ISM report

Samvit Jatia

May 2024

GitHub Repository

Motivation

One of the major issues in biological research is the prediction of protein sequence and, by extension, a factor critical in the understanding and counteraction of a disease like influenza. Accurate prediction of protein sequences is therefore helpful in explaining the structure and function of viral proteins, which is foundational in developing working vaccines and antiviral drugs. In case it is possible to predict the protein accurately, one would understand the viral mechanism and develop targeted therapy and preventive measures.

Background

The Transformer architectures, in particular BERT (Bidirectional Encoder Representations from Transformers), are at the forefront of natural language processing, allowing for the in-depth extraction of long dependencies and contextual relationships within text. Thus, the application of BERT-like models in data from DNA sequences, under the DNABERT implementation, shows that a pre-trained model can fare well in biological sequence analysis.

Pre-trained models such as DNABERT have various benefits:

- **Transferability**: They can be fine-tuned for various small labeled data-specific tasks and thus act as versatile bioinformatics tools.
- **Efficiency**: Pre-training can dramatically reduce the computational cost and time that it would normally take to train a model from scratch.
- **Performance**: These models often achieve superior performance, particularly in scenarios where labeled data is scarce.

Inspired by the successful results of DNABERT, which is a pre-trained transformer model for protein sequence prediction. Our model aims to solve the unique challenges of capturing long-range dependencies and complex semantic relationships prevalent in protein sequences.

Influenza Virus and Protein Structure

These proteins are very important to the functionality of the influenza viruses and the diversity in infecting the host cells. Our dataset is so diverse, including the different strains of influenza viruses: haemagglutinin (HA) and neuraminidase (NA), among others. Therefore, each protein contributes to the replication, escape from the immune response, and the pathogenesis of the disease by the influenza viruses in its viral life cycle.

Amino Acids

These are the components of proteins. Each has a unique side chain that will impact the chemical properties and interactions that the protein will have.

Protein sequence

A chain of amino acids linked to each other with peptide bonds. The sequence thus codes for a specific order of amino acids in proteins.

Protein Folding

It is the process by which a linear chain of amino acids folds into a functional 3-D structure of the protein. The folding is driven by various interactions among the amino acids, such as hydrogen bonds, hydrophobic interactions, and ionic bonds.

Thus, each and every protein of influenza has a unique sequence or amino acid signature, as from our dataset. This eventually decides the three-dimensional structure and biological activity of the protein. For example, haemagglutinin is a surface glycoprotein that is very important for binding with cellular receptors and hence plays a very important role in the mechanism of entry of the virus to host cells. Similarly, the other proteins of the virus carry similarly unique sequences and structures, important to protein function and hence to viral replication and pathogenicity. Improved understanding of these sequences allows for the development of strategies that are aimed at the inhibition of functions that the virus performs to avoid getting inside the body.

Methodology

Data Processing

The first phase of the methodology is pre-processing the protein sequence data, which forms the input to the Transformer model. In this section, we describe protein sequence extraction, tokenization, and building the vocabulary.

Sequence Data

The sequences included in the dataset are a collection of protein sequences that are far-reaching; they involve the different strains of the influenza virus, including key viral proteins such as hemagglutinin and neuraminidase, among others, which have equally important functions in the life cycle of the virus and pathogenicity.

Data Extraction

This involves taking raw data files and translating them into a protein sequence, formatted within a FASTA file that contains multiple protein sequences. The sequence for each protein is represented as a string for amino acid residues expressed in single-letter codes, for example: `'MNTQILILAISAFLCVRADKIC'`.

Tokenization

Convert this string into a shorter string, but of length k, termed k-mer. A k-mer is a string of length k that encodes a substring of the length k of the string. For instance, with $k = 3$, `'MNTQIL'` is tokenized to k-m. This tokenization paves the way for the model to work efficiently with the sequence data by breaking long chains into smaller, informative, and manageable sizes.

