Sentiment Classification Using LSTM on Disaster Tweets Dataset

Introduction

This project aims to build a sentiment classification model using Long Short-Term Memory (LSTM) networks to identify tweets related to disasters. The model processes and classifies tweets into disaster or non-disaster categories. This project demonstrates the application of NLP techniques and deep learning for text classification tasks.

Data Description

- **Dataset**: The dataset consists of two columns:
  - `text`: The tweet content.
  - `target`: The label indicating whether the tweet is about a disaster (1) or not (0).
- **Sample Data**:
  - `text`: "Just another day in paradise."
  - `target`: 0 (No disaster)
  - `text`: "Huge earthquake hit the city."
  - `target`: 1 (Disaster)

Steps and Methodology

1. **Importing Libraries**:
   - Imported libraries including TensorFlow, Keras, pandas, numpy, and others required for data manipulation and model building.
2. **Loading the Dataset**:
   - Loaded the dataset from a CSV file using pandas.
   - Displayed the first few rows and checked the shape of the dataset.
3. **Data Exploration**:
   - Checked the distribution of disaster vs. non-disaster tweets to understand class balance.
4. **Preprocessing**:
   - **URL Removal**: Implemented a function to remove URLs from the text.
   - **Punctuation Removal**: Removed punctuation from the text using `str.translate`.
   - **Stopwords Removal**: Removed common stopwords to focus on meaningful words using NLTK.
   - **Word Count**: Counted the frequency of unique words to understand vocabulary size.
5. **Data Preparation**:
   - **Tokenization**: Tokenized the text data using Keras' `Tokenizer`, converting text into sequences of integers.
   - **Padding**: Applied padding to ensure all sequences have the same length using Keras' `pad_sequences`.

    - **Reverse Mapping**: Created a reverse mapping for decoding sequences back into text for validation.
6. **Model Building**:
    - **Model Architecture**:
        - **Embedding Layer**: Converts word indices into dense vectors.
        - **LSTM Layer**: Captures long-term dependencies in sequences.
        - **Dense Layer**: Outputs the classification (disaster or not).
    - Compiled the model using binary cross-entropy loss and Adam optimizer.
7. **Training and Evaluation**:
    - Trained the model for 20 epochs with validation data to monitor performance.
    - Predicted the sentiments on training data and compared with true labels.

Results

1. **Model Performance**:
    - **Training Accuracy**: The model achieved good accuracy on the training data, demonstrating its effectiveness in learning from the given examples.
    - **Predictions**: Sample predictions for tweets were compared with actual labels to validate the model's performance.
2. **Example Predictions**:
    - For tweets from index 10 to 20:
        - **True Labels**: `[1, 0, 1, ...]`
        - **Predicted Labels**: `[1, 0, 1, ...]`
        - The model correctly predicted the sentiment for most of the tweets.

Conclusion

- The project successfully built an LSTM-based sentiment classification model to differentiate between disaster and non-disaster tweets. The model demonstrated effective performance in classifying tweets based on their content.
- The preprocessing steps, including URL removal, punctuation removal, and stopwords removal, were crucial in preparing the text data for modeling.

Future Work

- **Model Improvement**: Explore more advanced architectures such as Bidirectional LSTM or Transformer models for potentially better performance.
- **Hyperparameter Tuning**: Perform hyperparameter tuning to optimize the model's parameters.
- **Data Augmentation**: Use data augmentation techniques to enhance the dataset, especially if the class imbalance is significant.
- **Real-time Classification**: Deploy the model to classify tweets in real-time using a web application or API.
- **Evaluation Metrics**: Evaluate the model using additional metrics such as precision, recall, and F1-score to gain deeper insights into its performance.

Prepared Responses for Interview

1. **Introduction**: "This project focuses on classifying tweets as related to disasters or not using LSTM networks. We processed and prepared the data, built an LSTM model, and achieved effective classification results."
2. **Data Description**: "The dataset consists of tweets and their labels indicating disaster or non-disaster. We performed preprocessing tasks like removing URLs, punctuation, and stopwords to prepare the data for modeling."
3. **Steps and Methodology**: "We tokenized and padded the text data, built an LSTM-based model, and trained it for 20 epochs. We then evaluated the model's performance on the training data."
4. **Results**: "The model showed good accuracy and effectively predicted the sentiment of tweets. Example predictions were compared with true labels to verify accuracy."
5. **Conclusion**: "The LSTM model successfully classified tweets into disaster and non-disaster categories, demonstrating its capability in sentiment analysis tasks."
6. **Future Work**: "Future improvements could include exploring advanced models, hyperparameter tuning, data augmentation, real-time classification, and evaluating additional metrics.