schinta7@kent.edu

Assignment-4 Text and Sequence Data

AKHILA CHINTA-811308674

AIM

The goal of the binary classification problem for the IMDB dataset is to divide movie reviews into positive and negative categories. The dataset comprises 50,000 reviews; 10,000 words out of the top 10,000 are evaluated; training samples are restricted to 100, 5000, 1000, and 100,000 samples; validation is carried out on 10,000 samples. The data has been prepared. After that, the data is fed into a pretrained embedding model and the embedding layer, and various strategies are tried to gauge performance.

PREPARING THE DATA

- The dataset preparation process transforms each review into a set of word embeddings, where each word is represented by a fixed-size vector.
- This limits the number of samples to 10,000. Also, rather than using a string of words, a set of numbers representing individual words was generated from the reviews. Despite my having the list of numbers, the neural network's input is unsuitable for it.
- Tensors need to be constructed using numbers. One possible use for the integer list would be to create a tensor with samples and word indices of integer data type and form.
- For me to do that, I must ensure that every sample is the same length, which means I must use dummy words or numbers to ensure every review is the same length.

METHOD

For this IMDB dataset, I found two distinct methods for creating word embeddings.

1. Custom-trained embedding layer
2. pre-trained word embedding layer using the GloVe model.

In this work, we used the popular pretrained word embedding model GloVe, which is trained on a lot of textual data.

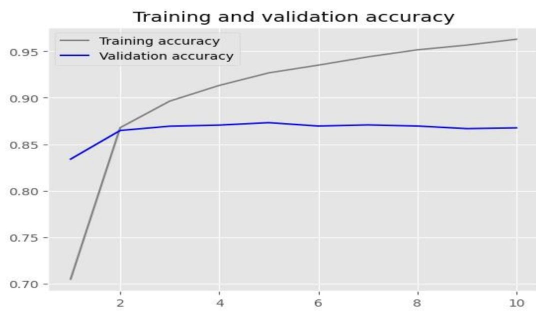
evaluated accuracy across sample sizes: 100, 5000, 1000, and 10,000 by comparing custom-trained and pretrained embedding layers on the IMDB dataset.

tested models using pretrained and custom-trained embeddings on IMDB reviews with different sample sizes, evaluating accuracy on test sets.

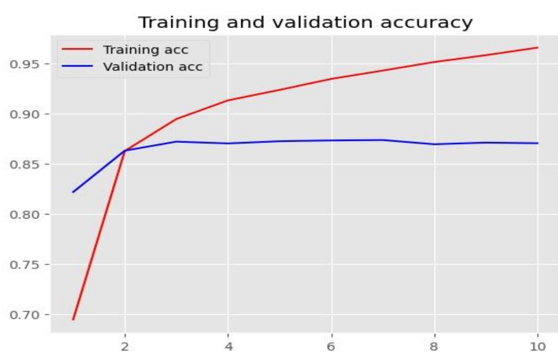
schinta7@kent.edu

CUSTOM-TRAINED EMBEDDING LAYER

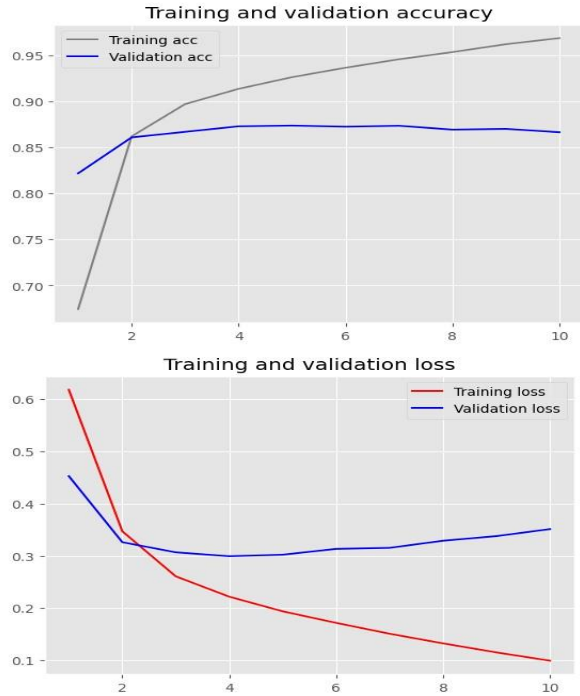
1. Custom-trained embedding layer with training sample size = 100



