Aditya Luthra **2K22/CO/30**

Github Link : https://github.com/Adiiii02/DL-LAB/blob/main/EXP-4/rnn-lstm.ipynb

Text Generation using RNNs and LSTMs

Objective

The aim of this experiment is to explore text generation using Recurrent Neural Networks (RNNs) and LSTMs and understand the impact of different word representations:

1. One-Hot Encoding
2. Trainable Word Embeddings

Train both RNN and LSTM models on a dataset of 100 poems and compare the performance of these encoding techniques.

Dataset

The dataset consists of 100 poems, each containing multiple lines of poetry. This dataset is used to train the text generation models.

Approaches

1. One-Hot Encoding

- Tokenize the text into words.
- Convert each word into a one-hot vector.
- Train both RNN and LSTM models using one-hot encoded word sequences.

2. Trainable Word Embeddings

- Tokenize the text into words.
- Convert each word into an index.
- Use an embedding layer in both RNN and LSTM models.
- Train the embedding layer along with the model.

Model Variants

The following four models were trained for 70 epochs and compared :

Model Type	Encoding Type	Training Loss	Training Time
RNN	One-Hot Encoding	0.0203	1660s
LSTM	One-Hot Encoding	0.0013	7850s
RNN	Trainable Embeddings	0.0245	771s
LSTM	Trainable Embeddings	0.0009	916s

Comparison and Analysis

- **Training Time:** Models using trainable word embeddings took lesser time to train .
- **Loss Convergence:** LSTM models demonstrated better loss convergence compared to RNNs, especially with longer sequences.
- **Generated Text Quality:**
    - One-hot encoding models tended to produce repetitive sequences due to limited representation.
    - Trainable embeddings led to more fluent and diverse text generation.
- **Memory Efficiency:** Using trainable embeddings significantly reduced memory usage compared to one-hot encoding.

Results

- The LSTM models outperformed RNNs in generating coherent poetry.
- Trainable embeddings improved word associations and allowed for better generalization.
- One-hot encoded models, while simpler, struggled with vocabulary size limitations.

This study highlights the importance of word representation in text generation and demonstrates the advantages of trainable embeddings over one-hot encoding.