The Report of

Text Classification System (Movie Review Classification)

Created By:

1- Sarah Osama Mohamed Elfishawy.
2- Mahmoud Attya Mohamed Attya.
3- Yasmin Hamdy Mahmoud.
4- Lilian Stephanos Yonan.
5- Osama Hafez Fathy Hafez.

Introduction

- The goal of this project is to develop an effective sentiment analysis model capable of classifying movie reviews from the IMDb dataset as either positive or negative. Sentiment analysis is a fundamental task in Natural Language Processing (NLP) that helps to discern the emotional tone behind a body of text. This capability is essential for various applications, including market analysis, brand monitoring, and understanding consumer sentiments, which can significantly influence business strategies and decision-making.

The project addresses the following key questions:

- How accurately can the model predict sentiment from textual data?
- What are the challenges in processing natural language data, and how can they be mitigated?

1- Dataset Description

- The dataset used in this project is the IMDb Reviews dataset, which contains a set of 50,000 movie reviews labeled as positive or negative. This dataset is widely used for sentiment analysis tasks and is available through TensorFlow Datasets. The data is split into two subsets:

- Training set: 25,000 reviews
- Test set: 25,000 reviews

Data Characteristics:

- Textual Content: Each review contains free-form text expressing opinions about various movies.
- Labels: Each review is labeled with a binary sentiment value:
    - 1: Positive sentiment
    - 0: Negative sentiment

- The dataset is structured in a way that each review is paired with a binary label (1 for positive and 0 for negative).

2- Model Architecture

- The proposed model employs a sequential architecture designed using TensorFlow and Keras, incorporating the following layers:

Embedding Layer:

- Purpose: Converts integer-encoded words into dense vector representations, allowing the model to capture semantic relationships.

Parameters:

- Input_dim: 10,000 (vocabulary size)
- Output_dim: 64 (dimensionality of the embedding space)

LSTM Layer:

- Purpose: Captures long-term dependencies in the sequential data, making it effective for text data.

Parameters:

- Units: 64
- Return_sequences: False (to output only the last hidden state)

Dropout Layers:

- Purpose: Reduces overfitting by randomly setting a fraction of the input units to 0 during training.
- Rate: 20%

Dense Layer:

- Purpose: Adds a fully connected layer to learn complex representations.

Parameters:

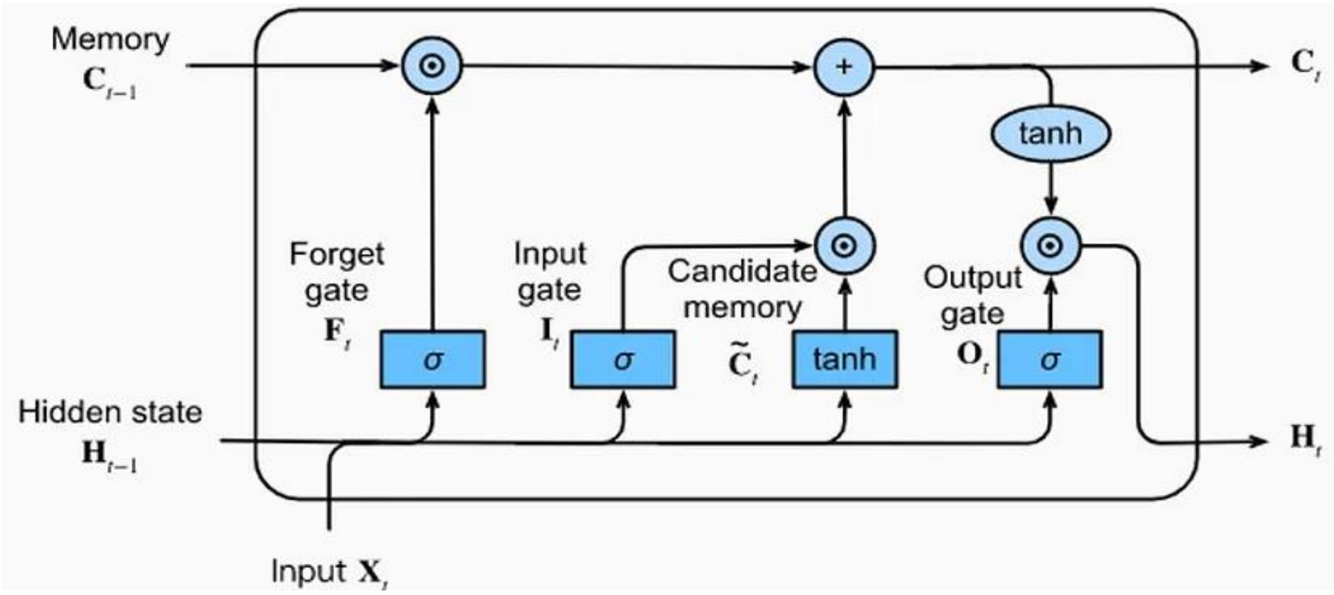
- Units: 64
- Activation: ReLU (Rectified Linear Unit)

Output Layer:

- Purpose: Provides the final classification output.

Parameters:

- Units: 1 (for binary classification)
- Activation: Sigmoid (to output a probability score)



(LSTM Model)

Model Summary

- The model architecture is defined as follows:

- Input shape: (None, 120) - where 120 is the maximum sequence length.
- The final output is a single probability indicating the sentiment class.

3- Training

- The model is trained using the following parameters:

- **Loss Function**: Binary Crossentropy, suitable for binary classification tasks.
- **Optimizer**: Adam optimizer, which adapts the learning rate during training.
- **Metrics**: Accuracy to evaluate the model's performance.

Training Procedure:

- **Epochs**: The model is trained for 5 epochs.
- **Batch Size**: Not explicitly mentioned; can be set as per requirements.
- **Validation Split**: The model is validated on a portion of the training data to monitor overfitting.

- During training, the model is fed with padded sequences of text data, enabling it to learn from the provided examples.

4- Evaluation

- Upon completion of training, the model is evaluated using the test dataset. The evaluation metrics include:

- Test Loss: Indicates how well the model performed on unseen data.
- Test Accuracy: Measures the proportion of correctly classified reviews. A high accuracy indicates a well-performing model.

Train Accuracy	Test Accuracy
99.9 %	92 %

5- Deployment

- To provide user-friendly access to the sentiment classification model, the application is deployed using Gradio. This framework allows the creation of interactive web interfaces for machine learning models.

Deployment Process:

- **Function Definition**: A function (predict_sentiment) is defined to preprocess input text and make predictions.
- **Gradio Interface Creation:**
  - Inputs: Text input field for users to enter movie reviews.
  - Outputs: Text output indicating predicted sentiment (Positive or Negative).
  - Title and Description: Provides context to users about the application's functionality.
- **Launching the Interface**: The Gradio interface is launched, enabling users to interact with the model in real time.

Examples

6- Conclusion

- The text classification project successfully demonstrates the implementation of a sentiment analysis model using the IMDb dataset. By following a structured approach—from data preprocessing and model training to evaluation and deployment—the project highlights the capabilities of deep learning in understanding human sentiments through text.

Future Work:

- Model Improvement: Exploring more advanced architectures, such as transformers (e.g., BERT or GPT), could enhance performance.

- Data Augmentation: Incorporating additional datasets or expanding the current dataset to include more diverse reviews may improve model robustness.

- Hyperparameter Tuning: Experimenting with different hyperparameters (e.g., learning rates, batch sizes) could optimize model performance further.

- This project lays a foundation for practical applications of sentiment analysis and provides insights into the development and deployment of machine learning models in real-world scenarios.