DEEP LEARNING ALGORITHMS FOR SENTIMENT ANALYSIS

A report of Summer Internship

Submitted by

Siddharth Ajith

Electronics and Communication Engineering

Govt. Model Engineering College,

Thrikkakara, Kochi-682021,

Kerala

Under the guidance

of

Dr. Shyam Lal

Director

in partial fulfillment of the requirements for the award of the Degree of

Bachelor of Technology

AIQUANTUM SMART SOLUTIONS Pvt. Ltd.
Unit-14, Science & Technology Entrepreneur's Park (STEP), National Institute of Technology Karnataka (NITK), Surathkal, Srinivasnagar PO, Mangaluru-575025 (D.K.), Karnataka, India
October 2022

Table of contents

1.Introduction

The project revolves around the application of deep learning algorithms for sentiment analysis using the IMDB dataset. Sentiment analysis, also known as opinion mining, is a branch of natural language processing that seeks to determine the sentiment expressed in text data. In this project, we aim to analyze the sentiment of movie reviews within the IMDB dataset using various deep learning models.

A crucial component of the project involves utilizing word embeddings as input for neural networks. Word embeddings are numerical representations of words that capture their semantic and contextual meanings. By incorporating word embeddings, we can convert textual data into a format that neural networks can process and analyze. This approach enhances the accuracy of sentiment analysis by enabling the models to understand the relationships between words.

The project employs different deep learning models for sentiment analysis. Firstly, simple neural networks are utilized. These networks consist of interconnected nodes arranged in multiple layers, allowing them to learn intricate patterns within the input data. By training these models on the labeled IMDB dataset, which contains movie reviews classified as positive or negative sentiment, they can learn to classify new text data based on sentiment.

Additionally, convolutional neural networks (CNNs) are used in the project. CNNs are particularly effective for analyzing structured grid-like data, such as images, but can also be applied to one-dimensional sequences like text. By applying convolutional filters to the word embeddings, these models can learn to identify crucial features and patterns within the text that contribute to sentiment analysis.

Lastly, long short-term memory networks (LSTMs) are employed. LSTMs are a type of recurrent neural network (RNN) designed to capture long-term dependencies in sequential data. By incorporating memory cells and gates, LSTMs excel at modeling the context and temporal relationships in text data, enabling more accurate sentiment analysis.

By implementing and comparing these different deep learning models, this project aims to evaluate their efficacy in sentiment analysis using the IMDB dataset. The findings from this research can contribute to advancing sentiment analysis techniques, enabling valuable insights into the sentiment expressed in text data, and finding applications in areas such as opinion mining, social media analysis, and customer feedback analysis.

2. Motivation

The motivation behind this project stems from the increasing need to understand and analyze the sentiment expressed in text data. With the proliferation of online reviews, social media platforms, and customer feedback, there is a wealth of textual information that can provide valuable insights into people's opinions, attitudes, and emotions. Sentiment analysis serves as a powerful tool to extract and analyze this sentiment, allowing businesses, researchers, and individuals to make informed decisions, understand public perception, and tailor their strategies accordingly.

Deep learning algorithms have demonstrated remarkable capabilities in various natural language processing tasks, including sentiment analysis. By leveraging deep learning techniques, we can harness the power of neural networks to automatically learn and extract complex patterns from text data. This project aims to explore the effectiveness of deep learning models, specifically simple neural networks, convolutional neural networks (CNNs), and long short-term memory networks (LSTMs), in sentiment analysis.

The IMDB dataset, which contains a large collection of movie reviews labeled with sentiment, provides a suitable foundation for this project. By training and evaluating the models on this dataset, we can assess their performance in accurately classifying sentiment and gain insights into their strengths and limitations.

