Hangman_TrexQuant

Aditya Raj

October 2024

1 Introduction

The purpose of this project is to implement an AI-powered Hangman game using a deep learning model. The project is composed of two major components:

- The `HangmanPlayer` class, responsible for managing game logic, guessing strategy, and keeping track of the game state.
- An LSTM-based neural network model, used to predict the next letter guess based on the current game state.

2 Class Design

2.1 HangmanPlayer Class

The `HangmanPlayer` class encapsulates the logic for playing a Hangman game, with a neural network providing the guesses. The class keeps track of the word being guessed, previously guessed letters, and remaining lives.

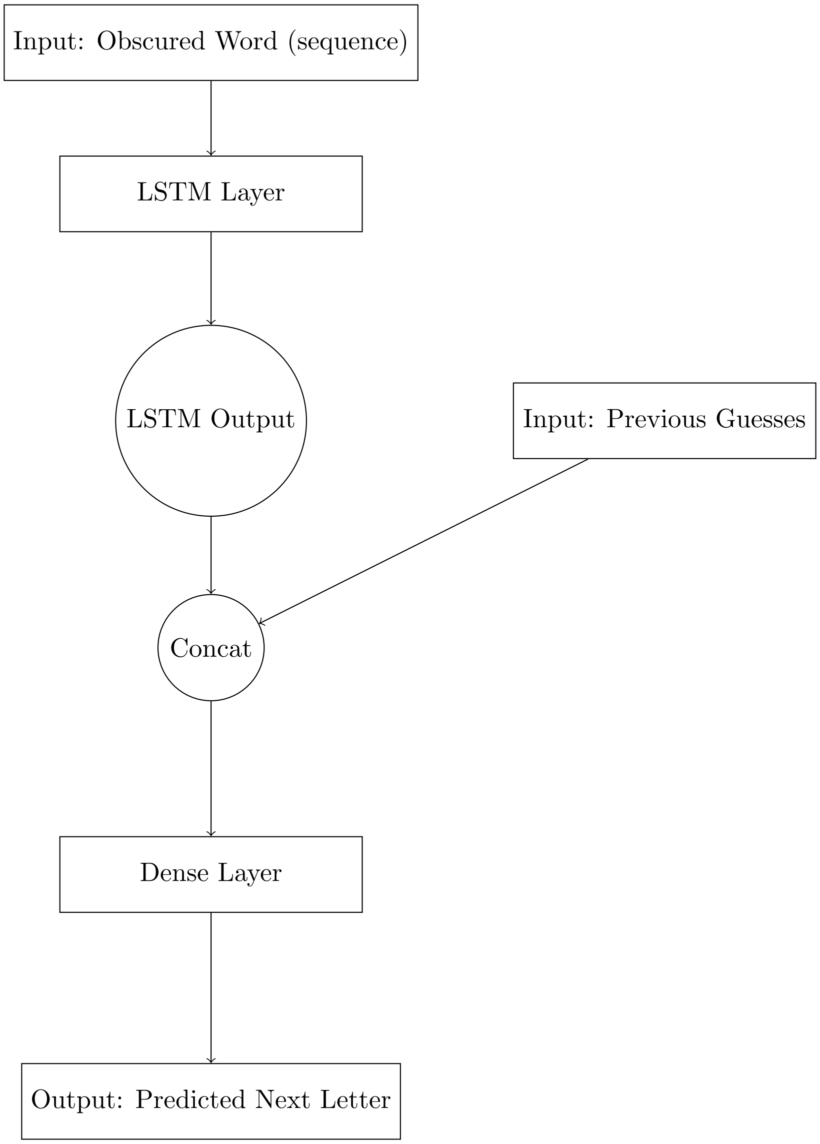
3 Neural Network Model

3.1 LSTMNet Class

The `LSTMNet` class is the neural network architecture used to predict the next letter to guess. It utilizes an LSTM (Long Short-Term Memory) network to process sequences of previously guessed letters and the obscured word.

3.1.1 Architecture

- **LSTM Layer**: The LSTM processes the sequence of obscured words seen, capturing temporal dependencies.
- **Dense Layer**: Combines the LSTM output with the encoded previous guesses to produce a probability distribution over the possible next letters.



3.2 Training Process

The model is trained using a dataset of words. For each word, the Hangman game is simulated, and the model is trained to predict the correct next letter based on the current game state.