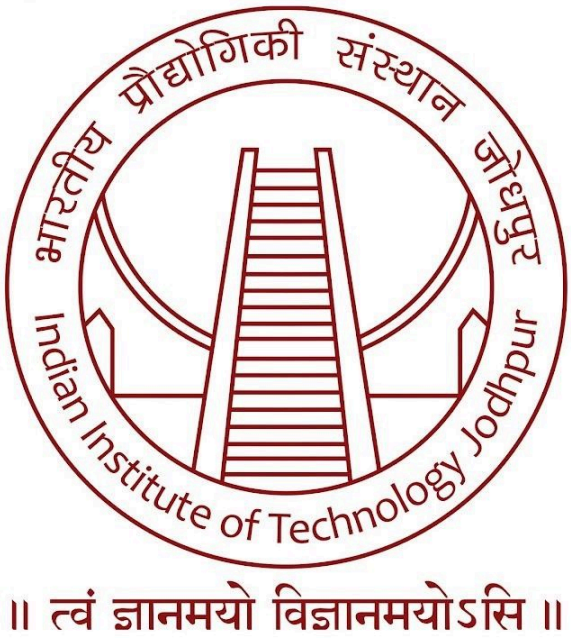
Project : Deep Learning

IMDB Movie DataSet



Team Members:

Samaksh Verma (B21AI037)
Ayush Jain (B21BB006)
Kartikay Yadav (B21BB014)
Rohan Galgat (B21AI030)

1.Problem Statement

Problem Statement

The objective of this project is to construct and authenticate a deep learning model capable of effectively forecasting the sentiment of movie reviews based on a dataset including 50,000 entries. The objective is to classify each review as either positive or negative through the analysis of the available textual data.

2. Solution strategy

LSTM has gained significant popularity in the field of sentiment classification. The Long Short-Term Memory (LSTM) model was introduced by Hoch Reiter and Schmid Huber in 1997 and has since been further developed and widely adopted by other researchers. They exhibit exceptional efficacy in addressing diverse and extensive problem sets and have gained widespread adoption. LSTMs are specifically engineered to disregard the issue of long-term

reliance. Retaining information for an extended duration is essentially their innate behavior, rather than a task they find challenging to acquire. The structure of all recurrent neural networks consists of a series of recurring modules inside the neural network architecture. At the level of recurrent neural networks (RNNs), the recurring module exhibits a straightforward structure, characterized by a solitary tanh layer. The experimental experiments in our research utilize the IMDB benchmark dataset, which comprises movie reviews categorized as either positive or negative.

To effectively apply LSTM, LSTM+CNN, and BERT models for sentiment analysis on the IMDb dataset, you can adopt the following solution strategies for each approach:

1. ***Using LSTM for Sentiment Analysis***

Preprocessing:
Tokenization: Convert text into tokens since LSTM models require numerical input.
- Padding: Ensure all input sequences are the same length for batch processing.
- Embedding:Use embeddings (like GloVe or Word2Vec) to transform tokens into meaningful vector representations.

Model Architecture:
-LSTM Layer: Implement one or more LSTM layers to process the sequence data.
- Dropout: Include dropout layers to prevent overfitting.
- Dense Layer: A fully connected layer followed by a classification layer, typically with a sigmoid activation function for binary classification.

Training:
- Loss Function: Use binary cross-entropy as the loss function.
- Optimizer: Employ optimizers like Adam or RMSprop.
- Epochs and Batch Size: Select appropriate epochs and batch sizes based on the training behavior.

2. **Using LSTM+CNN for Sentiment Analysis**

Preprocessing:
- Similar preprocessing steps as LSTM, ensuring data is tokenized, padded, and embedded.

Model Architecture:
- Convolutional Layer: Start with one or more convolutional layers to extract local features from the text.
- Pooling Layer: Use max pooling to reduce the dimensionality of the data.
- LSTM Layer: Follow CNN layers with LSTM layers to capture dependencies in the sequence data.
- Dense and Output Layers: End with a fully connected dense layer and a sigmoid output layer for binary classification.