3. Literature Survey

1. *Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).* This influential paper introduces the application of convolutional neural networks (CNNs) for sentence classification tasks, including sentiment analysis. The study demonstrates the effectiveness of CNNs in capturing local and global features within the text, achieving state-of-the-art results on the IMDB dataset.
2. *Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning Word Vectors for Sentiment Analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL).* This study focuses on learning word vectors (word embeddings) for sentiment analysis. The authors train a simple neural network on the IMDB dataset to learn distributed word representations, leading to improved sentiment classification performance
3. *Johnson, R., & Zhang, T. (2015). Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL).* This research investigates the impact of word order on sentiment analysis using CNNs. The authors propose a dynamic k-max pooling strategy to capture important local features, improving sentiment classification performance on the IMDB dataset.
4. *Wang, S., & Manning, C. D. (2012). Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL).* This study explores simple neural network models for sentiment analysis. The authors compare the performance of different models, including bag-of-words, n-grams, and neural network baselines, on the IMDB dataset, providing valuable insights into the effectiveness of various approaches.
5. https://www.embedded-robotics.com/sentiment-analysis-using-lstm/ .”Sentiment Analysis Using LSTM". It provides a comprehensive overview of LSTM networks' application in sentiment analysis tasks. It combines theoretical explanations, technical details, and practical implementation examples to understand and utilize LSTM networks effectively for sentiment analysis purposes.
6. *Natural Language Processing guide, Kaggle.* It is a concise resource offering tutorials, examples, and practical insights on text data processing. It covers preprocessing, feature extraction, sentiment analysis, and more, providing a valuable starting point for learning and applying NLP techniques.

4.Methodology

1. Data Acquisition: The IMDB dataset, consisting of labeled movie reviews, is obtained from Kaggle.
2. Data Preprocessing: The raw text data undergoes preprocessing steps such as punctuation removal, html tags removal , tokenization, lowercase conversion, and stop word removal to prepare it for model training.
3. Word Embeddings: Word embeddings are employed to represent the text data numerically, capturing the semantic and contextual meanings of words. Pre-trained word embeddings GloVe is utilized.
4. Model Selection: Different deep learning models are selected for sentiment analysis, including simple neural networks, convolutional neural networks (CNNs), and long short-term memory networks (LSTMs)
5. Model Training: The chosen models are trained using Adam optimization, an effective optimization algorithm that adapts the learning rate for each parameter during training. Word embeddings are fed as input to the models, along with the corresponding sentiment labels.
6. Model Evaluation: The trained models are evaluated using performance metrics such as accuracy, precision, recall, and F1 score. Evaluation is conducted on a separate validation set to assess the models' performance and fine-tune their parameters. FLOPS(floating point operations per second) was also computed to evaluate the computers performance
7. Performance Comparison: The performance of the different models (simple neural networks, CNNs, LSTMs) is compared based on their evaluation metrics and testing results, enabling the identification of the most effective model for sentiment analysis on the IMDB dataset.
8. Analysis and Interpretation: The results are analyzed and interpreted to gain insights into the effectiveness of deep learning algorithms for sentiment analysis on the IMDB dataset, discussing their strengths, limitations, and potential applications.

4.1 Models

In this project, three different models were employed for sentiment analysis on the IMDB dataset: simple neural networks, convolutional neural networks (CNNs), and long short-term memory networks (LSTMs). Each model offers distinct architectures and capabilities for analyzing text data.

