Lab Assignment 3
CSCI 5992 - Neural Networks and Deep Learning

Rajeev R Menon - 110581437

November 3, 2022

Problem 1

Methods

The dataset used in this experiment is the SMS spam collection dataset compiled by Almeida et al. [1]. It consists of 5574 messages which are classified into 'ham' (legitimate) and 'spam'. Of the 5574 messages in the dataset, 747 messages are classified as 'spam' and 4825 messages are classified as 'ham'.

The experiment was conducted on a 14-inch MacBook Pro with an 8-core M1 Pro CPU. The dataset was split into two - training(70%) and test(30%) datasets. A total of three different models were trained using different types of Recurrant Neural Networks (RNN).

The data is initially preprocessed by tokenizing and padding shorter messages up to the number of tokens in the longest message. All the models consisted of an initial embedding layer with input dimension same as the number of tokens which is one more than the number of words in the vocabulary (to accommodate unknown words) and output dimension of 32. The embedding layer is then followed by a simple RNN or Long Short-Term Memory (LSTM) layer or a Gated Recurring Unit layer in each of the three models with 32 nodes. The RNN layer is followed by a dense fully connected layer to a single output node. RMSProp algorithm [2] was used for optimization and the learning rate was set to the default value of 0.001. A binary cross-entropy loss function was used to train all models. All the models were trained for 10 epochs using the Keras library for Tensorflow in Python. A constant random seed of 69 was used in Python, NumPy and Tenserflow environments to reproduce results.

After training all three models, precision and recall were calculated for each model on prediction against the test dataset. The test dataset was then further divided into 3 almost equal splits consisting of short, medium and long messages. Predictions were run against three of these splits for each of the three models and precision and recall were calculated.

Model	Model Type	Spam Precision	Ham Precision	Spam Recall	Ham Recall
Vanilla RNN	Full test data	0.94	0.98	0.89	0.99
Vanilla RNN	Short SMS test data	0.42	1.00	0.71	0.99
Vanilla RNN	Medium long SMS test data	0.80	0.99	0.73	0.99
Vanilla RNN	Long SMS test data	0.99	0.95	0.91	0.99
LSTM Model	Full test data	1.00	0.98	0.90	1.00
LSTM Model	Short SMS test data	1.00	0.99	0.43	1.00
LSTM Model	Medium long SMS test data	1.00	0.99	0.43	1.00
LSTM Model	Long SMS test data	1.00	0.96	0.93	1.00
GRU Model	Full test data	0.98	0.99	0.93	1.00
GRU Model	Short SMS test data	0.71	1.00	0.71	1.00
GRU Model	Medium long SMS test data	0.95	0.99	0.82	1.00
GRU Model	Long SMS test data	0.99	0.97	0.95	0.99

Table 1: Analysis on various input lengths

Results

Precision and recall for all the experiments are given in table 1. Vanilla RNN model showed comparatively low accuracy for short spam messages while the precision for predicting ham remained almost the same. The recall of spam was lower for short and medium messages compared to long messages. The LSTM model offered high precision for spam and ham messages of all lengths. The recall values were low for short and medium spam messages compared to long messages while recall for ham messages remained the same. The GRU model, similar to the vanilla RNN model, offered low precision for short spam messages while the precision for ham messages did not change much. The recall for spam messages was lower for short messages compared to medium and long messages.

Analysis

The precision values for short spam messages were lower compared to the values for medium and long spam messages while the precision for ham messages remained almost the same for all the models. The recall values of short and medium messages were lower compared to recall for long messages for all the models. It was observed that LSTM and GRU models performed better than the vanilla RNN model. The LSTM model gives the overall best performance compared to both vanilla RNN and GRU networks.

The number of samples of ham messages is much higher than the number of samples of spam messages in the dataset. The higher variation in metrics for spam messages compared to ham messages can be attributed to this skewness of the dataset. The better performance of LSTM and GRU model can be attributed to the presence of additional units to retain information in memory for longer periods. While GRU models are more computationally efficient than LSTM models, LSTM models having an additional gate compared to GRU

models can remember longer sequences enabling it to outperform GRU models in tasks requiring modeling long-distance relations. Hence the better performance of the LSTM model can be attributed to the presence of more longer messages in the dataset.

Problem 2

Methods

The dataset used in this experiment is the SMS spam collection dataset compiled by Almeida et al. [1]. It consists of 5574 messages which are classified into 'ham' (legitimate) and 'spam'. Of the 5574 messages in the dataset, 747 messages are classified as 'spam' and 4825 messages are classified as 'ham'.