Training:
- Training strategy similar to LSTM, ensuring the use of dropout to mitigate overfitting and adjusting learning rates as necessary.

3. **Using BERT for Sentiment Analysis**

Preprocessing:
- Tokenizer:Use BERT's tokenizer to handle splitting of text into tokens and convert them into IDs understood by BERT.
- Special Tokens:Add special tokens required by BERT, such as [CLS] at the beginning and [SEP] at the end of each review.

Model Architecture:
- BERT Model: Utilize the pre-trained BERT model available from libraries like Hugging Face's Transformers.
- Additional Layers:Optionally add one or more dense layers on top of BERT if fine-tuning is needed for better performance.
- Output Layer: Employ a sigmoid activation function for the final output to predict sentiment.

Training:
- Fine-tuning: Since BERT is already pre-trained, focus on fine-tuning the model with a lower learning rate to adapt to the sentiment analysis task.
- Loss and Optimizer: Use of binary cross-entropy loss and a suitable optimizer, like AdamW, which is specifically adjusted for BERT.

General Strategy

- Evaluation: Regularly evaluate the model using a validation set and adjust parameters as needed based on performance metrics like accuracy and F1-score.
- Experimentation: Continuously experimenting with different architectures, the number of layers and hyperparameters to find the optimal setup for your specific dataset and task requirements.

Preprocessing Text

The NLTK library, implemented in Python, is utilized for the purpose of preprocessing textual data in order to facilitate natural language processing jobs. The process commences with the acquisition of the stopwords corpus, comprising frequently encountered words that frequently lack substantial semantic significance, such as "the" and "is". Subsequently, these stopwords are placed into a collection that is specifically tailored for the English language. The function preprocess_text is designed to accept a given textual input. The function in question involves the conversion of the text to lowercase and the tokenization of the text into individual words.

Following that, stopwords are eliminated from the tokenized text by a process of list comprehension, leading to the generation of a comprehensive list of filtered tokens that specifically omit stopwords. Ultimately, the tokens that have undergone filtration are consolidated into a unified string, and this processed text is subsequently returned by the function. The preprocessing stage holds significant importance in natural language processing (NLP) jobs as it serves to improve the analysis or modeling quality by deleting extraneous words that may not contribute to the overall semantic content of the text.

3. Dataset

IMDB dataset having 50K movie reviews for natural language processing or Text analytics. This is a dataset for binary sentiment classification containing substantially more data than previous benchmark datasets. We provide a set of 25,000 highly polar movie reviews for training and 25,000 for testing. So, predict the number of positive and negative reviews using either classification or deep learning algorithms.

The IMDb movie reviews dataset serves as a standard benchmark for evaluating the performance of various NLP techniques, including traditional machine learning methods like Support Vector Machines (SVM) and Naive Bayes, as well as modern deep learning approaches such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), LSTMs, and Transformer models like BERT.

Researchers often utilize this dataset to demonstrate improvements in accuracy, efficiency, and understanding of different architectures and learning paradigms in sentiment analysis.

4. Major innovations/ contributions

We created our own models for training and testing. We have created the following models and implemented them with input of custom statements.

a. LSTM+CNN
b. BERT
c. LSTM
d. CustomLSTM implemented from Research Paper
e. RNN

f. Using sci-kit learn library+RandomForest

Innovations

1. Bad words detection model: We have created a way to eliminate reviews from dataset that contain abusive words. THis would leads to clean and fair judgment of reviews in any case
2. LSTM implementation with CNN: We have modified the approach the LSTM model for sentiment analysis such that pattern recognition of spatial data in the text is done well and LSTm can selectively read, write and forget the data of kernels of different size.
3. Customize the tokenization of text in the data using different techniques.
4. Created a pipeline to train,test and predict the the model on new review data that can be used easily for different models and data in future.
5. Tested dataset on different models that are potentially the best solution for performing the given task. We have also perform transfer learning using BERT model.