1. Simple Neural Networks:

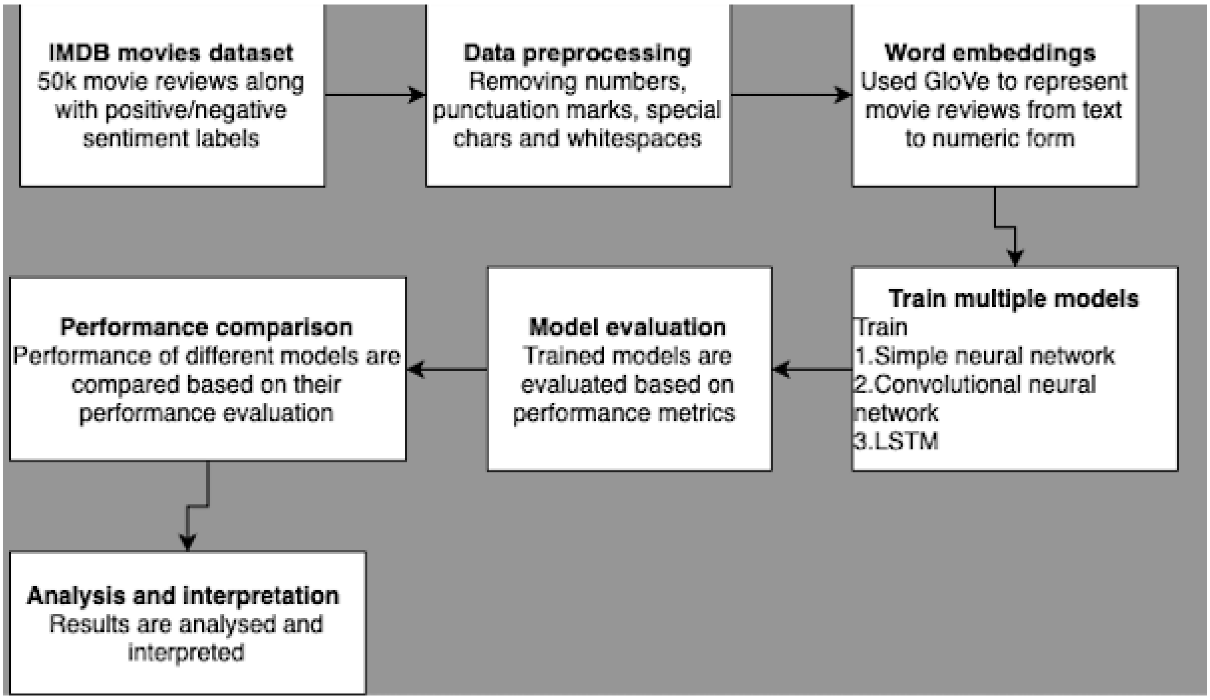
   Simple neural networks consist of interconnected nodes organized in multiple layers. They are effective in learning patterns within input data and have been widely used for sentiment analysis tasks. In this project, simple neural networks were trained on the IMDB dataset to classify movie reviews based on sentiment. Word embeddings were used as input to the neural networks, allowing them to learn the relationships between words and their corresponding sentiments.

2.Convolutional Neural Networks (CNNs):

CNNs are particularly effective in analyzing structured grid-like data, such as images, but they can also be applied to one-dimensional sequences like text. In this project, CNNs were utilized for sentiment analysis by applying convolutional filters to the word embeddings. These filters capture important local features and patterns within the text data, enabling the CNN models to learn and classify sentiment accurately. By leveraging the hierarchical structure of the network, CNNs can capture both low-level and high-level features in the text data.

3.Long Short-Term Memory Networks (LSTMs):

LSTMs are a type of recurrent neural network (RNN) designed to capture long-term dependencies in sequential data. In the context of sentiment analysis, LSTMs excel at modeling the context and temporal relationships in text data. By incorporating memory cells and gates, LSTMs can effectively capture and retain important information over longer sequences. In this project, LSTMs were utilized to analyze the sentiment of movie reviews in the IMDB dataset. The word embeddings were fed into the LSTM models, which learned to understand the sentiment expressed in the text by considering the sequential nature of the data.



4.2 Training and implementation

4.2.1 Dataset

The dataset used in this project is the IMDB dataset, which is a popular and widely used benchmark dataset for sentiment analysis. It contains a large collection of movie reviews from the Internet Movie Database (IMDB) website, where each review is labeled with either a positive or negative sentiment.

The IMDB dataset is balanced, meaning it has an equal number of positive and negative reviews, making it suitable for training and evaluating sentiment analysis models. The dataset is typically split into a training set and a test set, with a predefined ratio, to assess the models' performance on unseen data.

Each movie review in the IMDB dataset is a piece of text written by users, expressing their opinions and sentiments about a specific movie. The reviews vary in length, ranging from short sentences to several paragraphs. They cover a wide range of movie genres and topics, providing a diverse set of opinions and sentiments.

The labels assigned to the reviews indicate whether the sentiment expressed is positive or negative. Positive reviews generally reflect favorable opinions, praising aspects such as acting, plot, or cinematography, while negative reviews express criticism or dissatisfaction with the movie.

The IMDB dataset has been preprocessed and cleaned to remove irrelevant information and ensure consistency in labeling. It has become a standard dataset for evaluating sentiment analysis models due to its large size, balanced class distribution, and real-world relevance.