The experiment was conducted on a 14-inch MacBook Pro with an 8-core M1 Pro CPU. The dataset was split into two - training (70%) and test (30%) datasets. A total of three different models were trained using two RNN models with different pre-trained embedding layers. The first pre-trained embedding layer consists of word vectors obtained from GloVe, an unsupervised learning algorithm [3]. The second pre-trained embedding layer is obtained from the spaCy natural language processing library in Python [4].

The data is initially preprocessed by tokenizing and padding shorter messages up to the number of tokens in the longest message. All the models consisted of one of the pre-trained embedding layers with input dimension same as the number of tokens which is one more than the number of words in the vocabulary (to accommodate unknown words) and output dimension of 32. The embedding layer is then followed by an LSTM layer with 32 nodes. The LSTM layer is followed by a dense fully connected layer to a single output node. RMSProp algorithm was used for optimization and the learning rate was set to the default value of 0.001 [2]. A binary cross-entropy loss function was used to train all models. All the models were trained for 10 epochs using the Keras library for Tensorflow in Python. A constant random seed of 69 was used in Python, NumPy and Tenserflow environments to reproduce results.

Results

The model with GloVe embeddings reported a precision of 0.96 for spam messages and 0.99 for ham messages while the recall was 0.92 for spam and 0.99 for ham. The model with spaCy embeddings reported a precision of 0.98 for both spam and ham messages while the recall was 0.85 for spam and 1.00 for ham. Figures 1 and 2 shows the confusion matrices for the predictions made by the models using GloVe and spaCy embeddings respectively.

Analysis

The model with spaCy embeddings reported slightly lesser false positives compared to the one with GloVe embeddings. The model with GloVe embedding reported slightly lesser false

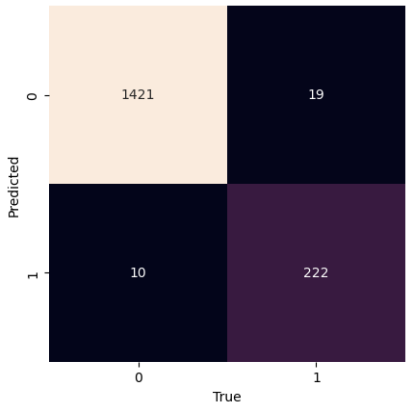

Figure 1: Confusion matrix for model with Glove embeddings

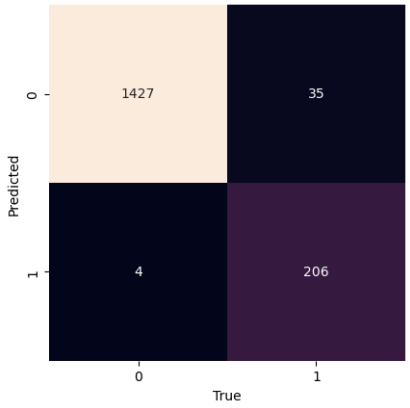

Figure 2: Confusion matrix for model with spaCy embeddings

negatives compared to the one with spaCy embeddings. It was observed that the model with spaCy embeddings performed slightly better than the model with GloVe embeddings in predicting ham messages while the model with GloVe embedding was better in detecting spam.

Although GloVe embeddings helped in detecting more spam, spaCy embeddings is observed to be more precise. GloVe embeddings were trained with more data than the spaCy embeddings. Therefore GloVe provides more accurate representation of general messages which lead to better prediction of ham messages with respect to more true negatives and lesser false negatives. The model with spaCy embeddings performing better in predicting spam and reporting lesser false positives can be attributed to the fact that it provides a better representation of spam messages in this particular experiment setup.

Compared to the first experiment with just tokenization, second experiment with pretrained embedding layers performed better. This can be attributed to the fact that embeddings provide a better representation of long-term relationships between tokens.

References

[1] Almeida, Tiago A., José María G. Hidalgo, and Akebo Yamakami. "Contributions to the study of SMS spam filtering: new collection and results." Proceedings of the 11th ACM symposium on Document engineering. 2011.

[2] Hinton, Geoffrey. Neural Networks for machine learning online course. https://www.coursera.org/learn/neural-networks/home/welcome

[3] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

[4] "Embeddings, Transformers and Transfer Learning". https://spacy.io/usage/embeddings-transformers