CSI5138 Assignment3

Name: Jiacheng Hou
Student Number: 300125708

1 Introduction

In the assignment, I aim to develop two text classification models using Vanilla RNN and LSTM for the IMDB Movie Review Dataset. For each model, I take the final output of the chain as the extracted text feature. Besides, I use the Glove for the embedding vectors. For Vanilla RNN and LSTM, I try state dimension from the list [20, 50, 100, 200, 500] and then tune other hyper-parameters to obtain the best classification result. Table 1 and Table 2 show hyper-parameters for different state dimensions in Vanilla RNN and LSTM, respectively.

Table 1: Vanilla RNN Model Hyper-Parameters

State Dimension	Epoch	Batch Size	Learning Rate	Recurrent Layers	Embedding Dimension	Output Dimension	Bidirectional	Dropout Probability	Test Accuracy
20	10	64	0.001	3	100	1	False	0.1	73.07%
50	10	64	0.001	3	100	1	True	0.2	71.48%
100	10	64	0.001	2	100	1	True	0.1	71.09%
200	10	64	0.001	1	100	1	True	0	66.75%
500	10	64	0.001	2	100	1	True	0.25	55.31%

Table 2: LSTM Model Hyper-Parameters

State Dimension	Epoch	Batch Size	Learning Rate	Recurrent Layers	Embedding Dimension	Output Dimension	Bidirectional	Dropout Probability	Test Accuracy
20	10	64	0.001	3	100	1	True	0.4	88.04%
50	10	64	0.001	2	100	1	True	0.4	87.01%
100	10	64	0.001	2	100	1	True	0.4	88.32 %
200	10	64	0.001	2	100	1	True	0.4	88.68%
500	10	64	0.001	2	50	1	True	0.4	89.21%

2 Vanilla RNN Model

This section includes two parts, the explanation of the Vanilla RNN model and results.

2.1 Model

The first part is to prepare data. I split the training dataset into training and validation set. Then for the training set, I only keep the top 25,000 words.

For those words that have been cut, I replace them with the $< unk >$ and $< pad >$ tokens. Then I use the pre-trained word embeddings from Glove with 50-dimensional or 100-dimensional vectors. Those words are initialized via a Gaussian distribution. Finally, I create packed padded sequences with bach size 64.

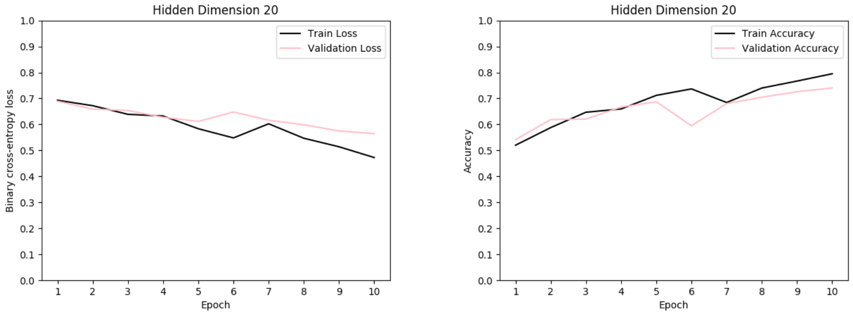
The second part is to build the model. The layers of the model include an embedding layer, multi-layer RNN and a linear layer. The embedding layer transforms word embedding vectors into a dense embedding vector. Following the embedding layer is the RNN layer. I try one RNN layer, two RNN layers and three RNN layers with bidirectional or not. In case the model goes into overfitting, I add dropout regularization, which is used on the connections between hidden states in one layer to hidden states in the next layer. Finally, the linear layer takes the final hidden state and feeds it through a fully connected layer, transforming it to the output dimension.

The third part is to train the model. Firstly, I use Adam optimizer. Next, I define the loss function, which carries out both the sigmoid and the binary cross-entropy steps. I train the model for ten epochs. At each epoch, training loss, training accuracy, validation loss and validation accuracy are calculated. After training has finished, I use the model on the testing dataset if the validation loss is the best.