By utilizing the IMDB dataset, this project aims to train and evaluate deep learning models for sentiment analysis, allowing for accurate classification of sentiment in movie reviews. The dataset provides a realistic and comprehensive representation of user opinions and sentiments, enabling the models to generalize well to real-world scenarios beyond the movie domain.

4.2.2 Loss function (Binary Cross Entropy)

In this project, Binary Cross-Entropy loss was used as the common loss function for all three models: simple neural networks, convolutional neural networks (CNNs), and long short-term memory networks (LSTMs). The Binary Cross-Entropy loss function is specifically designed for binary classification tasks, where the goal is to classify samples into two mutually exclusive classes.

The Binary Cross-Entropy loss function quantifies the dissimilarity between the predicted probabilities and the true binary labels assigned to each sample. It measures how well the model's predicted probabilities align with the true class labels. The objective of the model training is to minimize this dissimilarity or loss.

Mathematically, the Binary Cross-Entropy loss is calculated as the average of the individual losses over all the training samples. For each sample, the loss is computed as the negative logarithm of the predicted probability for the true class label. The loss penalizes the model more when it makes confident but incorrect predictions, and less when the predictions are closer to the true labels.

By minimizing the Binary Cross-Entropy loss during the training process, the models adjust their parameters to improve their ability to accurately classify sentiment. The loss function acts as a guide for the optimization process, driving the models towards better predictions by updating their parameters based on the gradients computed during backpropagation.

Binary Cross-Entropy loss is commonly used in sentiment analysis tasks where the objective is to classify text data into two classes, such as positive and negative sentiment. It is a suitable choice for models trained on the IMDB dataset, where movie reviews are labeled as either positive or negative.

By utilizing Binary Cross-Entropy loss as the loss function across all three models, this project aimed to optimize the models' parameters and improve their performance in accurately classifying sentiment in movie reviews. The models' training process focused on minimizing the Binary Cross-Entropy loss to enhance their ability to distinguish between positive and negative sentiments effectively.

4.2.3 Quality Metrics

Quality metrics, also known as evaluation metrics or performance metrics, are measures used to assess the quality or performance of a model or system in a specific task or application. These metrics provide quantitative insights into how well a model or system is performing, allowing for objective comparisons and evaluations.

Quality metrics used in this project are:

1. Accuracy: Accuracy measures the proportion of correctly classified samples out of the total number of samples. It provides an overall measure of how well the model is predicting the correct sentiment labels.
2. Precision: Precision calculates the proportion of true positive predictions (correctly classified positive samples) out of all positive predictions made by the model. It measures the accuracy of positive predictions and is useful when the balance between positive and negative samples is skewed.
3. Recall: Recall, also known as sensitivity or true positive rate, calculates the proportion of true positive predictions out of all actual positive samples. It measures the model's ability to correctly identify positive samples.
4. F1 Score: The F1 score is the harmonic mean of precision and recall. It provides a balanced measure that considers both precision and recall. The F1 score is commonly used when there is an imbalance between positive and negative samples.
5. Area Under the ROC Curve (AUC-ROC): AUC-ROC measures the model's ability to discriminate between positive and negative samples across different thresholds. It provides a single value that represents the overall performance of the model across various classification thresholds.