Contributions

1. Ayush Jain(B21BB006):
    - Designing and implementing LSTM +CNN model, RNN Model and custom LSTM.
    - Performing custom tokenization
    - Reading the research papers and Ideation of creating the models
    - Creating all the Reports and data processing
    - HyperParameter tuning and dropout layer addition
    - Debugging

2. Samaksh Verma(B21AI037):
    - Implementation of BERT model, LSTM model and scikit learn pipeline using random forest
    - Reading research papers and ideation for model creation
    - Optimization and training of models
    - Creating all the Reports
    - Debugging
    - Data visualization and processing

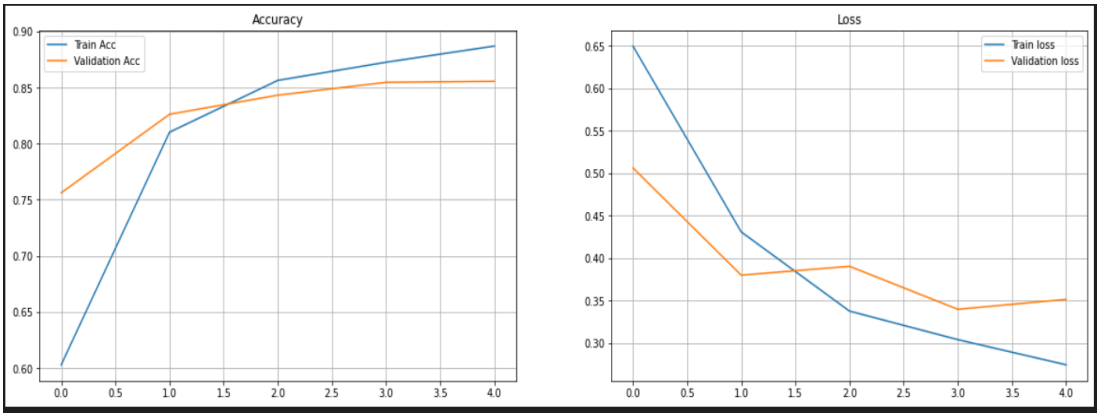
3. Kartikay yadav(B21BB014):
    - Data cleaning and report making
    - Model development discussion
    - Parameter tuning

4. Rohan Galgat(B21AI):
    - Data cleaning and report making
    - Data visualization
    - Hyperparameter tuning

5. Results

>>We have the following accuracies and prediction for the model/approaches:

1. Random Forest with sci-kit learn:
    - Accuracy : 85.92

2. Custom RNN model:
    - Accuracy : 85%
    - Loss curves
    - 

    - Prediction:

```
index = 32
print(df['review'][index])
print('='*70)
print(f'Actual sentiment is  : {df["sentiment"][index]}')
print('='*70)
pro = predict_text(df['review'][index])
status = "positive" if pro > 0.5 else "negative"
pro = (1 - pro) if status == "negative" else pro
print(f'predicted sentiment is {status} with a probability of {pro}')
```

```
My first exposure to the Templarios & not a good one. I was excited to find this title among the offerings from Anchor Bay Video, which has brought us other cult classics such as "Spider Baby". The print quality is excellent,
======================================================================
Actual sentiment is  : negative
======================================================================
predicted sentiment is negative with a probability of 0.9905090117827058
```