2.2 Results

This section explains the loss and accuracy results for different state dimensions in the Vanilla RNN model.

When the hidden dimension is 20, hyper-parameters are epoch 10, batch dimension 64, learning rate 0.001, recurrent layers 3, embedding dimension 100, output dimension 1, embedding dimension 100, bidirectional false, dropout probability 0.1. After ten epochs of training, I choose the model on the testing dataset if the validation loss is the best, and the test accuracy is 73.07%. Figure 1a shows the training dataset and validation dataset loss change over time. Figure 1b shows the training dataset and validation dataset accuracy change over time.

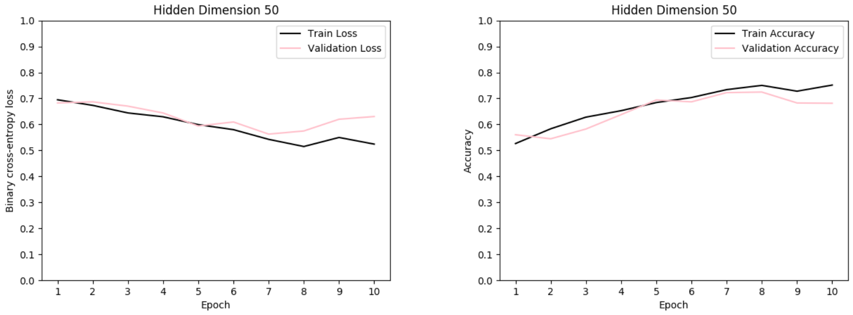


(a) Training dataset loss and validation dataset loss for 10 epochs

(b) Training dataset accuracy and validation dataset accuracy for 10 epochs

Figure 1: Vanilla RNN model results with hidden dimension 20

When the hidden dimension is 50, hyper-parameters are epoch 10, batch dimension 64, learning rate 0.001, recurrent layers 3, embedding dimension 100, output dimension 1, embedding dimension 100, bidirectional true, dropout probability 0.2. After ten epochs of training, I choose the model on the testing dataset if the validation loss is the best, and the test accuracy is 71.48%. Figure 2a shows the training dataset and validation dataset loss change over time. Figure 2b shows the training dataset and validation dataset accuracy change over time.

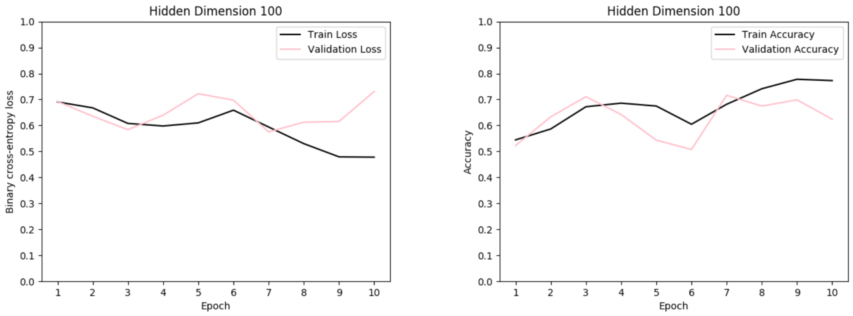


(a) Training dataset loss and validation dataset loss for 10 epochs

(b) Training dataset accuracy and validation dataset accuracy for 10 epochs

Figure 2: Vanilla RNN model results with hidden dimension 50

When the hidden dimension is 100, hyper-parameters are epoch 10, batch dimension 64, learning rate 0.001, recurrent layers 2, embedding dimension 100, output dimension 1, embedding dimension 100, bidirectional true, dropout probability 0.1. After ten epochs of training, I choose the model on the testing dataset if the validation loss is the best, and the test accuracy is 71.09%. Figure 3a shows the training dataset and validation dataset loss change over time. Figure 3b shows the training dataset and validation dataset accuracy change over time.