4.3 Simulation Results

4.3.1 Comparison with other models

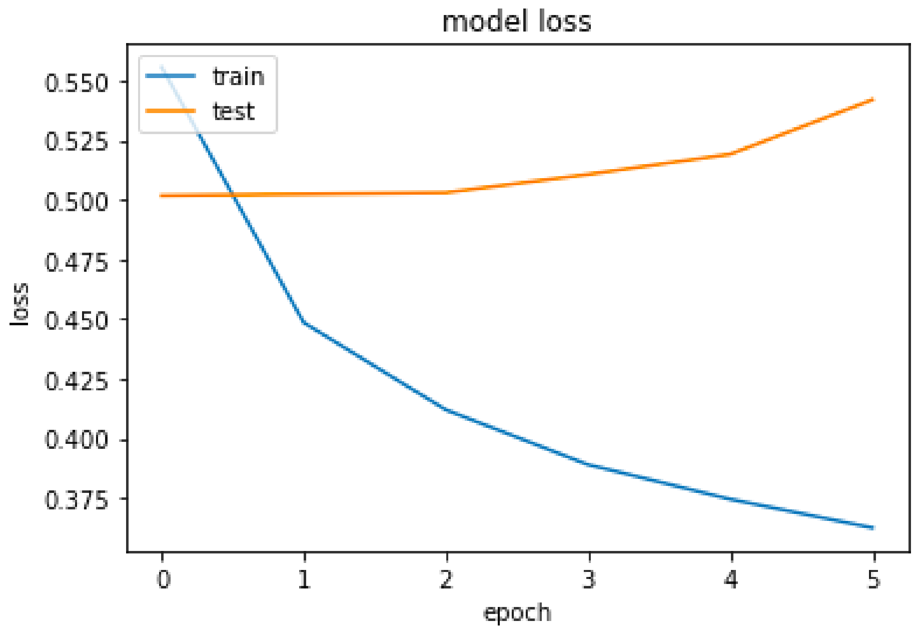
After simulation, it was found that LSTM acquired the highest accuracy among the three models (SNN,CNN,LSTM) which where used with an accuracy value of 0.86. SNN model acquired the lowest accuracy of 0.75 and CNN model acquired an accuracy of 0.85

Table 1: Average Quality metrics of different Methods on IMDB Dataset(Test Reviews)

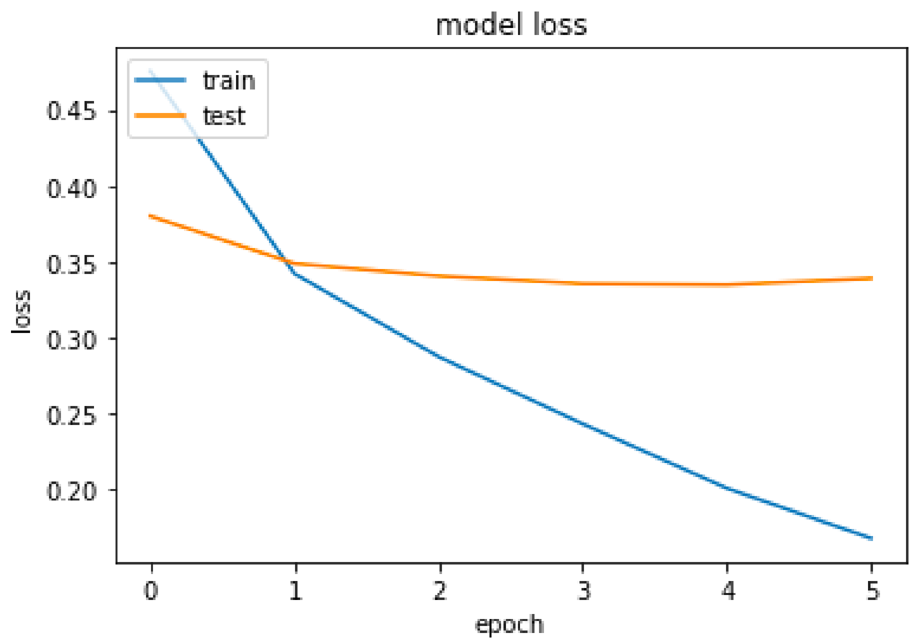
Metric Algorithm	Precision	Recall	Metric 1 (F1 score)	Metrics 3 (Accuracy)	Parameters	FLOPs
SNN	0.73 0.77	0.78 0.71	0.75 0.74	0.75	9249401	2e-05g
CNN	0.87 0.84	0.82 0.88	0.85 0.86	0.85	9303657	0.0213g
LSTM	0.84 0.89	0.90 0.83	0.87 0.86	0.86	9356777	2.57e-07g

