Assignment Report

Deep learning 1402
Dr. Fatemizade

HW4

By Mohammad Amin Molaei Arpanahi
402011148
402012018

Question 1
Analysis of BBBP Dataset with MLP, LSTM, and BiLSTM Models

1. Introduction

In this extensive assignment, I delved into the Blood-Brain Barrier Penetration (BBBP) dataset with the objective of predicting the permeability of chemical compounds through the blood-brain barrier using various machine learning models. The primary focus was on implementing, training, and evaluating Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), and Bidirectional LSTM (BiLSTM) models.

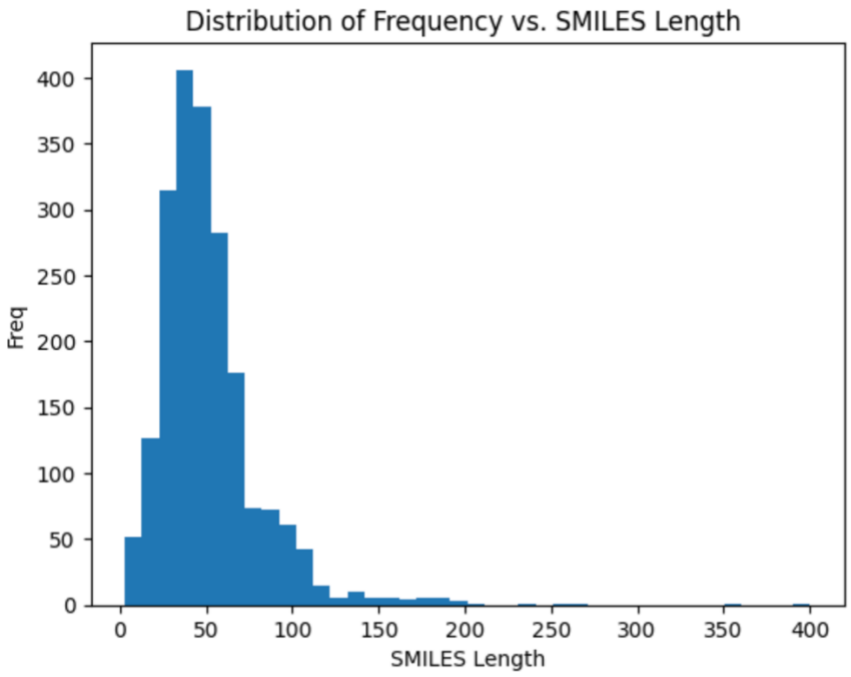
2. Data Exploration and Preprocessing

2.1 Dataset Overview

Upon loading the BBBP dataset, I conducted an initial exploration to understand its structure and gain insights into the features that would influence our modeling decisions. The dataset comprises molecular structures represented by Simplified Molecular Input Line Entry System (SMILES) strings, accompanied by binary labels indicating blood-brain barrier permeability.

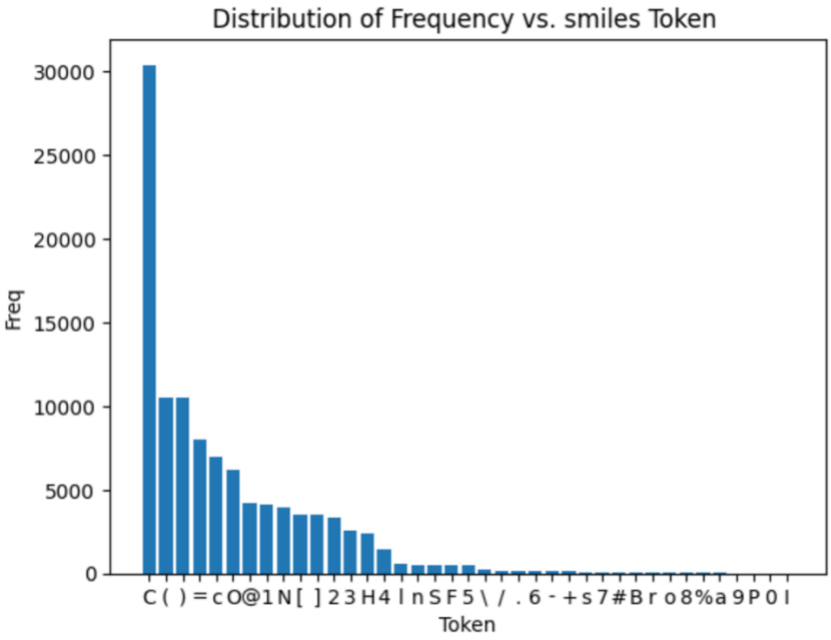
2.2 SMILES Length Distribution

To gain a better understanding of the sequence lengths in the dataset, I visualized the distribution of SMILES string lengths using a histogram. This analysis is crucial for determining an appropriate strategy for handling variable-length sequences during model training.



2.3 Unique Character Analysis

An investigation into the unique characters present in the SMILES strings was undertaken. This analysis is foundational for the subsequent one-hot encoding step, ensuring that all relevant characters are appropriately represented in the numerical encoding.



2.4 One-Hot Encoding

To prepare the SMILES strings for input into machine learning models, I employed one-hot encoding on the unique characters present. This conversion facilitated the transformation of SMILES strings into numerical vectors, preserving the sequence information.

2.5 Padding Sequences

To ensure uniformity in input size across all sequences, I used PyTorch's **pad_sequence** function to pad the sequences with zeros. This step is crucial for the effective training of neural network models that require fixed-size inputs.

3. Model Implementation

3.1 Multilayer Perceptron (MLP)

The first model implemented was a Multilayer Perceptron (MLP). The architecture comprised two fully connected layers, with the input size determined by the flattened one-hot encoded vectors.

3.2 LSTM with Fully Connected Layer

A Long Short-Term Memory (LSTM) model was implemented to capture sequential dependencies in the data. This model included an additional fully connected layer for enhanced learning capabilities.

3.3 Bidirectional LSTM (BiLSTM) with Fully Connected Layer

To further improve sequence modeling, a Bidirectional LSTM model was implemented. This model processes sequences in both forward and backward directions, capturing more intricate dependencies in the data.

4. Model Training and Evaluation

4.1 K-Fold Cross-Validation

The models were subjected to K-Fold cross-validation with 5 folds to assess their performance robustness. Each model underwent training and evaluation on different subsets of the dataset, allowing us to observe variations in performance.

4.2 Training Loop

The training loop involved optimizing the models using the Adam optimizer and binary cross-entropy loss. Training and validation accuracies were monitored for each epoch, providing insights into the convergence and generalization capabilities of the models.

5. Model Evaluation on Smiles Length Bins

5.1 Binning Smiles Lengths

To explore how the models perform on sequences of varying lengths, the test set was sorted based on the lengths of SMILES strings before padding. The sorted list was then divided into 10 bins, each representing a range of SMILES string lengths.

5.2 Model Evaluation on Bins

For each bin, the accuracy of each model (MLP, LSTM, BiLSTM) was evaluated using the corresponding subset of the test set. This granular analysis provided insights into how each model performs on different lengths of SMILES strings.

6. Conclusion

All the result are reported at the notebook.

The insights gained from this assignment can serve as a foundation for further optimizations, hyperparameter tuning, and exploration of advanced model architectures to improve predictive performance on the challenging task of blood-brain barrier permeability prediction.