Vocabulary Creation

Create a vocabulary of unique k-mers from the dataset. The model will use this vocabulary as a set of input and output tokens. The index for each unique k-mer is created to relate the k-mer sequence with the number indices. For instance, k-mer `'MNT'` could be mapped to index 0, `'NTQ'` to 1, and so on.

Further to this, there are special tokens in the vocabulary: for example, `[MASK]` for the masked language modeling task, that allows training when parts of the sequence are masked.

Data Cleaning

Map the protein sequences to numerical sequences with the use of the vocabulary. Then the encoded data is to be split into training, validation, and testing sets. Usually, in this process, 70% of the data is considered for training, 15% for validation, and 15% for testing.

Training

The training phase of our ProteinBERT model involves several critical steps, from data preparation to model optimization. Here, we outline the methodology and key aspects of the training process, detailing how the model is trained to predict amino acid sequences in protein data from the influenza virus.

Model Architecture

Transformer Model:

- We utilize a Transformer model, well-known for its effectiveness in capturing dependencies in sequential data through self-attention mechanisms.
- The model comprises multiple layers and attention heads, which allow it to learn complex relationships within the sequences.

Training Process

Training Loop:

- The model is trained for a maximum of 500 iterations. In each iteration, a batch of data is processed, and the model parameters are updated to minimize the training loss.
- The training loop involves:
  - **Batch Preparation:** Randomly selecting start indices and creating batches of sequences and targets.
  - **Masking:** Applying random masking to a percentage of the input k-mers to train the model using a masked language modeling objective.
  - **Forward Pass:** Passing the masked sequences through the model to obtain logits and compute the loss.
  - **Backward Pass and Optimization:** Calculating gradients and updating model parameters using the AdamW optimizer.

Evaluation:

- The model's performance is evaluated on the validation set at regular intervals (every iteration) to monitor progress and adjust training strategies as needed.
- The validation loss is calculated to gauge how well the model is generalizing to unseen data.
- Early stopping is implemented to prevent overfitting and ensure efficient training. If the validation loss does not improve for a set number of iterations (patience), training is halted.

Early Stopping:

- Early stopping is triggered based on two conditions: achieving a target validation loss threshold or exceeding a patience counter without improvement.
- This mechanism helps in conserving computational resources and ensuring the model does not overfit the training data.

Saving the Model:

- Once training is complete, or early stopping is triggered, the final model is evaluated on the test set to determine its performance on completely unseen data.
- The trained model is then saved for future inference and fine-tuning tasks.

By leveraging a robust Transformer architecture and a systematic training approach, this model is optimized to capture long-range dependencies and semantic relationships in protein sequences, enhancing its accuracy and utility in biological research and applications such as vaccine development and antiviral drug design.

Results

We experimented with different k-mer sizes (3, 4, 5, 6) to determine the most effective sequence length for our model. The results indicated that the best performance was achieved with k-mer sizes of 3 and 5.

Loss Evaluation

- **Final Loss for K-mer 3:** 0.0017
- **Final Loss for K-mer 5:** 0.0019

Training and Validation Loss

Training Loss: The training loss consistently decreased over the iterations, indicating that the model was effectively learning the patterns within the protein sequences.

Validation Loss: The validation loss also showed a significant decrease, suggesting that the model generalizes well to unseen data.

Success of the Loss Function

Cross-Entropy Loss: We employed the cross-entropy loss function, which is suitable for classification tasks. This loss function was effective in minimizing the error between the predicted amino acid sequences and the actual sequences.

Efficient Learning: The loss curves for both training and validation demonstrate rapid convergence, highlighting the efficiency of our model and the robustness of the loss function.

Overall, the results underscore the efficacy of our approach, with the model achieving low loss values and demonstrating strong predictive capabilities for protein sequences using the selected k-mer sizes. This success is pivotal for applications in biological research, including vaccine development and antiviral drug design.