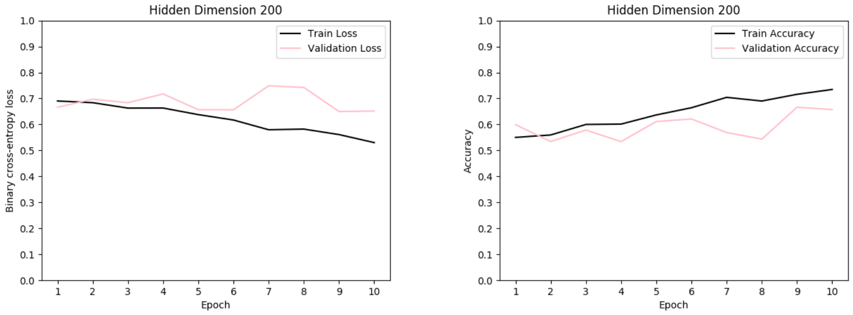


(a) Training dataset loss and validation dataset loss for 10 epochs

(b) Training dataset accuracy and validation dataset accuracy for 10 epochs

Figure 3: Vanilla RNN model results with hidden dimension 100

When the hidden dimension is 200, hyper-parameters are epoch 10, batch dimension 64, learning rate 0.001, recurrent layers 1, embedding dimension 100, output dimension 1, embedding dimension 100, bidirectional true, dropout probability 0. After ten epochs of training, I choose the model on the testing dataset if the validation loss is the best, and the test accuracy is 66.75%. Figure 4a shows the training dataset and validation dataset loss change over time. Figure 4b shows the training dataset and validation dataset accuracy change over time.



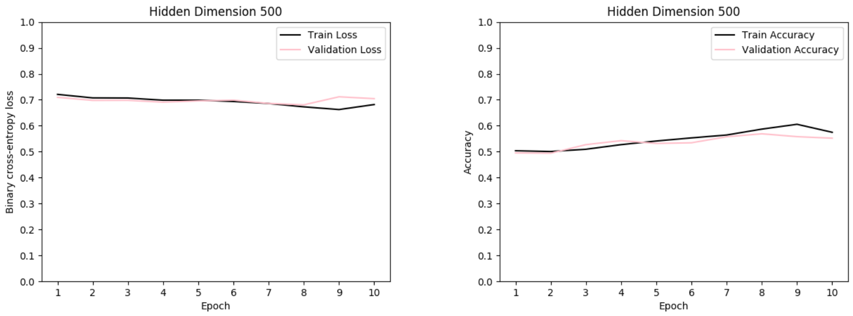
(a) Training dataset loss and validation dataset loss for 10 epochs

(b) Training dataset accuracy and validation dataset accuracy for 10 epochs

Figure 4: Vanilla RNN model results with hidden dimension 200

When the hidden dimension is 500, hyper-parameters are epoch 10, batch dimension 64, learning rate 0.001, recurrent layers 2, embedding dimension 100, output dimension 1, embedding dimension 100, bidirectional true, dropout probability 0.25. After ten epochs of training, I choose the model on the testing dataset if the validation loss is the best, and the test accuracy is 55.31%.

Figure 5a shows the training dataset and validation dataset loss change over time. Figure 5b shows the training dataset and validation dataset accuracy change over time.



(a) Training dataset loss and validation dataset loss for 10 epochs

(b) Training dataset accuracy and validation dataset accuracy for 10 epochs

Figure 5: Vanilla RNN model results with hidden dimension 500

3 LSTM Model

This section includes two parts, the explanation of the LSTM model and results.

3.1 Model

For the LSTM model, there are two different parts compared with the Vanilla RNN models.

The first part is to use a different RNN architecture called an LSTM when building the model.

