Report of MINI PROJECT: Predicting Movie Rating by LSTM

Submitted by-Divyanshu Rajoria
Hardik Arora
Hemant Prajapati
Shiven Rastogi

Submitted to-Sayantan Sinha

Week 1: Introduction and Data Preparation

Introduction

The project aims to develop a model the predicts Movie Ratings using Long Short-Term Memory (LSTM) for analyzing IMDB movie reviews. For this we used Sentiment analysis that plays a crucial role in understanding audience perceptions and sentiments towards movies, aiding decision-making in the film industry.

Problem Statement and Dataset

The problem statement is to review Movie Ratings using LSTM. The dataset used for this analysis is the IMDB dataset, which contains a collection of 50,000 movie reviews labeled as positive or negative sentiment. Each review includes the review text and its corresponding sentiment label.

The dataset is divided into a training set and a testing set, with 40,000 reviews for training and 10,000 reviews for testing. Each review is preprocessed to remove HTML tags, non-alphabetic characters, and stop words. Additionally, all text is converted to lowercase to ensure consistency in text representation.

Text Preprocessing

The text preprocessing steps applied to the IMDB dataset include:

1. **HTML Tag Removal:** HTML tags are removed from the review text to extract meaningful content.
2. **Non-Alphabetic Character Removal:** Non-alphabetic characters, such as punctuation marks and special symbols, are removed to focus on words' semantic meaning.

3. **Stop Word Removal:** Common stop words (e.g., 'the', 'and', 'is') are removed from the review text as they do not contribute significantly to sentiment analysis.
4. **Lowercasing:** All text is converted to lowercase to standardize text representation and avoid duplication of words based on case sensitivity.

Week 2: Model Building and Training

Text Encoding and Padding

Text encoding involves converting text data into numerical form that can be understood by machine learning models. In this project, the Tokenizer from Keras is used for text encoding. It assigns a unique integer to each word in the vocabulary and replaces words with their corresponding integers in the text data.

Padding sequences involves ensuring that all input sequences to the LSTM model have the same length. This is important because neural networks require fixed-length inputs. Padding adds zeros or truncates sequences to achieve a consistent length. In the provided code, sequences are padded to a fixed length of 200 characters.

Importance of Padding and Truncating Sequences for LSTM Input

Padding and truncating sequences are crucial for LSTM input due to the architecture of LSTM networks. LSTMs are designed to process sequences of fixed length, and varying sequence lengths can lead to inefficiencies or errors in the model. Padding ensures that all sequences have the same length, while truncating removes excess information beyond the specified length.

Model Architecture

The LSTM model architecture consists of three main layers: embedding layer, LSTM layer, and output layer.

Embedding Layer:

Converts integer-encoded words into dense vectors of fixed size.

Each word is represented by a dense vector in a high-dimensional space.

The embedding layer learns meaningful representations of words based on their context in the training data.

LSTM Layer:

Processes sequential data, capturing long-term dependencies and patterns.

LSTMs have memory cells that can maintain information over time steps, making them suitable for analyzing sequential data like text.

Output Layer:

Produces binary sentiment predictions (positive or negative) using a sigmoid activation function.

The output layer's activation function converts the model's final output into a probability score, indicating the likelihood of a positive sentiment.

Rationale Behind Choosing Specific Hyperparameters

A higher embedding dimension can capture more nuanced relationships between words but requires more computational resources.

LSTM Units (units=128):

The number of LSTM units defines the complexity and capacity of the LSTM layer.

More LSTM units can capture more complex patterns in the data but may also increase training time and resource requirements.

Training and Evaluation

During training, the model learns to minimize a binary cross-entropy loss function using the Adam optimizer. The optimizer adjusts the model's weights based on backpropagation, optimizing the model for better sentiment prediction accuracy. Evaluation on the test set provides insights into the model's generalization performance, including accuracy and loss metrics.

Week 3: Model Optimization and Analysis

Model Optimization

Model tuning techniques such as learning rate scheduling and dropout regularization were applied to enhance model performance. Learning rate scheduling adjusted the learning rate during training, while dropout regularization reduced overfitting by randomly disabling neurons during training.

These optimizations significantly improved model generalization and reduced overfitting, leading to better sentiment analysis accuracy.

Performance Analysis

The model's performance was evaluated based on accuracy, precision, recall, and F1 score metrics. The analysis revealed:

Accuracy: The model achieved high accuracy in sentiment prediction, indicating its effectiveness in distinguishing positive and negative reviews.

Precision and Recall: Precision measures the proportion of true positive predictions among all positive predictions, while recall measures the proportion of true positive predictions among all actual positive instances. The model

demonstrated balanced precision and recall, reflecting its ability to correctly identify positive and negative sentiments.

F1 Score: The F1 score, which combines precision and recall, provided a comprehensive assessment of the model's performance. A high F1 score indicates a good balance between precision and recall, highlighting the model's robustness in sentiment analysis.

Week 4: Results, Conclusion, and Future Work

Results and Discussion

The final results of sentiment analysis reveal the model's performance in accurately classifying reviews into positive and negative sentiments. Examples of correctly classified reviews showcase the model's effectiveness. However, instances of incorrectly classified reviews indicate areas for improvement.

Strengths of the Model:

High accuracy in sentiment prediction.
Generalization to unseen data demonstrated by the test set performance.

Weaknesses and Areas for Improvement:

Challenges in handling nuanced language.
Over-reliance on certain keywords or phrases for sentiment inference.

Conclusion

The Sentiment Analysis model successfully achieves its objective of accurately predicting sentiment in IMDB movie reviews. The project's objectives were achieved, showcasing the feasibility of using deep learning techniques for sentiment analysis tasks.

Future Work

Potential future work and enhancements for the model include exploring advanced Natural Language Processing (NLP) techniques and expanding the dataset.

References:

- Towards Data Science - NLP Section
- Analytics Vidhya - Deep Learning and NLP Resources
- : http://iieta.org/Journals/is