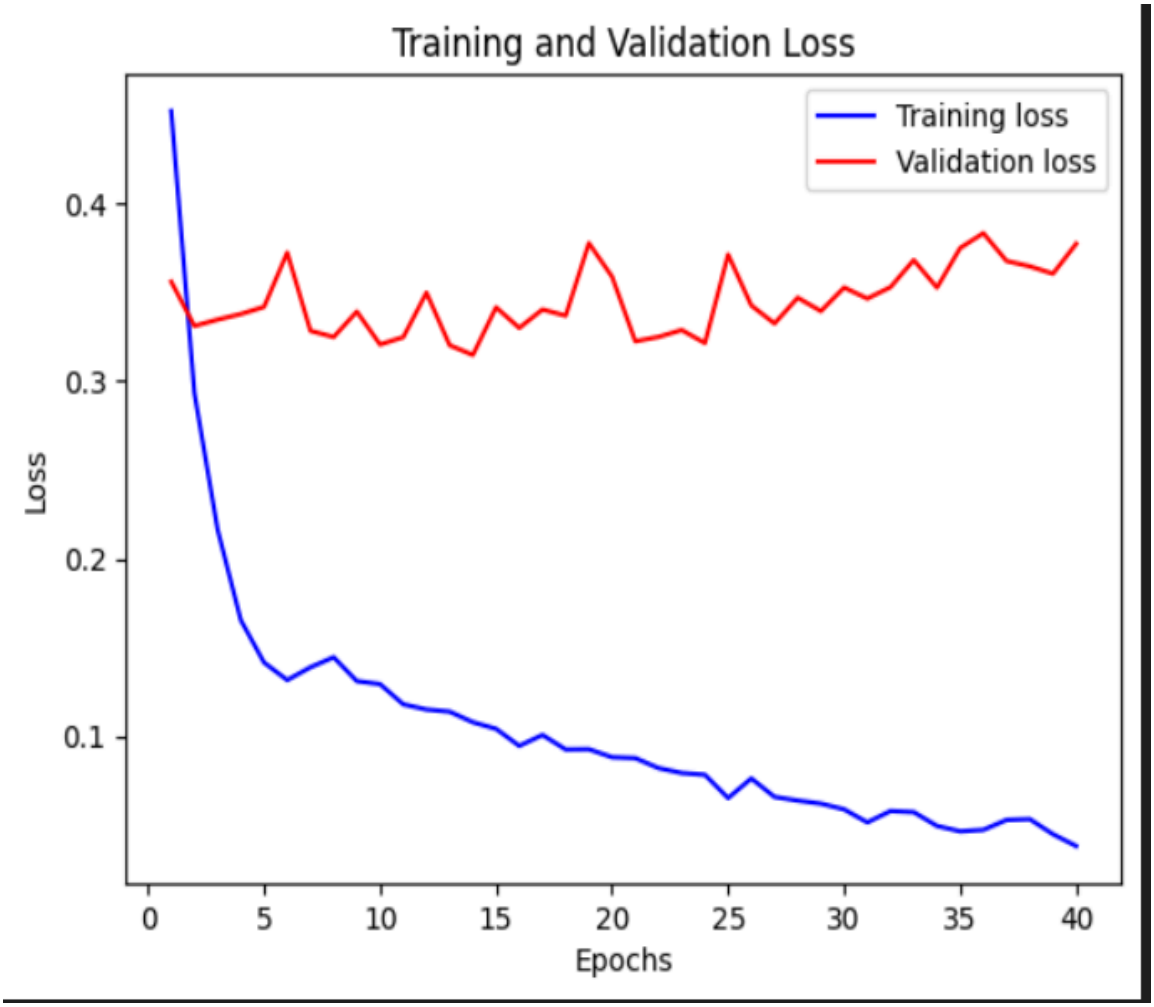
    -

3. Custom LSTM +CNN:

- Accuracy : 88%
- Prediction:

```
Sentence: This movie was fantastic! The acting was superb and the plot kept me engaged throughout.
Predicted Sentiment: Positive
Confidence: 1.00
---
Sentence: I was really disappointed with this book. The characters were poorly developed and the story was predictable.
Predicted Sentiment: Negative
Confidence: 1.00
---
Sentence: The restaurant had great ambiance, but the food was mediocre at best.
Predicted Sentiment: Positive
Confidence: 0.96
---
Sentence: I absolutely loved this song! It's catchy and the lyrics are meaningful.
Predicted Sentiment: Positive
Confidence: 1.00
---
```

