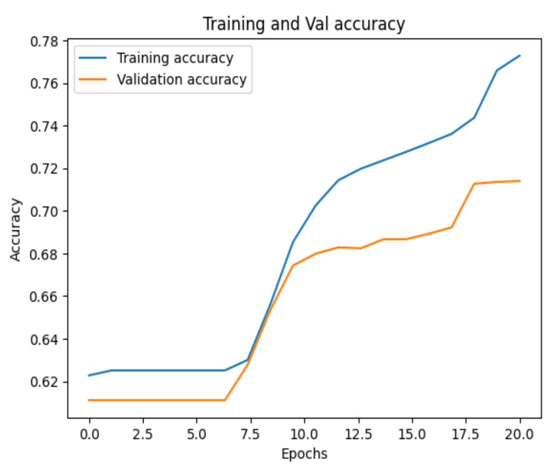


Figure 2c) LSTM with fewer units

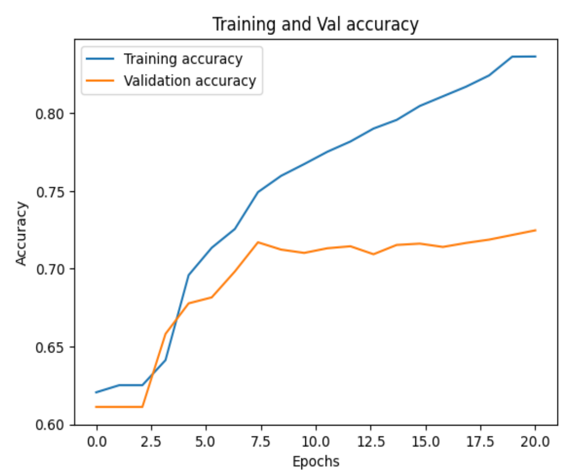


Figure 2d) LSTM with more units

DEEP LEARNING HW 3

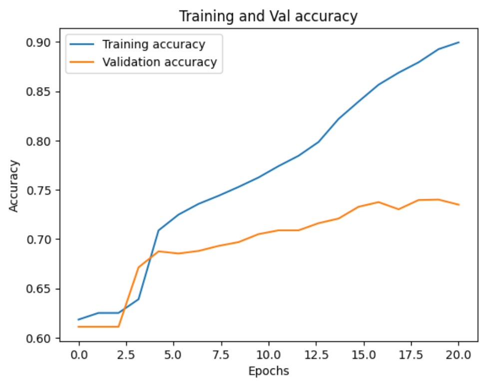


Figure 2e) Adding more Dense Layers

Overfitting: Of the three models, the More Dense Layers model overfits much later than the other three models as seen in Figures 2c, d,e. And this model overfits the most as there's a huge drop in validation accuracy of 0.7350 when the training accuracy is 0.8993 as seen in Table 2.

Generalizable: Of the three models the LSTM with more units is more generalizable as it had the highest test accuracy of 0.7412 as seen in Table 2.

Adding regularizers and dropouts lowered the training accuracy as can be seen in Table 2.