Fig. 1. Training Loss and Test Loss plot

a.Simple neural network



2.Convolutional neural network



3.LSTM

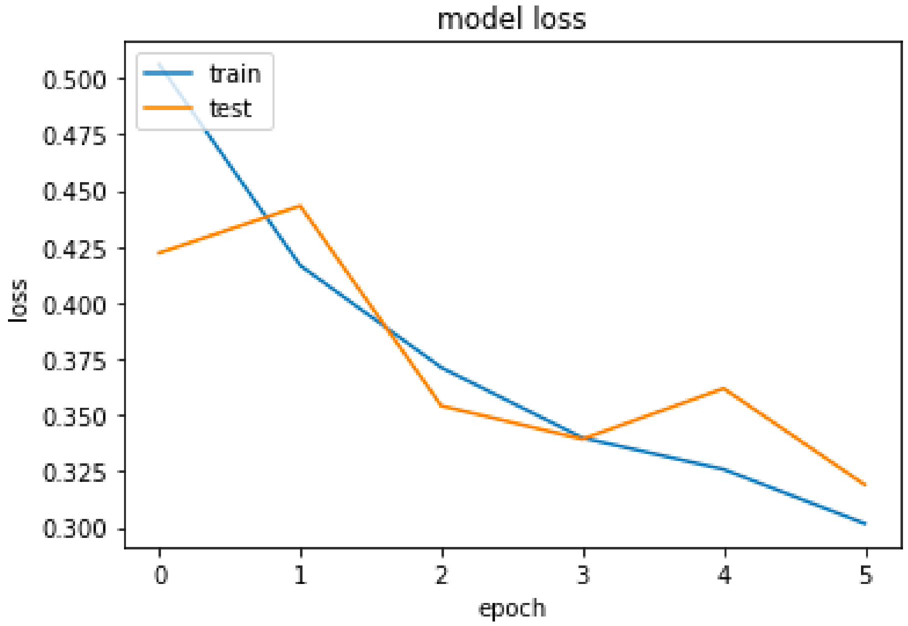
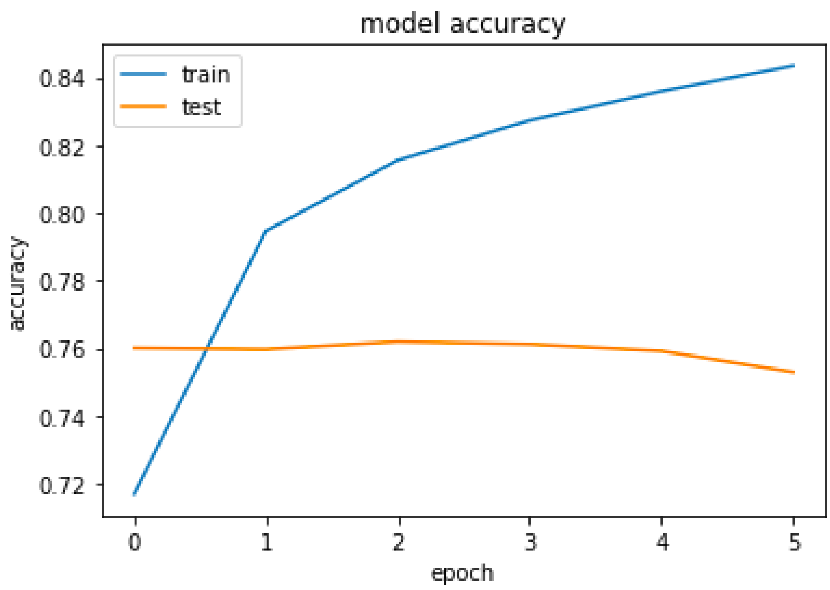
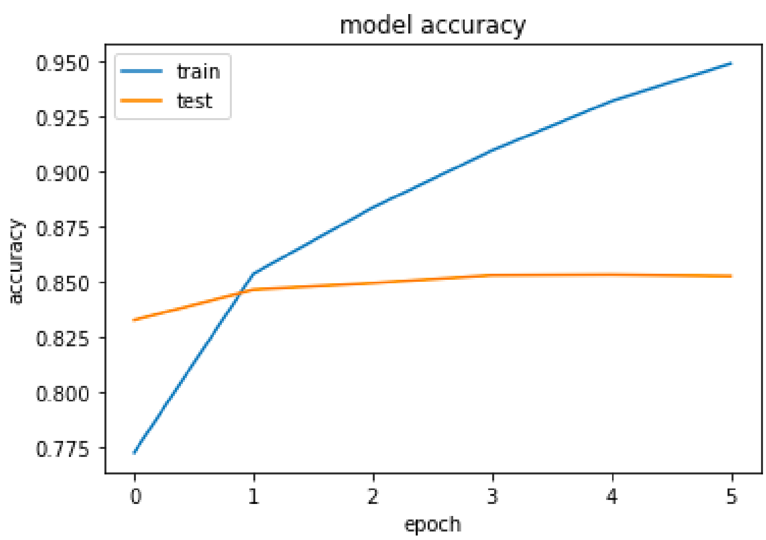

