Next Word Prediction Using LSTM Model

Project Overview

This project implements a next-word prediction model using a **Long Short-Term Memory (LSTM)** network, commonly used in natural language processing tasks for sequence prediction. The model predicts the next word in a given sentence based on learned patterns from a custom dataset. It can serve applications like autocomplete and typing assistance.

Data

The dataset for this project was derived from a text file containing a corpus of sentences. Here's an outline of the data preprocessing steps:

- Tokenization: Text data was tokenized into individual words and encoded as integer sequences.

- Sequence Generation: Each sentence was split into sequential input-output pairs, where each pair consists of a sequence of words as input and the next word in the sequence as output.

- Padding and Encoding: Sequences were padded to ensure uniform input length, and the output labels were one-hot encoded.

The dataset's vocabulary contained **total_words** unique words, with sequences padded to a maximum length of **max_sequence_len.**

Model Architecture

The model consists of the following layers:

1. Embedding Layer: Maps each word index to a dense vector representation of 100 dimensions.

2. LSTM Layer: A 150-unit LSTM layer captures dependencies between words in a sequence, learning the contextual structure of sentences.

3. Dense Output Layer: A **softmax** layer outputs probabilities across all vocabulary words, predicting the most likely next word.

Training and Evaluation

- Loss Function: Categorical cross-entropy loss was used, suitable for multi-class classification.

- Optimization: Adam optimizer was chosen for efficient gradient-based optimization.

- Metrics: Accuracy, precision, and recall metrics were tracked during training to monitor model performance.

The model was trained for **15 epochs**, after which it achieved satisfactory accuracy on the training data, indicating effective learning of sequential word patterns.

Evaluation Results

A confusion matrix was generated on a sample of predictions, focused on the **top 30 classes** to avoid memory overload. This allowed an evaluation of the model's performance across frequently used words, providing insight into misclassifications and prediction accuracy for each word class.

Prediction Example

The model was tested by inputting a starter phrase **(e.g., "I am")** and generating the next three words based on the highest-probability predictions. This showcased the model's ability to form coherent sentence continuations, indicating it had successfully captured word dependencies.

Essential Information

- Data Size: The model processes a significant vocabulary with padded sequences, making it adaptable for various sentence structures.

- Output Limitation: Due to computational constraints, the model limits vocabulary in the confusion matrix to the most frequent words, focusing on core language patterns.

- Use Case: This model is suited for autocomplete functionalities, offering quick, contextually relevant suggestions for continued sentence creation.