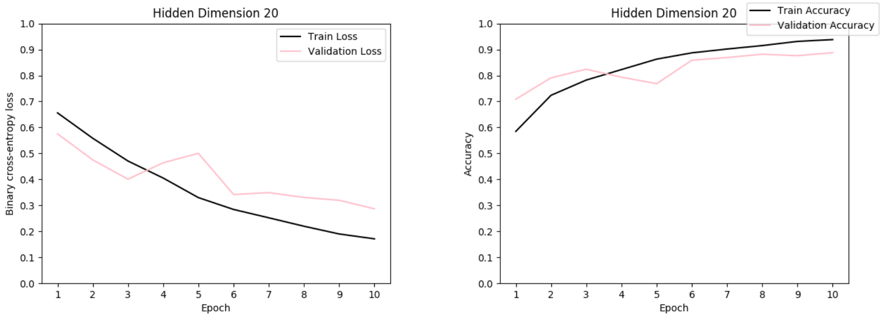
The second part is that I am not going to learn the embedding for the $< pad >$ token in the training dataset. I want to tell the model that padding tokens are irrelevant to the classification. I get the $< pad >$ token index from the vocabulary and pass the index to the embedding layer. The embedding for the $< pad >$ token will remain at what it is initialized to.

3.2 Results

This section explains the loss and accuracy results for different state dimensions in the LSTM model.

When the hidden dimension is 20, hyper-parameters are epoch 10, batch dimension 64, learning rate 0.001, recurrent layers 3, embedding dimension

100, output dimension 1, embedding dimension 100, bidirectional true, dropout probability 0.4. After ten epochs of training, I choose the model on the testing dataset if the validation loss is the best, and the test accuracy is 88.04%. Figure 6a shows the training dataset and validation dataset loss change over time. Figure 6b shows the training dataset and validation dataset accuracy change over time.

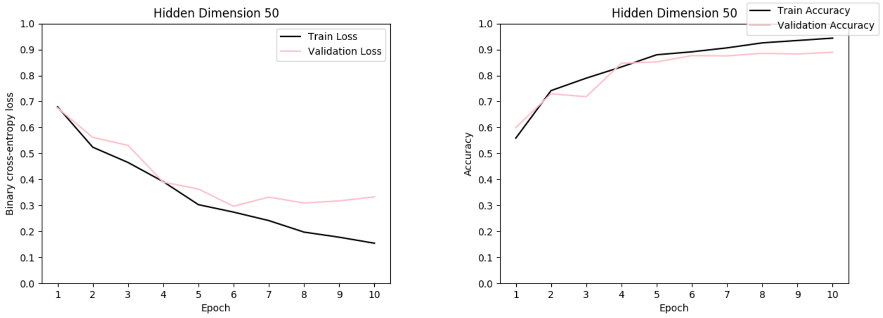


(a) Training dataset loss and validation dataset loss for 10 epochs

(b) Training dataset accuracy and validation dataset accuracy for 10 epochs

Figure 6: LSTM model results with hidden dimension 20

When the hidden dimension is 50, hyper-parameters are epoch 10, batch dimension 64, learning rate 0.001, recurrent layers 2, embedding dimension 100, output dimension 1, embedding dimension 100, bidirectional true, dropout probability 0.4. After ten epochs of training, I choose the model on the testing dataset if the validation loss is the best, and the test accuracy is 87.01%. Figure 7a shows the training dataset and validation dataset loss change over time. Figure 7b shows the training dataset and validation dataset accuracy change over time.

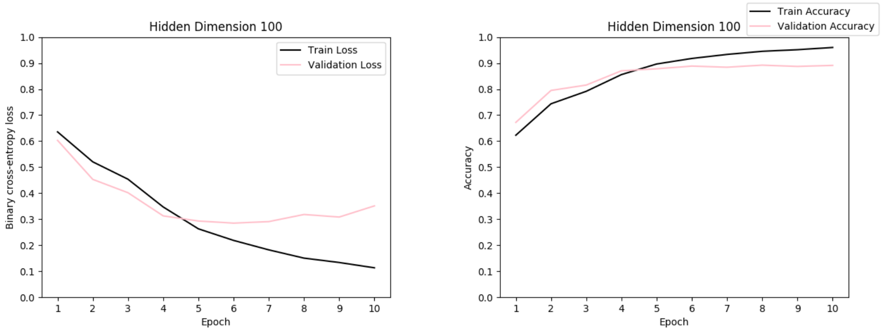


(a) Training dataset loss and validation dataset loss for 10 epochs

(b) Training dataset accuracy and validation dataset accuracy for 10 epochs

Figure 7: LSTM model results with hidden dimension 50

When the hidden dimension is 100, hyper-parameters are epoch 10, batch dimension 64, learning rate 0.001, recurrent layers 2, embedding dimension 100, output dimension 1, embedding dimension 100, bidirectional true, dropout probability 0.4. After ten epochs of training, I choose the model on the testing dataset if the validation loss is the best, and the test accuracy is 88.32%. Figure 8a shows the training dataset and validation dataset loss change over time. Figure 8b shows the training dataset and validation dataset accuracy change over time.