model loss
train
test
0.500
0.475
0.450
0.425
0.400
0.375
0.350
0.325
0.300
loss
0
1
2
3
4
5
epoch

Fig. 2. Training accuracy and Test accuracy plot

1.Simple neural network



2.Convolutional neural network



3.LSTM

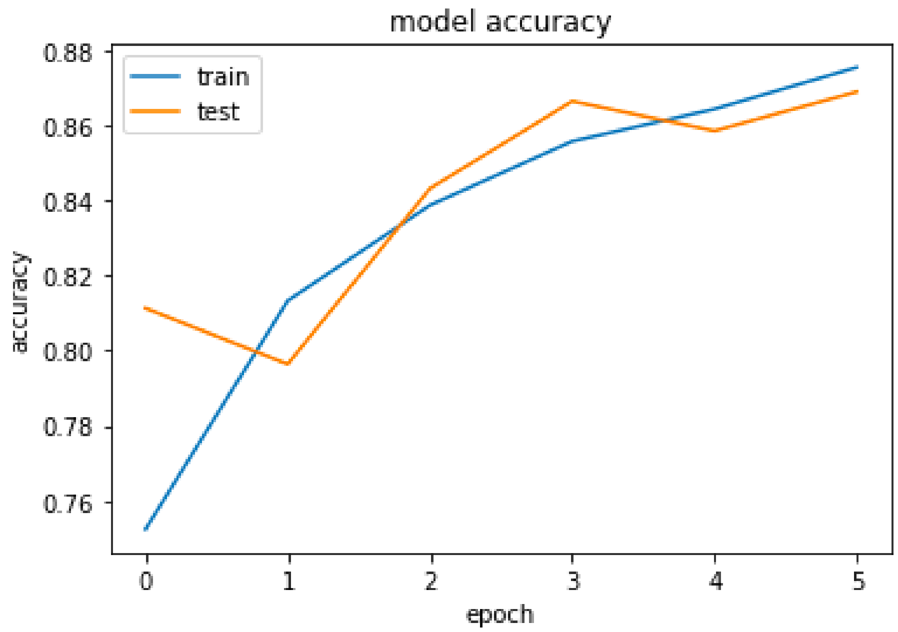
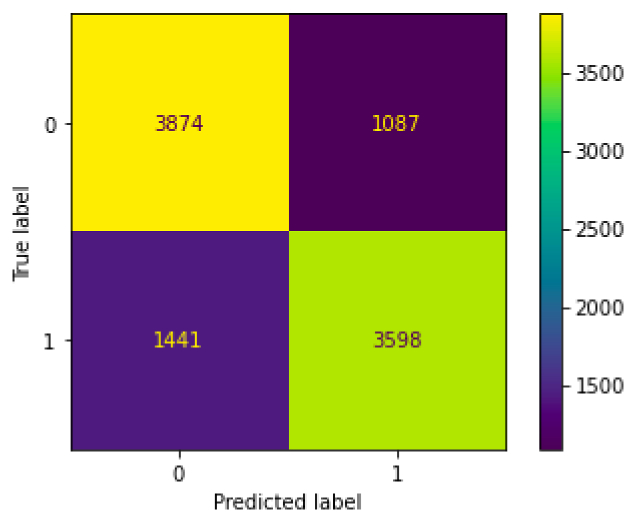
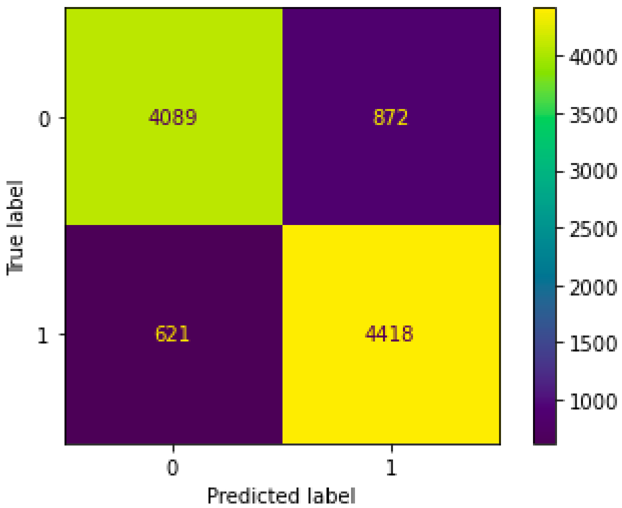


