SEASONS OF CODE 2024

Sentiment Analysis and Text Generation using Many-to-One LSTMs

Mentors: Shreyas Katdare & Dion Reji

Date: May 23, 2024

Instructions

1. This project is expected to be completed within 8 weeks. A detailed (tentative) timeline of the project, and checkpoints are attached in Appendix A and Appendix B.

2. Our primary mode of communication will be WhatsApp. (See Appendix C for detailed contact details). We shall release weekly resources and update official announcements on this GitHub repo. Click here.

3. **Pre-requisites.** We will be working with Python throughout the project. So a basic understanding of Python, and useful libraries like numpy, pandas, etc. is well appreciated. However, we will provide enough resources to get you started with required Python background. Also, Machine learning in itself is a mathematically heavy subject. So, basic understanding of Linear Algebra and Calculus is essential.

4. This project will be completed through a series of weekly assignments, which will be evaluated to monitor your progress. The final certificate will be awarded based on your completion of these assignments. Collectively, these assignments will guide you through the entire project.

5. You are supposed to maintain a GitHub repository of all the files you use for this project. The final evaluation will be done based on this repository.

6. **Submission.** We expect you to submit all the necessary code for running your project, with a suitable README on how to run the files, as a GitHub repo mentioned above. You are also required to prepare a (short) report on your implementation. You are free to experiment, and add on more features to the project, but be sure to mention its details in the report. (More on report and submissions later!)

7. **Honesty is the Best Policy!** We expect you to behave honestly and with integrity throughout the project. Though helping out each other is well appreciated, we expect you to submit the work you have done yourself. So, **DO NOT PLAGIARISE**. This includes copying others' code, as well as code available online, or any other public domain. Be sure to acknowledge all sources and references you have used during the project in your report. Though you are free to use online tools like ChatGPT, be sure you understand what you code.

Overview

This project explores the application of Long Short-Term Memory (LSTM) neural networks in two significant natural language processing (NLP) tasks: sentiment analysis and text generation. By leveraging the capabilities of LSTMs, we aim to build robust models for predicting sentiment labels from textual data and generating coherent text sequences.

We present here a broad overview of the project. More details on how to implement will be shared on the go.

Sentiment Analysis

Objective

The primary goal of this task is to develop a sentiment detection model that can predict sentiment labels (0 or 1) based on text reviews. These reviews can come from various domains such as movies, products, or any other context where sentiment analysis is applicable.

Data Description

Sentiment Dataset: This dataset includes text reviews and corresponding sentiment labels. Each review is labeled as either positive (1) or negative (0).

Approach

Preprocessing.

- **Text Cleaning**: Remove unnecessary characters, punctuation, and special symbols from the reviews.
- **Tokenization**: Split the text into individual words or tokens.
- **Stop Word Removal**: Eliminate common stop words that do not contribute to the sentiment.
- **Vectorization**: Convert the text reviews into numerical representations using techniques such as bag-of-words or word embeddings.

Many-to-One LSTM

- **Architecture**: Utilize a many-to-one LSTM architecture where the LSTM processes the entire sequence of words in a review and outputs a single sentiment label.
- **Input Representation**: Feed the numerical representation of the text reviews as input to the LSTM model.

Training and Evaluation

- **Dataset Split**: Split the dataset into training and testing sets to evaluate model performance.
- **Training**: Train the LSTM model using the training set.
- **Evaluation**: Assess the model's performance on the testing set using metrics like accuracy, precision, recall, and F1-score.

Text Generation

Objective

The secondary goal is to utilize many-to-one LSTMs for text generation. By training the model on a text corpus such as *Alice's Adventures in Wonderland*, the LSTM learns to predict the next word in a sequence, generating coherent and contextually relevant sentences.

Data Description

Text Dataset: The text of *Alice's Adventures in Wonderland* by Lewis Carroll, or any other suitable text corpus, serves as the training data for the text generation model.

Approach

Preprocessing

- **Text Cleaning**: Remove unnecessary characters and whitespace from the text.
- **Tokenization**: Split the text into sequences of words or characters.
- **Sequence Creation**: Create sequences of a fixed length from the text data, where each sequence is used to predict the next word.

Many-to-One LSTM

- **Architecture**: Implement a many-to-one LSTM architecture where the model processes a sequence of words and predicts the next word in the sequence.
- **Training**: Train the LSTM model using the prepared sequences.

Challenges in Text Generation

- **Language Variability**: Natural language is diverse, with multiple words having similar meanings. The model must handle this variability effectively.
- **Context Dependence**: Generating contextually relevant text requires the model to understand the context and maintain coherence across sequences.

Techniques for Improved Generation

- **Entropy Scaling**: Introduce controlled randomness into text generation to produce diverse outputs.
- **Softmax Temperature**: Use the softmax temperature hyperparameter to control the randomness in the model's predictions. Lower temperatures make the model more confident in its predictions, while higher temperatures introduce more variability.

Implementation

- **Training**: Train the LSTM model on the text dataset.
- **Generation**: Use the trained model to generate text by predicting the next word in a sequence based on a given input.

Tech Stack

- **Language**: Python
- **Libraries**: pandas, numpy, keras, tensorflow, collections, nltk

Appendix A: Timeline

Week 1: Set up development environments and install necessary libraries and packages.
Learning basic Python syntax and basic libraries like pandas, numpy.

Week 2: Review project requirements and dataset specifications.
Learning overall architecture, mathematics and approach for sentiment analysis and text generation using LSTM.

Week 3: Introduce the concept of LSTM neural networks and the many-to-one architecture.
Part 1 of the project begins
Implement a many-to-one LSTM model for sentiment analysis using Keras and TensorFlow.

Week 4: Split the dataset into training and testing sets.
Train the LSTM model on the training set and evaluate its performance on the testing set.
Analyze model metrics such as accuracy, precision, recall, and F1 score.
Discuss potential improvements and optimizations for the sentiment analysis model.

Week 5: Part 2 of the project begins
Perform data preprocessing tasks such as text cleaning, tokenization, and sequence creation.
Prepare the text data for training the LSTM model for text generation.

Week 6: Implement a many-to-one LSTM model for text generation using Keras and TensorFlow.
Train the LSTM model using the prepared text dataset.
Explore techniques such as entropy scaling and softmax temperature for enhancing text generation quality.

Week 7: Evaluate the trained LSTM model for text generation using qualitative assessment.
Generate sample text sequences and analyze the coherence and relevance of generated text.
Fine-tune the LSTM model parameters and hyperparameters based on feedback and observations.

Week 8: Present final results and outcomes of sentiment analysis and text generation tasks.

Appendix B: Checkpoints

Checkpoint 1: Learning basics of Python and necessary libraries

Checkpoint 2: Sentiment Analysis (Part 1) Model Development

Checkpoint 3: Sentiment Analysis Model Evaluation and Optimization

Checkpoint 4: Text Generation (Part 2) Model Development

Checkpoint 5: Text Generation Model Evaluation and Fine-Tuning

Appendix C: Contact us

Shreyas Katdare
Ph. 9819460807 (WhatsApp)
Email: 22b0636@iitb.ac.in

Dion Reji
Ph. 7907729867 (WhatsApp)
Email: 22b0029@iitb.ac.in