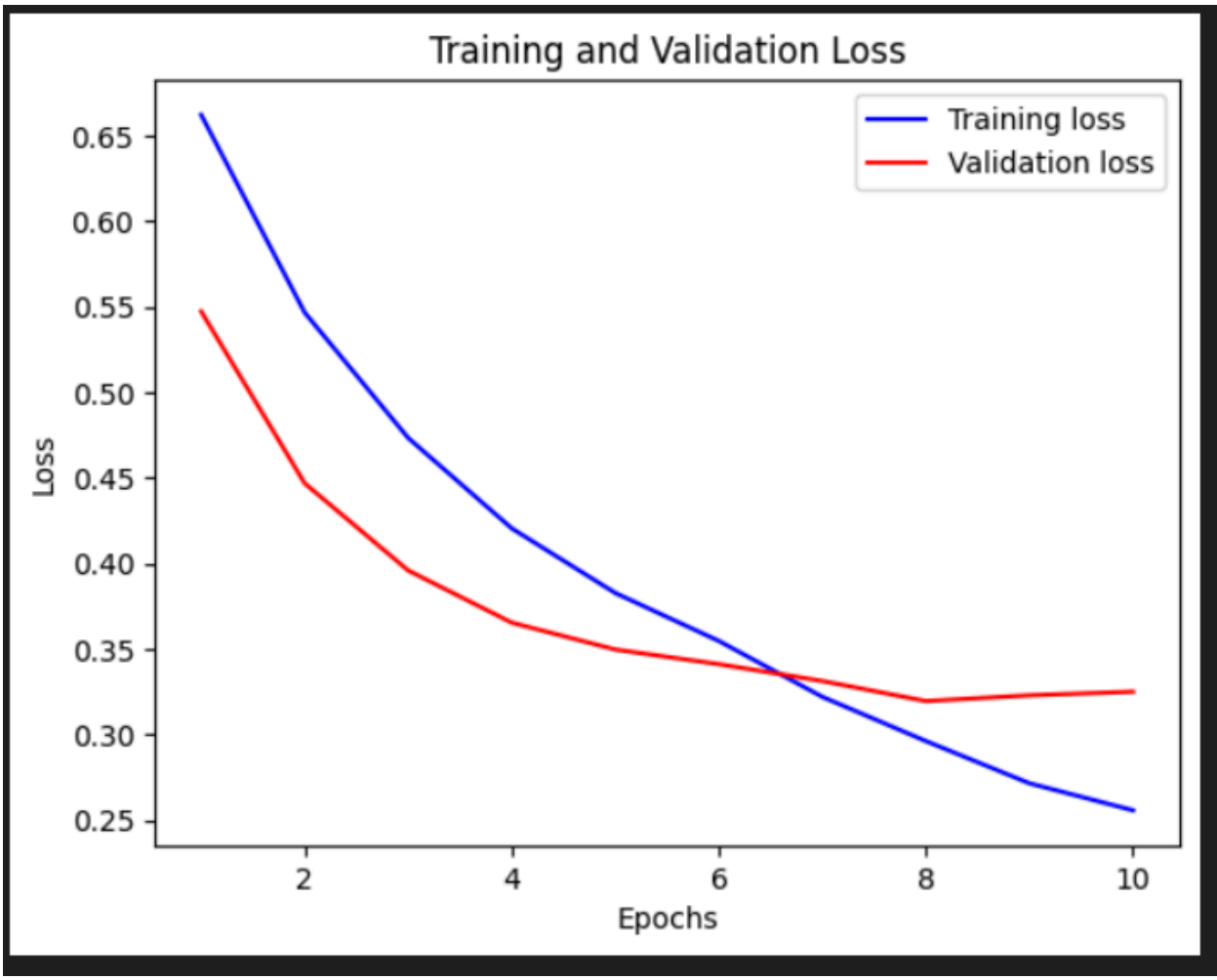
- Loss curves



4. Research Based CNN +LSTM:
   - Accuracy :88%
   - Prediction:

```
Sentence: This movie was fantastic! The acting was superb and the plot kept me engaged throughout.
Predicted Sentiment: Positive
Confidence: 0.97
---
Sentence: I was really disappointed with this book. The characters were poorly developed and the story was predictable.
Predicted Sentiment: Negative
Confidence: 1.00
---
Sentence: The restaurant had great ambiance, but the food was mediocre at best.
Predicted Sentiment: Positive
Confidence: 0.67
---
Sentence: I absolutely loved this song! It's catchy and the lyrics are meaningful.
Predicted Sentiment: Positive
Confidence: 1.00
---
```