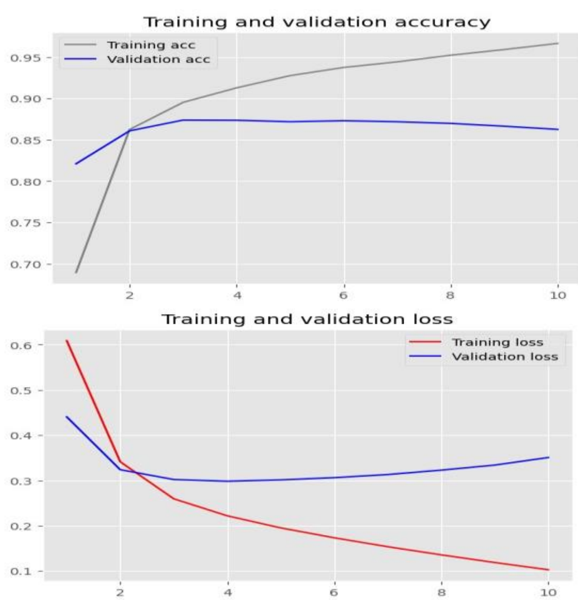
2. Custom-trained embedding layer with training sample size = 5000



3. Custom-trained embedding layer with training sample size = 1000



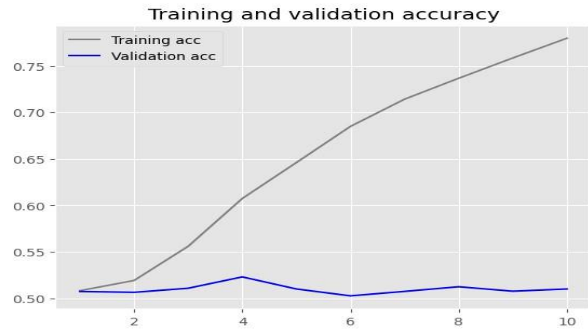
4. Custom-trained embedding layer with training sample size = 10000

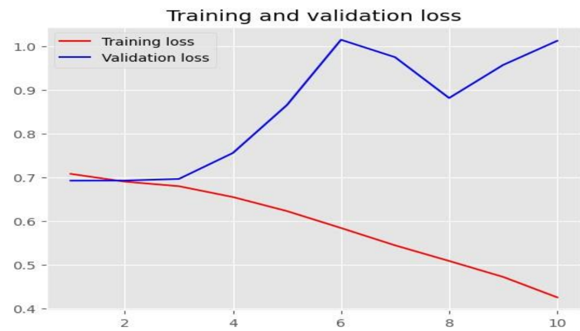


The precision of the specially trained embedding layer varied based on the size of the training sample, from 96.29-96.50. Training with a 1000-person sample size produced the highest accuracy.

