AML Lab 2 Report

Team Number: 3

Team Name: The Anti Triads

Team Members:

USN	Name
01FB15ECS278	Shashank Saran
01FB15ECS286	Shreyas Vivek Patil
01FB15ECS289	Siddhanth Vinay
01FB15ECS290	Siddharth Ganesan

Dataset:

Our dataset contains 20,000 samples from the Kaggle-Quora Question Answer pairs dataset as downloaded using the web-service API provided for the team.

Steps Followed:

1) The samples obtained from the web-service was converted into a pandas dataframe.
2) From the list of samples, all the question pairs were taken into a list and tokenized into a list of words.
3) All words were converted to lowercase and words below a frequency of 2 were removed to get a vocabulary size of 9048.
4) For all the words, the web service was invoked to get the corresponding embeddings and for words which threw an error when invoking the web service, the exception was caught, and a vector of 50 zeros was appended as the vector for those words.
5) The corresponding word-vector pair list obtained was converted into a pandas dataframe and an entry was added for out of vocabulary words corresponding to a vector of zeros.
6) Using the vector column of the dataframe and converting it into a numpy array, the embedding matrix was formed.
7) Using the indices for the words in the dataframe, each question from both the question pairs were converted into an embedding sequence and for words not present in the word-vector dataframe, the index corresponding to the out of vocabulary was appended into the sequence.
8) All sequences were then padded to the length of the maximum sequence (232) by appending all the sequences to the index corresponding to the vector of zeros.
9) The model was then designed by following an architecture similar to the diagram specified in the question-paper where each input was embedded separately, and an

LSTM was built for each of them, the output of which was concatenated and fed to a neural network with a single dense layer connected to an output layer with a sigmoid activation function. A detailed architecture will be provided in the next section.

10) The model was then trained on the sequences, using 17,000 sequence pairs for training (85% of the 20,000 samples) and the remaining 3000 sequence pairs for validation.
11) Numerous models with varying number of hidden units and other varying other hyperparameters were tried out, additionally adding regularizers and dropouts, while also adding additional dense layers to the neural network to see if there was any improvement in training, until the best network was finalized upon.

Final Model Architecture

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_5 (InputLayer)         (None, 232)               0
_________________________________________________________________
input_6 (InputLayer)         (None, 232)               0
_________________________________________________________________
embedding_6 (Embedding)      (None, 232, 50)           452800
                                                       input_5[0][0]
                                                       input_6[0][0]
_________________________________________________________________
lstm_16 (LSTM)               (None, 20)                1120
                                                       embedding_6[0][0]
                                                       embedding_6[1][0]
_________________________________________________________________
concatenate_11 (Concatenate) (None, 10)                0
                                                       lstm_16[0][0]
                                                       lstm_16[1][0]
_________________________________________________________________
dropout_20 (Dropout)         (None, 10)                0
                                                       concatenate_11[0][0]
_________________________________________________________________
batch_normalization_18 (BatchNo (None, 10)             40
                                                       dropout_20[0][0]
_________________________________________________________________
dense_19 (Dense)             (None, 20)                220
                                                       batch_normalization_18[0][0]
_________________________________________________________________
dropout_21 (Dropout)         (None, 20)                0
                                                       dense_19[0][0]
```

```
_________________________________________________________________________________
_____________________
batch_normalization_19 (BatchNo (None, 20)       80          dropout_21[0][0]
_________________________________________________________________________________
_____________________
dense_20 (Dense)             (None, 1)           21

                                                 batch_normalization_19[0][0]
=================================================================================
======================
Total params: 454,281
Trainable params: 1,421
Non-trainable params: 452,860
_________________________________________________________________________________
_____________________
```

More model details:

Optimizer: Adadelta optimizer.

Batch size: 64

Learning Rate left as default as suggested by Keras documentation for the Adadelta optimizer.

Number of epochs: 25 (training time was very high on a CPU machine)

LSTM: 20 units

Dense layer:

1. Number of units: 20
2. Activation function: Exponential Linear Unit
3. Regularizers: Kernel Regularizer: l2(0.1)
4. Activity Regularizer: l1(0.01)
5. Dropout 0.2

Loss function: Mean squared error

The regularizers, dropouts and batch normalization were added to the network to help deal with both overfitting and to help in accelerating convergence to the best possible minima.

Model Results:

Training Accuracy: **0.6264**

Training Loss: 0.3628

Validation Accuracy: **0.6367**

Validation Loss: 0.3646