**ADVANCED MACHINE LEARNING**

**TEXT AND SEQUENCE(GROUP-10)**

**ASSIGNMENT-4**

**Summary**

**Introduction**

The major goal of this experiment is to compare and analyze the effectiveness of two different Recurrent Neural Network (RNN) architectures when used for sentiment analysis on the IMDb movie review dataset. The first model under consideration uses randomly initialized word embeddings, whereas the second model incorporates pre-trained GloVe (Global Vectors for Word Representation) embeddings.

**Dataset**

The IMDb movie review dataset, which was used in this experiment, contains 50,000 movie reviews that have been annotated with sentiment labels indicating whether the review is positive or negative. Each review is represented as a series of integers, with each integer representing a word's index in a specified dictionary or vocabulary.   
  
To maintain consistency in the input data for model training, I standardized the length of the reviews by truncating or padding them to a fixed length of 150 words. This preprocessing phase is critical for ensuring input dimension consistency across all samples, allowing neural network models to be trained and evaluated more efficiently.

**Model Architecture**:

Using Keras, I created two RNN models: one with randomly initialized embeddings and another with pretrained embeddings.

**RNNModel:**

* EmbeddingLayer: Input length of 150, embedding dimension of 32.
* Bidirectional LSTM Layer (64 units, return sequences).
* DropoutLayer (rate = 0.5) to prevent overfitting.
* BatchNormalization Layer.
* Bidirectional LSTM Layer (32 units).
* DropoutLayer (rate = 0.5).
* BatchNormalization Layer.
* DenseLayer with sigmoid activation for binary classification.

**Pretrained RNN Model:**

* EmbeddingLayer initialized with GloVe embeddings (100-dimensional).
* Bidirectional LSTM Layer (64 units, return sequences).
* DropoutLayer (rate = 0.5). λ BatchNormalization Layer.
* Bidirectional LSTM Layer (32 units).
* DropoutLayer (rate = 0.5). λ BatchNormalization Layer.
* DenseLayer with sigmoid activation for binary classification.

**Performance of RNN Model with Random Embeddings:**

RNN Model with sample size 100

A graph with blue line and orange line

Description automatically generatedA graph with a line

Description automatically generated

RNN Model with sample size 500:

A graph with blue line and orange line

Description automatically generatedA graph of loss and loss

Description automatically generated

RNN Model with sample size 1,000:

A graph with a line and a line

Description automatically generatedA graph with a line and a line

Description automatically generated

**Performance of Pretrained RNN Model:**

Pre Trained RNN Model for 100 Training Samples :

A graph with blue lines and white text

Description automatically generated A graph with blue line and red line

Description automatically generated

Pre Trained RNN Model for 500 Training Samples :

A graph with blue and orange lines

Description automatically generated A graph of a graph

Description automatically generated

Pre Trained RNN Model for 1,000 Training Samples :

A graph with a line and a line

Description automatically generated A graph with orange lines and blue lines

Description automatically generated

**Results:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Embedding Technique** | **Maxlen** | **Training Sample Size** | **Loss and Accuracy** |
| RNN Model | 150 | 100 | loss: 0.6996 - accuracy: 0.5142 |
| RNN Model | 150 | 500 | loss: 0.6405 - accuracy: 0.6538 |
| RNN Model | 150 | 1000 | loss: 0.9805 - accuracy: 0.5616 |
| Pretrained RNN Model | 150 | 100 | loss: 0.6958 - accuracy: 0.5142 |
| Pretrained RNN Model | 150 | 500 | loss: 0.6940 - accuracy: 0.5242 |
| Pretrained RNN Model | 150 | 1000 | loss: 0.9805 - accuracy: 0.5616 |

**RNN Model:**

The RNN model exhibited stable training and validation accuracies throughout epochs, with both training and validation losses plateauing quickly, suggesting limited learning. However, the model showed improved performance compared to the 100-sample case. There was a slight decrease in accuracy compared to the 500-sample case, possibly indicating overfitting**.**

**Pre trained RNN Model:**

Both the RNN model with random embeddings and the pretrained model performed similarly, with only minor improvements detected in the pretrained model. Despite the use of pretrained embeddings, which typically encode semantic information from large text corpora, the pretrained model failed to outperform its counterpart using randomly initialized embeddings.   
  
This finding implies that the pretrained embeddings did not significantly improve the model's capacity to extract sentiment information from IMDb movie reviews when compared to the randomly initialized embeddings. Despite the potential advantages of using pretrained embeddings in NLP tasks like semantic understanding and generalization, their impact on sentiment analysis performance in this case appeared to be minimal.

The consistency in performance between the two models suggests that the pretrained embeddings did not add significant new information or capture domain-specific nuances crucial to sentiment analysis in movie reviews. This finding emphasizes the necessity of assessing the usefulness of pretrained embeddings in specific application domains, and also shows that alternative methodologies or revisions may be required to effectively use pretrained embeddings for sentiment analysis tasks on movie review datasets.

**Conclusion:**

The experiment found that increasing the training sample size enhanced the model's performance. Pretrained embeddings did not significantly improve model performance compared to random embeddings. Both models exhibited overfitting, particularly with larger training sample sizes. Experiments with different hyperparameters, architectures, and embedding sizes may improve performance.

**Key Points:**

**-**Two RNN models were compared for sentiment analysis of IMDb reviews: one with random embeddings and one with pre-trained GloVe embeddings.   
- Both models demonstrated consistent accuracies and modest learning, with training and validation losses immediately plateauing.   
- Performance improved with greater sample sizes, while overfitting happened more frequently with larger samples.   
- Pretrained embeddings did not significantly improve performance over random embeddings.   
- Experimenting with alternative hyperparameters and designs could produce better results.

**Recommendations:**

* Experiment with several word representations, such as Word2Vec or FastText, and tune pretrained embeddings during training to improve alignment with the IMDb dataset.
* Test more complex RNN architectures, employ techniques like dropout or L2 regularization to prevent overfitting, and include variants in reviews to boost dataset variety.
* Fine-tune hyperparameters such as learning rate and batch size, combine predictions from several models, and incorporate movie-related information to improve context understanding. These basic measures can improve the sentiment analysis performance on IMDb movie reviews.