Sentiment Analysis on IMDB

Sruthi Reddy Rekula
August 23 2023
https://github.com/shruthireddyrekula/Sentiment_analysis_IMDB_TF

Sentiment analysis involves determining the sentiment expressed in text, such as whether a movie review is positive or negative. Observations for IMDB data on sentiment analysis using different models (MLP, LSTM, and Transformer) with the model's performance on the dataset and insights from the results. Here are the observations for each model:

Multi-Layer Perceptron (MLP):

- MLP model used here consists of an embedding layer as first layer, one hidden layer and an output layer. The model is designed for a binary classification task with 2 output classes. The 'relu' activation function is used in the hidden layers to introduce non-linearity, and sigmoid is used in the output layer for binary-class probability estimation.
- The IMDB dataset is preprocessed by truncating and adding padding sequences to create fixed_length inputs that can be fed to machine learning models, as many models require input data to be same shape.
- The hyperparameters used during training are batch-size of 128, adam optimizer with binary_crossentropy loss and 10 epochs.
- There are slight signs of overfitting since the training accuracy and loss are slightly better than the validation accuracy and loss. However, the differences are not substantial, and the model seems to be learning to some extent from the data.
- Overall Analysis: The MLP model does perform well on the IMDB test dataset. The test accuracy of approximately 86.63% suggests that the model generalizes to unseen data effectively.

Long Short Term Memory(LSTM):

- Created LSTM Model using ‘Sequential’ API, adding layers of embedding, LSTM and dense output
- Preproccesed the text data by using ‘sequence.pad_sequences’ function to ensure sequences are of the same length.

- The hyperparameters used during training are batch-size of 128, adam optimizer with binary_crossentropy, loss and 10 epochs.
- The model's performance is quite good, as it achieved high accuracy on both the training and validation datasets, and it was able to generalize well to new, unseen data in the test dataset. The test accuracy is in line with the validation accuracy, indicating that the model is robust and capable of making accurate predictions on real-world data.
- The LSTM model shows significantly higher accuracy on the training set compared to the MLP, indicating that it is better at learning the representations of the training data.

Transformer:

- Trained IMDB dataset using transformer model with BERT tokenizer. The BERT based model is used for converting text sentences into tokens and padded tensors.
- These tokenized tensors are then used to determine a movie review as positive or negative by binary classification.
- The hyperparameters used during training are learning rate of 1e-5 , batch-size of 12, Adam optimizer with binary_crossentropy loss and 10 epochs
- The Transformer model demonstrates exceptional accuracy on both the training and validation sets, indicating that it has learned the complex patterns in the data very well. It also achieves high accuracy on the test set, suggesting that it generalizes effectively to unseen data. The transformer model outperforms both the MLP and LSTM by a significant margin.

In summary, the MLP model has its strengths in simplicity, fast training, and low resource requirements. However, it may not be the best choice for text classification tasks like IMDM dataset, where contextual and semantic relationships play a crucial role in achieving high accuracy. Deeper models like LSTM and advanced models like Transformers have shown superior performance on challenging NLP tasks, by leveraging the power of contextual understanding and pre-trained representations, transformers provide valuable insights into sentiment classification, making them a pivotal tool in natural language processing tasks.

Model	Train_Accuracy	Validation_Accuracy	Test_Accuracy
MLP	86.63%	86.63%	86.63%
LSTM	94.85%	86.98%	86.98%
Transformer	**99.61%**	91.70%	91.70%

The graphical representations provide below is a clearer visualization of the performance differences between the models, helping us understand the strengths and weaknesses of each model on the task at