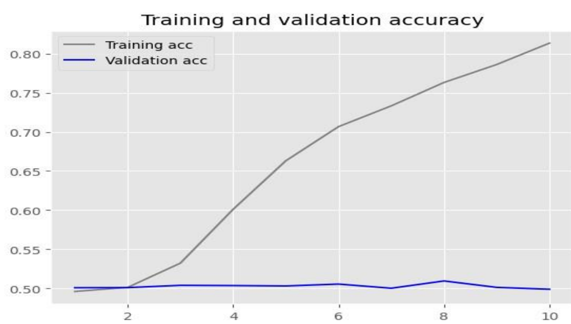
PRETRAINED WORD EMBEDDING LAYER

1. Pretrained word embedding layer with training sample size = 100

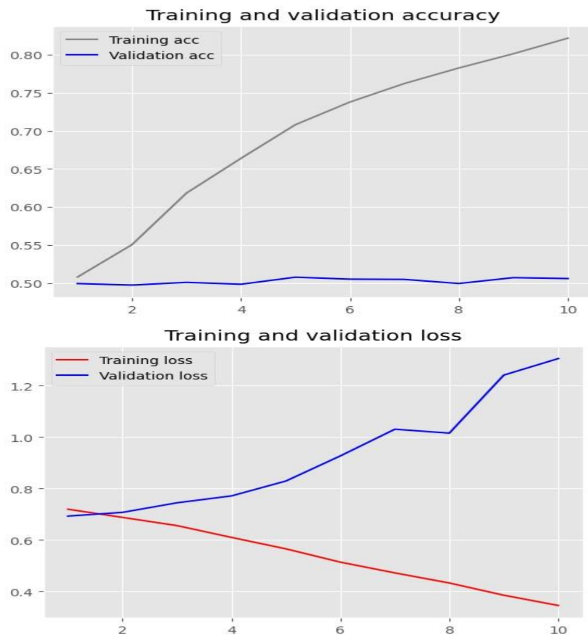




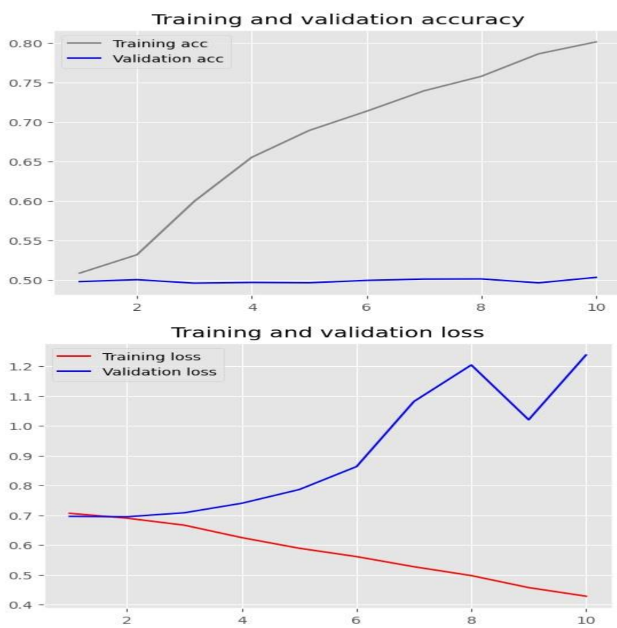
2. Pretrained word embedding layer with training sample size = 5000



3. Pretrained word embedding layer with training sample size = 1000



4. Pretrained word embedding layer with training sample size = 10000



GloVe, a word embedding technique that has been trained, has an accuracy range of 78.19-81.47. It peaks at 100 samples, but as sample sizes increase, it becomes overfit and loses accuracy. Task constraints influence which strategy is best, which leads to uncertainty.

RESULTS

Model	Embedding Technique	Training Sample Size	Training Accuracy (%)	Test loss
1	Custom-trained embedding layer	100	96.35	0.34
2	Custom-trained embedding layer	5000	96.29	0.34
3	Custom-trained embedding layer	1000	96.50	0.35
4	Custom-trained embedding layer	10000	96.37	0.34
5	Pretrained word embedding (GloVe)	100	81.47	1.21
6	Pretrained word embedding (GloVe)	5000	79.68	0.83
7	Pretrained word embedding (GloVe)	1000	78.19	1.11
8	Pretrained word embedding (GloVe)	10000	78.79	1.08

CONCLUSION

However, in this experiment, comparing the custom-trained embedding layer and pretrained word embedding layer, the first one performs better than the second, when training with more training sample numbers.