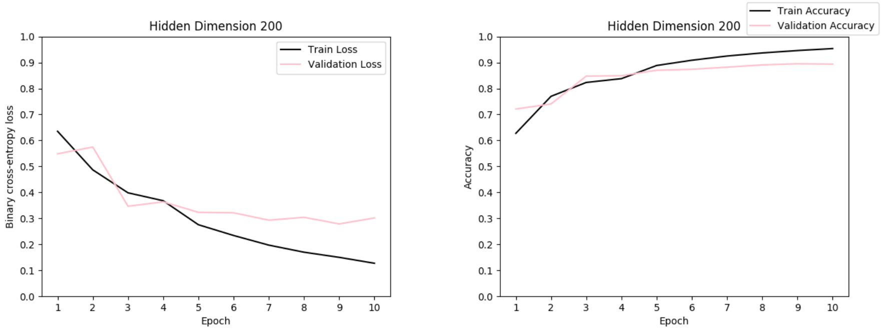


(a) Training dataset loss and validation dataset loss for 10 epochs

(b) Training dataset accuracy and validation dataset accuracy for 10 epochs

Figure 8: LSTM model results with hidden dimension 100

When the hidden dimension is 200, hyper-parameters are epoch 10, batch dimension 64, learning rate 0.001, recurrent layers 2, embedding dimension 100, output dimension 1, embedding dimension 100, bidirectional true, dropout probability 0.4. After ten epochs of training, I choose the model on the testing dataset if the validation loss is the best, and the test accuracy is 88.68%. Figure 9a shows the training dataset and validation dataset loss change over time. Figure 9b shows the training dataset and validation dataset accuracy change over time.

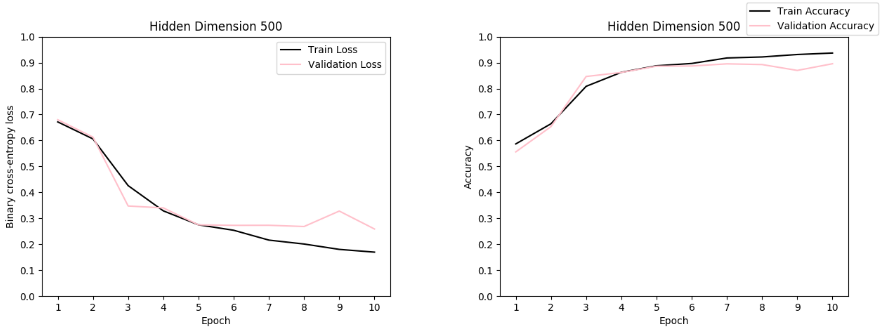


(a) Training dataset loss and validation dataset loss for 10 epochs

(b) Training dataset accuracy and validation dataset accuracy for 10 epochs

Figure 9: LSTM model results with hidden dimension 200

When the hidden dimension is 500, hyper-parameters are epoch 10, batch dimension 64, learning rate 0.001, recurrent layers 2, embedding dimension 50, output dimension 1, embedding dimension 100, bidirectional true, dropout probability 0.4. After ten epochs of training, I choose the model on the testing dataset if the validation loss is the best, and the test accuracy is 89.21%. Figure 10a shows the training dataset and validation dataset loss change over time. Figure 10b shows the training dataset and validation dataset accuracy change over time.



(a) Training dataset loss and validation dataset loss for 10 epochs

(b) Training dataset accuracy and validation dataset accuracy for 10 epochs

Figure 10: LSTM model results with hidden dimension 500