CSCI-739 Topics in Intelligent Systems Project

LSTM Implementation on C++

Shlok Gupta, Pratyush Jena, Karamcheti Pritham

Overview

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) particularly well-suited for learning from sequences of data, such as time series, speech, text, or video. They were developed to address the challenge of long-term dependencies, where the network needs to remember information for long periods, a task at which traditional RNNs often fail.

Key Features of LSTMs:

- Memory Cells: The core concept of an LSTM is the memory cell, which can maintain its state over time. These cells are connected through layers, much like a standard neural network.
- Gates:

  Forget Gate: Decides what information should be thrown away from the cell state.

  Input Gate: Updates the cell state with new information.

  Output Gate: Determines what the next hidden state should be, based on the current cell state and the input.

- Handling Long-Term Dependencies: By selectively adding or removing information to the cell state, LSTMs can learn which data in a sequence is important to keep and which can be discarded, effectively overcoming the problem of long-term dependency.

In this project we design an LSTM Neural Network on C++ to perform occupancy detection based on the environmental sensors reading dataset. This is time series data which makes LSTM one of the suited algorithms to use. The dataset contains the below fields.

1. Date and Time
2. Temperature
3. Humidity
4. Light
5. CO2
6. Humidity Ratio
7. Occupancy – Label (0 or 1)

Implementation Details

The code uses CUDA kernels for high performance implementation and below are the operations for which kernels are defined.

1. Sigmoid and Partial derivative
2. Tanh and Partial Derivative
3. Error Calculation
4. Inner Product
5. Cell State update
6. Backward Pass
7. Weight update

The code repository contains primarily 4 modules.

- File Processor
- Data Processor
- LSTM class
- Main function

Main function run does not expect any command line arguments, the below hyper parameters are defined and later used.

- Number of Cells
- Timesteps
- Learning rate
- Iterations (epochs)

Features Supported

- Implementation on GPU created CUDA kernels for forward, backward and other elementary operations.
- System Requirements: Linux, MacOS and CUDA toolkit.
- Task: Classification

Steps to run.

- Run compile.sh
- Run lstm_exec executable