A-Domain Prediction

-Adrita Das

Architectures

- **Model Architecture**

1. Input Layer: The input layer receives input sequences of amino acid types encoded as one-hot vectors. The input size is determined by the number of amino acid types, including the missing token and padding token.

2. LSTM Layer: The LSTM layer is responsible for capturing sequential dependencies in the input sequences. It takes the one-hot encoded input sequences and processes them through LSTM units. The hidden size determines the number of hidden units in the LSTM layer. The number of layers determines the depth of the LSTM architecture, and the dropout parameter controls the dropout probability between LSTM layers.

1. Fully Connected Layer: The output of the LSTM layer is passed through a fully connected layer. The hidden size of the LSTM layer is used as the input size for the fully connected layer, and the output size is determined by the number of output classes, obtained from the dataset label map.

Hyperparameters

- The model's hyperparameters used in the example code are as follows:
- Input_size : The number of amino acid types in the input sequences, including missing token and padding token
- Hidden_size :The number of units in LSTM layer. It was set to 256
- Num_layers : The number of LSTM layers in the model. It was set to 2.
- Dropout : The dropout probability of the LSTM layer. It was set to 0.2 here, i.e 20% of the LSTM units will be dropped randomly during training.
- Output_size : The number of output classes, obtained from the dataset label_map.

Criterion and Optimizer

- The criterion(loss function) and optimizer used for training the model are as follows:
- Criterion: Cross Entropy Loss.
- Optimizer : Adam
- Learning Rate : 0.01

Performance of the Different Models

- The performance of this model was at 28.7% accuracy. The performance of the previous LSTM model in which no dropout or regularization was applied, was significantly low.
- Adding regularization and increasing the number of layers, definitely boosted the model accuracy.
- Also, very complex models didn’t quite seem to be working with this dataset, as the data was very small.
- The change in the hyperparameters did not quite change the model performance. Increasing the hidden_size of the LSTM units did boost model performance.