Recommender System Report

Kaggle team ID: Zi Han Zhao

Student ID: zhaozih3

1 Describe how you processed your data and what features you used. Your exploratory analysis here should motivate the model you use in the next section.

2 Describe your model. Explain and justify your decision to use the model you proposed. How will you optimize it? Did you run into any issues due to scalability, overfitting, etc.? What other models did you consider for comparison? What were your unsuccessful attempts along the way? What are the strengths and weaknesses of the different models being compared?

Two other alternative models are Gated Recurrent Unit (GRU) model and Transformer TFBertModel.
GRU is a improved version of recurrent neural network [1]. The reference of my GRU design is TensorFlow RNN Text binary classification for IMDB review [2]. The training/test dataset formatting of GRU is as following: first the number of each word is counted and sorted through "*reviewText*" and "*summary*" entries in *train.json*. The label is "*overall*". Top 20000 most-frequent used word list are stored. Second, looking into each single data in *train.json*, if a word in the data is found in word list, I encode the word to a number representing the index of word list. E.g, if "*the*" is found in word list and it is in index 0 of word list, it means the word is 1st most frequent in training data and index 0 is the encoding output of "*the*". The words not in the word list are abandoned. The layers are embedding layer, GRU layer and Dense layer. The output is multi-class corresponding to "*overall*" five floating numbers.
Similar with LSTM model, GRU model is capable of long-term next-sequence prediction. Compared with LSTM, the number of parameters is reduced so that the occurrence of overfit reduces and the speed of training increases. **TODO: cons by comparing with current model.**
TFBertModel is also a popular model for next-line prediction [3]. The input dataset is the string sequence of combining "*reviewText*" and "*summary*" entries. The label is "*overall*". The sequences are encoded into "*input_ids*" list. The layers are Input Layer, TFBertMainLayer, Dropout layer, Dense output layer. The output is also multi-class.

TFBertModel makes use of Transformer model structure to learn the relations among words or sub-words. The accuracy of prediction is higher under large enough dataset and iterations [4]. However, **TODO: cons by comparing with current model.**. Besides, the computation resources are heavy and it is time-consuming.

3 Describe your results and conclusions. How well does your model perform compared to alternatives, and what is the significance of the results? Which feature representations worked well and which do not? What is the interpretation of your model's parameters? Why did the proposed model succeed why others failed (or if it failed, why did it fail)?

References

[1] Understanding GRU Networks
https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be

[2] Text classification with an RNN
https://www.tensorflow.org/tutorials/text/text_classification_rnn

[3] Multi-Label, Multi-Class Text Classification with BERT, Transformers and Keras
https://towardsdatascience.com/multi-label-multi-class-text-classification-with-

[4] BERT Explained: State of the art language model for NLP
https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-fo