Assignment 4: Text and Sequence Data

Introduction

This project aims to build a model that classifies movie reviews on the IMDB dataset as positive or negative. The analysis focuses on a subset of the data, considering only the top 10,000 most frequent words. Training will be conducted on various sample sizes (100, 500, 1000, and 100,000), with a dedicated validation set of 10,000 samples. Additionally, reviews will be cutoff after 150 words.

Key Challenge

The core question is: which word embedding method yields superior performance in sentiment classification?

Objective

This research aims to identify the most effective approach for sentiment analysis in the IMDB dataset, specifically predicting whether a movie review is positive or negative.

Data and Preprocessing

- The analysis utilizes the IMDB movie review dataset containing sentiment labels (positive or negative).
- Preprocessing involves converting reviews into word embeddings, restricting vocabulary to the top 10,000 words.
- Reviews are transformed into integer sequences, with each integer representing a unique word.

Technique: Two word embedding methods are compared

Custom-trained Embedding Layer:

A separate embedding layer is trained specifically on the IMDB review dataset.

Pre-trained Word Embedding Layer (GloVe):

- This popular model utilizes a large corpus of text data (Wikipedia and Gigaword 5) to train word embeddings, capturing semantic and syntactic relationships between words.
- The 6B version with 400,000 words and 6 billion tokens is employed.

Results summary table

Embedding technique	Training sample size	Accuracy	Loss
Custom-trained embedding layer	100	1.000	0.5365
	500	0.9725	0.4853
	1000	0.9787	0.4088
	10000	0.9772	0.0910
Pretrained word embedding layer (GloVe)	100	1.0000	0.0256
	500	0.9560	0.1141
	1000	0.9450	0.1673
	10000	0.8763	0.2925

Custom-trained Embedding Layer

- For a training sample size of 100, the custom-trained embedding layer achieved perfect accuracy (1.0000) with a loss of 0.5365.
- With an increase in training sample size, the accuracy remained consistently high, ranging from 0.9725 to 0.9787, and the loss decreased gradually from 0.4853 to 0.4088.

Pretrained Word Embedding Layer (GloVe)

- The pretrained word embedding layer consistently outperformed the custom-trained embedding layer across all training sample sizes.
- For a training sample size of 100, the pretrained word embedding layer achieved perfect accuracy (1.0000) with a significantly lower loss of 0.0256.
- As the training sample size increased, the accuracy decreased slightly, ranging from 0.8763 to 0.9560, and the loss decreased gradually from 0.02925 to 0.1141.

Recommendations

❖ **Use Pretrained Word Embeddings (GloVe)**

Pretrained word embedding layers, such as GloVe, consistently outperform custom-trained embedding layers in terms of accuracy and loss across all training sample sizes. Therefore, it is recommended to utilize pretrained word embeddings, like GloVe, for text classification tasks, especially when dealing with limited data.

- **Consider Data Augmentation Techniques:**

Implement data augmentation techniques to increase the size and diversity of the training data. Techniques such as data augmentation through translation, rotation, or adding noise to text data can help improve model generalization and performance, particularly with limited training data.

- **Explore Fine-Tuning Pretrained Embeddings:**

Explore fine-tuning pretrained word embeddings to adapt them to the specific domain of the dataset. Fine-tuning pretrained embeddings on the target dataset can further improve model performance by capturing domain-specific semantics.