- Loss curves

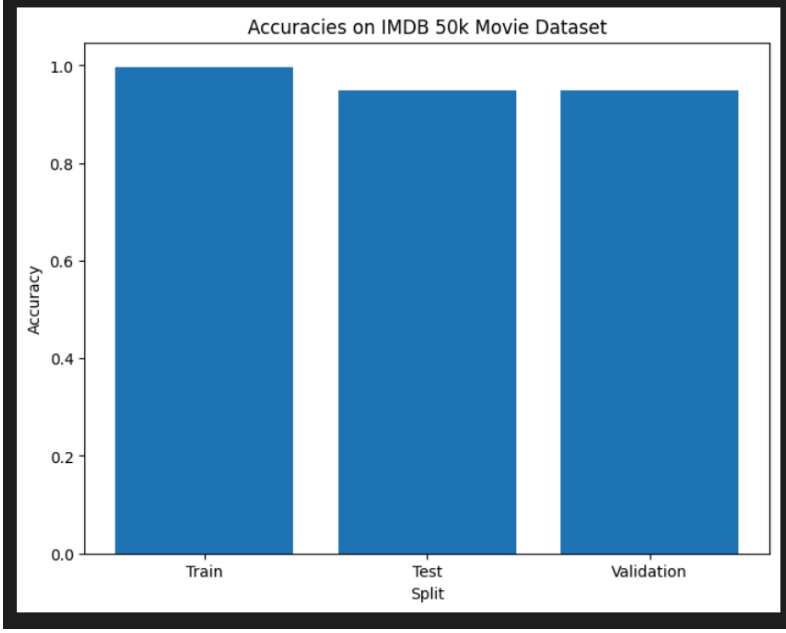


5. BERT:
    - Accuracy :95%
    - Prediction:

```
...  [2, 0, 1, 2]
     Statement: This movie was absolutely fantastic!
     Predicted Sentiment: positive

     Statement: I didn't enjoy the film at all. It was boring and predictable.
     Predicted Sentiment: negative

     Statement: The acting was great, but the story was lacking.
     Predicted Sentiment: neutral

     Statement: I absolutely loved this song! It's catchy and the lyrics are meaningful.
     Predicted Sentiment: positive
```

- Loss curves

```
Epoch 1/3 - Train Acc: 0.9861, Test Acc: 0.9488, Val Acc: 0.9497
Epoch 2/3 - Train Acc: 0.9940, Test Acc: 0.9504, Val Acc: 0.9493
Epoch 3/3 - Train Acc: 0.9972, Test Acc: 0.9494, Val Acc: 0.9493
```



6. Analysis of the solution with discussion on the possible weaknesses

---

1. LSTM for Sentiment Analysis

Pros:

- Sequence Modeling: LSTMs excel in handling sequences (like text), capturing long-term dependencies in data which is crucial for understanding the overall sentiment expressed in longer reviews.
- Context Awareness:LSTMs process data sequentially and therefore maintain context throughout the sequence, which is vital for accurately interpreting the meanings of words based on previous words in a sentence or passage.
- Flexibility with Input Length: Unlike fixed-size inputs in traditional neural networks, LSTMs can handle inputs of varying lengths, making them suitable for review texts that can significantly differ in size.

Cons:

- Computational Cost:LSTMs are typically slower to train than some other architectures due to their complex recurrent structure. They involve more computations per layer, which increases training time.
- Vanishing Gradient Problem: Although LSTMs are designed to mitigate the vanishing gradient problem better than standard RNNs, they can still suffer from this issue in very long sequences, potentially leading to less effective learning from earlier tokens in a text.
- Overfitting: Given enough parameters and training time, LSTMs can overfit on smaller or less varied datasets, learning to memorize training data rather than generalizing from it.

2. LSTM + CNN for Sentiment Analysis

Pros:

- Feature Extraction:CNNs are excellent at extracting hierarchical features from data. In the context of text, CNNs can efficiently detect key phrases and combinations of words that can be crucial indicators of sentiment, complementing the LSTM's ability to understand sequence and context.
- Efficiency: Incorporating CNN layers can lead to faster processing in some cases since CNNs can parallelize their operations more effectively than LSTMs. This can also reduce the training time when used appropriately.
- Improved Performance: Combining LSTM and CNN can lead to better performance as the model leverages both local features extracted by CNNs and global context provided by LSTMs, capturing more nuanced sentiment expressions.

