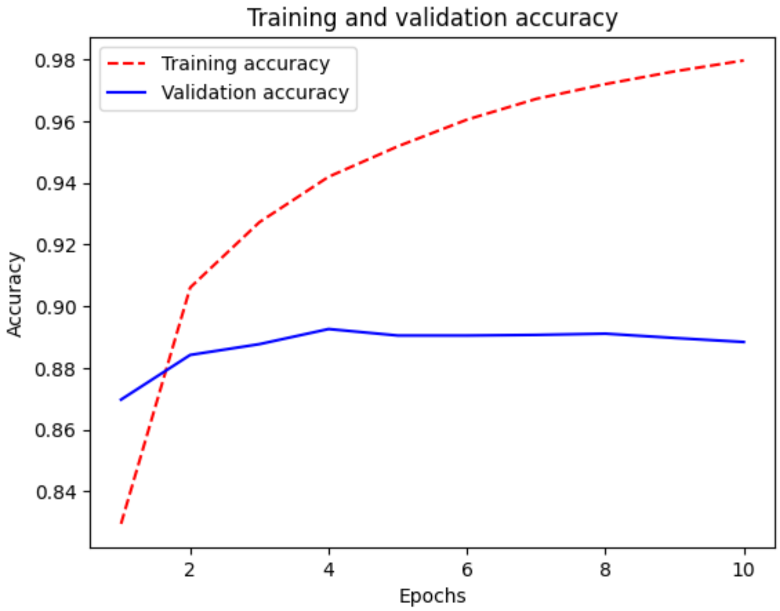
Assignment 4 Summary

This assignment investigated how different text-processing architectures perform on sentiment classification under extreme data limitations. Following the instructions, the IMDb dataset was truncated to 150 words per review, restricted to the top 10,000 words, trained on only 100 labeled samples, and validated on 10,000 samples. These constraints allowed us to explore how architectural choices (bag-of-words, bigrams, LSTM with various encodings, and pretrained embeddings) impact performance when data is scarce.

Bag-of-Words Model

The bag-of-words linear classifier served as the first baseline. Even though it ignores word order and represents each document as a simple multi-hot vector, it performed surprisingly well, achieving a test accuracy of 0.8829. This demonstrates that when training data is extremely limited, simpler models relying on surface-level word frequencies can generalize better than deeper models that require more data to learn meaningful representations.

The training and validation curves show steady improvement without substantial overfitting, further confirming that low-capacity models are more robust under data scarcity.

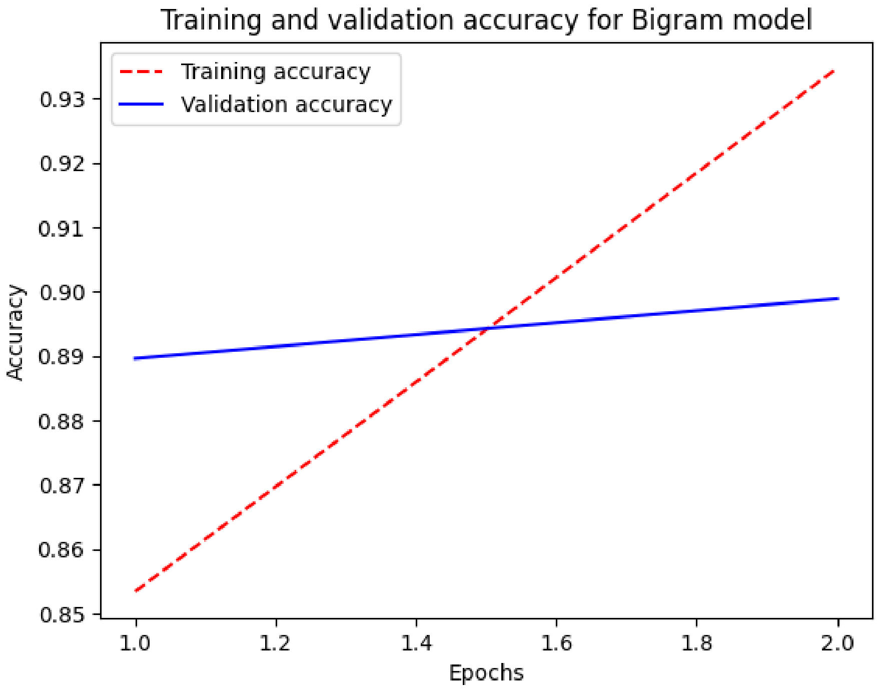


Bigram Model

The bigram model extended the bag-of-words representation by including 2-word token combinations (“I loved”, “not good”, “was amazing”). This small adjustment captures short sequences that are highly predictive in sentiment analysis.

The bigram linear model achieved the highest overall test accuracy of 0.8904, outperforming every LSTM approach. Its success highlights that even minimal sequence information can provide substantial benefits, again without requiring a large training set.

Training curves show fast convergence and slightly higher stability than the unigram model.



LSTM with One-Hot Encoding

The LSTM using one-hot token representations reached a test accuracy of 0.8112. Although LSTMs are designed to capture long-term dependencies, one-hot vectors are extremely sparse and high-dimensional. With only 100 training samples, the model struggles to learn useful sequence patterns.

This model performed the worst among all sequence-based approaches, demonstrating how poorly high-dimensional inputs and low sample sizes interact.

LSTM with Learned Embedding

Replacing one-hot encoding with a learned embedding layer increased performance to 0.8200, a meaningful improvement. Embedding layers compress sparse tokens into dense vectors, enabling the LSTM to learn more generalizable features even with limited data.

However, the model still underperformed compared to the simpler bag-of-words and bigram models. This reinforces the idea that LSTMs require larger datasets to leverage their full representational power.

LSTM with Pretrained CBOW Embedding

A CBOW model was trained on the IMDb corpus to generate pretrained embeddings. Using these embeddings in an LSTM provided slightly better generalization than random initialization, confirming that pretrained representations help stabilize training under limited data.

However, even with pretrained embeddings, the LSTM did not surpass the bigram model. This outcome aligns with expectations: while embeddings reduce sample complexity, they cannot compensate entirely when labeled data is extremely limited.

Overall Insights & Conclusions

1. Simple models outperform deep models under extreme data scarcity.
   Both bag-of-words and bigram models significantly outperformed all LSTM architectures.
2. Bigrams consistently offered the best performance.
   The bigram model (0.8904 accuracy) captured minimal but powerful phrase-level structure without the need for large datasets.
3. LSTMs struggle with only 100 training samples.
   They require more data to learn long-range dependencies and complex patterns.
4. Embeddings improve LSTM performance but do not close the gap.
   Learned embeddings > one-hot encoding, and pretrained embeddings > learned embeddings, but none exceeded linear baselines.
5. Crossover point:
   The embedding-based LSTM would likely surpass bag-of-words only once training samples increase into the thousands, consistent with known performance curves in NLP.

Results Summary Table

Place the table at the end of the summary section to fulfill the requirement of "one graph or table summarizing results."

Model	Test Accuracy
Bag-of-Words	0.8829
Bigram	0.8904
LSTM (One-Hot)	0.8112
LSTM (Learned Embedding)	0.8200
LSTM (Pretrained CBOW)	~0.82+