Different Regularizers

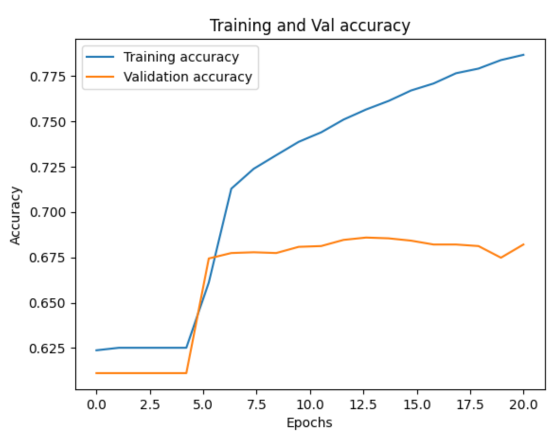


Figure 2f) Adding L2 Regularizer

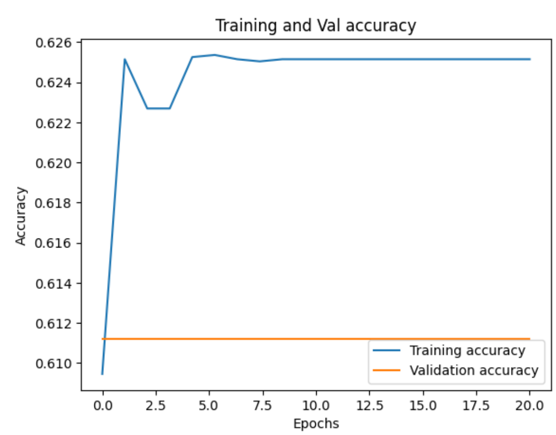


Figure 2g) Using L1_L2 Regularizer

Adding dropouts

DEEP LEARNING HW 3

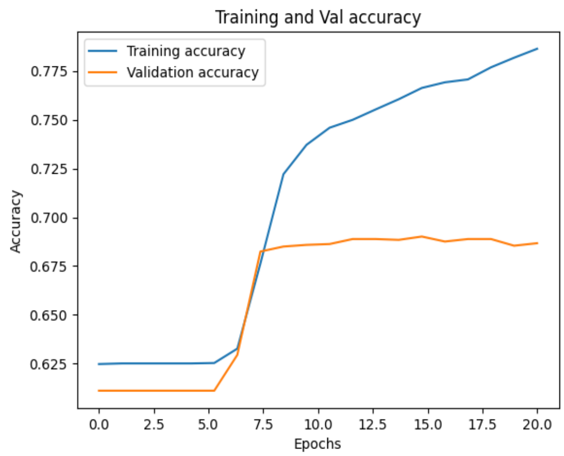


Figure 2h) Adding dropouts to Dense Layers

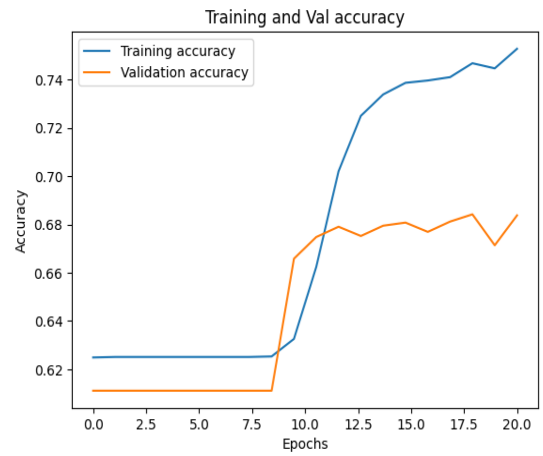


Figure 2i) Adding dropouts to LSTM layers

Overfitting: Dropouts to LSTM layers model starts overfitting a little later compared to the dropouts to Dense layers model. Dropouts to Dense layers model had a huge drop between training and validation accuracy of about 0.10 as can be seen in table 2 thus this model was overfitting more.

Generalizable: Dropouts to Dense layers model had the higher test accuracy of 0.7064 than the Dropouts to LSTM layers model as seen in Table2.

Table 2: Comparing various LSTM models for Problem 2

Model	Epochs before overfitting	Train acc.	Val acc.	Test acc
Higher LR (0.001)	**2.5**	**0.9596**	**0.7644**	**0.7832**
Lower LR (0.00001)	15	0.7285	0.7025	0.7333
LSTM (fewer units)	12	0.7645	0.7140	0.7299
LSTM (more units)	7.5	0.8408	0.7247	0.7412
More Dense layers	15	0.8993	0.7350	0.7405
L2 regularizer	6	0.7895	0.6820	0.6993
L1_L2 regularizer	1	0.6394	0.6112	0.6291
Dropouts to Dense layers	7.5	0.7886	0.6867	0.7064
Dropouts to LSTM layers	11	0.7515	0.6837	0.6874

Underfitting: None of the LSTM models are underfitting as all training and validation accuracies are above 0.5 as seen in Table 1.

Overfitting: Both the Lower Learning rate model and the More Dense layer model overfits the latest (epoch 15). The Higher learning rate model overfits the quickest (epoch 2.5).

PRASANNA, LAKSITA
3/5/25

DEEP LEARNING HW 3

Best model: The Higher learning rate model was the best model for LSTM as it had the highest test accuracy of 0.7832 which was greater than that of all models. Therefore, the higher learning rate model was also more generalizable on unseen data compared to the other models.