Image Captioning using Attention Mechanism

Week 3: Introduction to Sequential Models (RNN, LSTM)

Welcome to the **third week** of our exciting journey into the world of sequential models! This week, we will delve deep into some of the most fascinating and powerful tools in machine learning: Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks.

What We Will Cover This Week:

1. **Introduction to Sequential Models**
    - Understanding the fundamental concepts behind sequential data and why specialized models are necessary for processing it.
2. **Recurrent Neural Networks (RNNs)**
    - Exploring the structure and function of RNNs.
    - Learning how RNNs maintain a memory of previous inputs to handle sequences.
    - Discussing the common issues faced with RNNs, such as vanishing and exploding gradients.
3. **Long Short-Term Memory (LSTM) Networks**
    - Diving into LSTM architecture and understanding how they overcome the limitations of traditional RNNs.
    - Examining the components of LSTMs, including cell state, hidden state, and gates (input, forget, and output gates).
4. **Training Sequence Models for Time-Series Data**
    - Applying RNNs and LSTMs to time-series data to make predictions and analyze trends.
    - Implementing practical examples to solidify our understanding.
    - Experimenting with different datasets and tuning models for better performance.

Resources

To help you understand and implement the concepts covered this week, here are some recommended resources:

- Recurrent Neural Networks (RNNs), Clearly Explained!!!
- What is Recurrent Neural Network (RNN)? Deep Learning Tutorial 33 (Tensorflow, Ker…
- Recurrent Neural Network (RNN) Tutorial | RNN LSTM Tutorial | Deep Learning Tutori…

Assignment-3: Sentiment Analysis using RNNs (Deadline: 19th June)

Problem Statement:

Develop a Recurrent Neural Network (RNN) model for sentiment analysis using TensorFlow and Keras. Utilize a dataset containing reviews (e.g., movie reviews, product reviews) and construct an RNN architecture suitable for text classification. You may choose to start with an LSTM or GRU layer. The objective is to build a model that can accurately classify the sentiment of reviews (positive, negative, neutral) and evaluate its performance using standard metrics.

Data Preparation:

- Download a dataset containing a variety of reviews with labeled sentiments (e.g., IMDb movie reviews, Amazon product reviews).
- You can use this Dataset also.
- Clean the text data.
- Tokenize the reviews and convert them to sequences.
- Pad sequences to ensure uniform input length for the RNN.

Model Development:

- Choose between LSTM or GRU layers for the RNN model.
- Ensure your model includes essential components such as embedding layers for text representation and dense layers for classification.

Training the Model:

- Split the dataset into training, validation, and test sets.
- Train your model using the training set.
- Apply techniques like data augmentation (e.g., synonym replacement) and regularization (e.g., dropout) to improve model performance.

Evaluation:

- Evaluate the model on a separate test set not used during training.
- Report metrics such as accuracy, precision, recall, F1 score, and any other relevant performance metrics.

Submission Instructions

Submit the code and analysis on your respective GitHub and Google Drive link as stated in the Submission Form.