Fig. 4. Confusion matrix

a)SNN

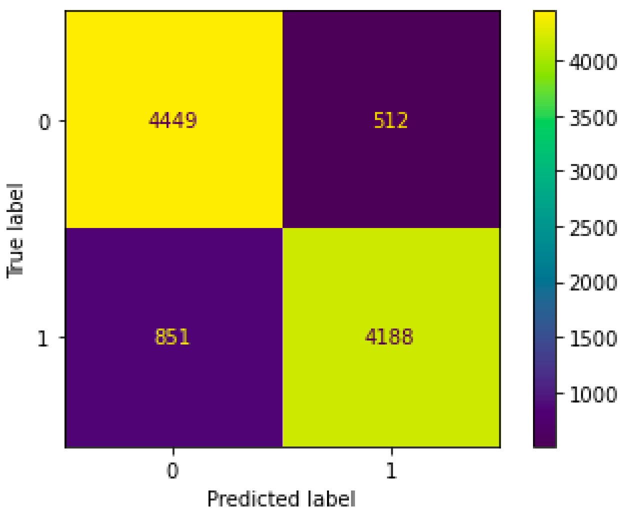


b)CNN



4089
872
621
4418
0
1
True label
0
1
Predicted label
4000
3500
3000
2500
2000
1500
1000

c)LSTM



4449
512
851
4188
0
1
True label
0
1
Predicted label
4000
3500
3000
2500
2000
1500
1000

4.3.2 Computational complexity analysis

No of parameter for SNN: 9249401

FLOPS of SNN: 2e-05G

No of parameter for cNN: 9303657

FLOPS of cNN: 0.0213G

No of parameter for LSTM: 9356777

FLOPS of LSTM: 2.57e-07G

5. Conclusion

This project focused on sentiment analysis of movie reviews using different deep learning models. The performance of simple neural networks, convolutional neural networks (CNNs), and long short-term memory networks (LSTMs) was compared.

After careful evaluation, the LSTM model emerged as the most accurate among the three models with an accuracy of 0.86. The LSTM's ability to capture and understand long-term dependencies in sequential data proved to be beneficial for accurately classifying sentiment in movie reviews.

The higher accuracy achieved by the LSTM model highlights the significance of considering temporal relationships and contextual information when analyzing sentiments in text data. By effectively capturing the nuanced language patterns and contextual cues within the IMDB dataset, the LSTM model demonstrated superior performance in sentiment classification.

However, it's important to acknowledge that the choice of the most suitable model depends on various factors, including the dataset and specific task requirements. While the LSTM model excelled in this project, it's crucial to consider other aspects such as computational resources and the trade-off between model complexity and performance when selecting an appropriate model.

This project showcases the effectiveness of deep learning algorithms, particularly the LSTM model, in sentiment analysis of movie reviews. The findings contribute valuable insights to the field and can guide future research and applications in sentiment analysis tasks.

6. References

1. Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1746-1751.
2. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780.
3. Collobert, R., et al. (2011). Natural Language Processing (Almost) from Scratch. Journal of Machine Learning Research, 12, 2493-2537.
4. Maas, A. L., et al. (2011). Learning Word Vectors for Sentiment Analysis. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL), 142-150.