**Report for rating prediction**

1. **Data preprocessing:**
2. Change all words to lower case.
3. Remove new line symbol “\n” and slash symbol “\”.
4. Remove words with numbers.
5. Remove punctuations.
6. Remove stopwords from nltk.corpus, excluding “not”, “nor” and “not”.
7. Word stemming with PorterStemmer.
8. Add “<pad>” to empty sentences.
9. **Generate Word2Vector model:**

We use genism to create word2vector.txt which contains all frequent words in train dataset. And in the next step, we load this word2vetor model, and map the word into its corresponding vector in a dimension of 100. This reduce input dimensionality since one-hot-encoding is not efficient for many words. Compared to other pre-trained word2vec model such as “glove.6B.100d”, our model is more succinct.

1. **Split training and validation set**

Since test set will be provided by the supervisor of this course, we only split the data set into training set and validation sets. The split fraction is set to 0.9. When the supervisor uses the model for test dataset, preprocessing for test dataset should be first conducted.

1. **Torchtext handling**

The torchtext package consists of different data processing utilities. For example, the Field class models common text processing datatypes that can be represented by tensors, the iterator class will load batches of data from a dataset.

1. **Build neural network**

We create two neural network: one is CNN, another is GRU in recurrent network.

1. The learning rate we tried were 1, 0.1, 0.01, 0.001, 0.0001 and 0.00001. We choose the learning rate to be 0.0001, because larger learning rate lead to fast growth of training loss and poor prediction. This is possibly due to fast convergence to a suboptimal solution. And small learning rate also leads to slow convergence.
2. The dropout rate we tried were 0.3, 0.5 and 0.7. It turned out the 0.5 gave relatively better result.
3. The batch size selected was 512, as we tried from 64,128,256,512 and 1024. Due to the long training time for small batch size, we selected 512 to accelerate the training process.
4. The training epoch is set to 6, because with more epochs (>10), although the accuracy and training loss for training set still increases, the scores for validation set start to drop. We set epoch to 6 to avoid overfitting. Another method we adopt to avoid overfitting is to introduce dropout with value of 0.5.
5. **Result**

After training, the accuracy and training loss for training dataset and validation set are as followed:

|  |  |  |
| --- | --- | --- |
| **GRU network** | training set | validation set |
| training loss | 0.81 | 0.84 |
| accuracy | 0.64 | 0.62 |

|  |  |  |
| --- | --- | --- |
| **CNN network** | training set | validation set |
| training loss | 0.81 | 0.84 |
| accuracy | 0.64 | 0.62 |

1. **Overview for the process**