Cons:

- Complexity: Using LSTM in combination with CNN increases model complexity, which can make the training process more challenging and resource-intensive. It also requires more careful tuning of parameters to avoid overfitting and to ensure that the model learns effectively from both architectures.
- Integration Challenges: Balancing the contributions from both LSTM and CNN components requires careful architectural planning and hyperparameter tuning. Improper integration can lead to one part dominating the other, which can negate the benefits of a hybrid approach.

- Resource Requirements: As both CNNs and LSTMs are individually demanding in terms of computational resources, their combination demands even more in terms of memory and processing power, possibly limiting their use in resource-constrained environments.

3. BERT (Transfer Learning)

Pros :

- Resource Intensive: BERT models are computationally expensive to train and fine-tune, requiring significant GPU resources and memory, which can be a barrier for smaller organizations or individual researchers.

- Fine-Tuning Complexity: While BERT can achieve state-of-the-art performance, it requires careful fine-tuning and hyperparameter adjustments. This process can be time-consuming and requires deep expertise in machine learning, particularly in handling transformer models.

- Overkill for Simple Tasks: For some simpler sentiment analysis tasks, BERT can be overkill, leading to unnecessary complexity and computational expense. Simpler models might achieve comparable results more efficiently.

Cons:

- Pre-trained Knowledge: BERT has been pre-trained on a large corpus of text and has learned rich representations of language. This pre-existing knowledge helps the model quickly adapt to the specific task of sentiment analysis with relatively little data.

- 2. Contextual Embeddings: BERT generates embeddings that are context-dependent, allowing for a nuanced understanding of word meanings based on their usage in sentences, which is crucial for accurately determining sentiments.

- 3. High Performance: BERT has demonstrated state-of-the-art performance on a variety of NLP tasks, including sentiment analysis. This means you can expect high accuracy and robustness from models fine-tuned from BERT.

- 4. Less Feature Engineering: BERT allows you to minimize the need for manual feature engineering (like stemming, n-grams, etc.), as it can inherently handle complex features from raw text.

- 5. Flexibility: BERT can be fine-tuned with additional layers as needed, making it adaptable to a wide range of sentiment analysis tasks, whether binary classification (positive/negative) or more granular sentiment scales.

7. Conclusion

This report has thoroughly explored the application of various deep learning models, including LSTM, CNN+LSTM, and transfer learning using BERT, on the IMDb 50k review dataset for sentiment analysis. The LSTM model showcased its strength in capturing long-range dependencies within text data, proving effective in understanding the context and flow of sentiments in movie reviews. The CNN+LSTM hybrid approach further enhanced performance by integrating CNN's ability to extract salient text features with LSTM's sequential data processing, offering a robust model that excels in handling complex linguistic structures.

The implementation of BERT through transfer learning emerged as a particularly powerful strategy, leveraging pre-trained contextual embeddings to significantly improve the accuracy and efficiency of sentiment classification. BERT's capacity to understand nuanced language details with minimal need for additional feature engineering places it at the forefront of natural language processing tasks.

Each model demonstrated its unique strengths and potential drawbacks, underscoring the importance of choosing the right architecture based on specific project requirements and available resources. The comparative analysis also highlighted the rapid advancements in deep learning technologies and their transformative impact on natural language processing. The findings from this investigation not only contribute valuable insights for academic research but also offer practical guidance for industry practitioners aiming to enhance sentiment analysis systems. Continuous refinement of these models and integration of emerging techniques will undoubtedly push the boundaries of what is possible in sentiment analysis and beyond.

8. References

Research paper for LSTM model :
https://www.researchgate.net/publication/341873850_Text_based_Sentiment_Analysis_using_LSTM

Other Research Papers:
For Understanding custom models:

1. ***https://aclanthology.org/C16-1047.pdf***
2. ***https://paperswithcode.com/paper/llambert-large-scale-low-cost-data-annotation***

3. ***https://www.youtube.com/watch?v=6A2w-KYG4Ko&t=1616s***

—---------- Thank You—----------