Neural Network-based Protein Function Prediction

Bren Jay Magtalas, Daniel De Castro *†
{bcmagtalas2, ddecastro2}@up.edu.ph

***Abstract*—The study[1] presents a neural network-based approach for protein function prediction. The model utilizes protein sequences and their corresponding ground truth labels to train a multi-label classification model. The architecture of the model consists of several dense layers with the ReLU activation function. The model is trained using the Adam optimizer and the binary cross-entropy loss function. The performance of the model is evaluated using metrics such as binary accuracy and Area Under the Curve (AUC). The results show that the model achieves high accuracy and AUC scores, indicating its effectiveness in protein prediction tasks. The study highlights the potential of neural networks in protein function prediction and suggests further research in this area.**

***Index Terms*—protein function prediction, multi-label classification, gene ontology, GO term, neural network**

I. Introduction

PROTEINS play a crucial role in the body by fulfilling various essential functions and responsibilities. These include maintaining DNA structure, promoting muscle growth, supporting the production of antibodies, enhancing immunity against diseases, and performing numerous other vital tasks [1]. Proteins are large molecules consisting of 20 distinct types of building blocks called amino acids. The human body makes a large number of proteins, and each protein is composed of amino acids linked sequentially. The protein's unique sequence of amino acids determines its structure and function, allowing it to carry out a vast array of biological tasks. The characterization and annotation of protein functions serve as valuable resources for a wide range of sensitive biological and computational applications. These applications include critical areas such as developing new drugs and therapies for various diseases.

Traditional experimental procedures for discovering protein functions in genomics laboratories can often be costly and labor-intensive. The process may require significant resources, including specialized equipment, and skilled personnel. Researchers typically invest substantial time and effort into conducting experiments, performing data analysis, and interpreting the results to determine protein functions. Computational-intelligent methods and algorithms have been introduced to more efficiently predict protein functions from amino acid sequences. These approaches leverage machine learning and statistical modeling to analyze large datasets of protein sequences and infer their functions based on similarities to known proteins [2].

[1]Our research paper is inspired by the CAFA 5 Kaggle competition. At the time of writing, our model achieved a score of 0.47434 on the competition leaderboard. It is important to note that the leaderboard is calculated using approximately 50% of the test data, and the final results will be based on the other 50%. Therefore, the final standings may differ from those reported in this paper.

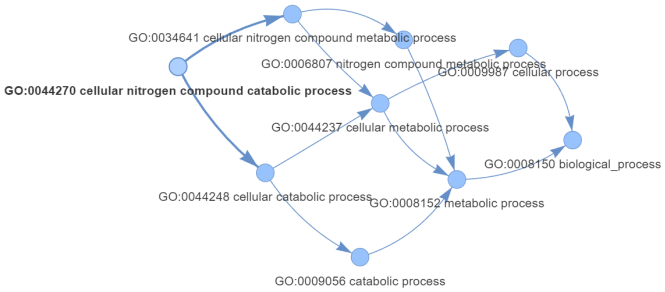

Fig. 1: Cellular Nitrogen Compound Catabolic Process (GO:0044270)

Assigning a specific function to a protein can be challenging due to several factors such as proteins having multiple functions, along with their ability to interact with multiple partners. Proteins can participate in diverse biological processes and carry out different roles within a cell or organism. Gene Ontology (GO) is a hierarchical framework that provides a systematic description of the biological functions of proteins. It offers a model that captures the multifaceted nature of protein function at various levels of abstraction [3].

GO is represented as a directed acyclic graph, where functional descriptors (terms or classes) serve as nodes connected by relational links such as "is_a" and "part_of". This graph-based approach facilitates the organization and classification of protein functions, providing a framework to capture the relationships and dependencies between different functional descriptors within the GO. The graph comprises three distinct subgraphs, known as subontologies: Molecular Function (MF), Biological Process (BP), and Cellular Component (CC). These subontologies are defined by their respective root nodes, which serve as the starting points within each subgraph. MF focuses on describing the specific biochemical activities or tasks performed by proteins, BP captures broader biological events or processes in which proteins participate, and CC describes the subcellular structures or locations where proteins are active [4].

The protein's function is represented by a subset of one or more of the subontologies. By associating a protein with relevant terms in the appropriate subontologies, researchers can effectively capture and represent the diverse aspects of its function. Annotations of protein functions within the GO are accompanied by evidence codes that can be categorized into experimental and non-experimental. Experimental evidence codes are based on direct experimental observations or measurements that provide empirical support for a specific protein function. Non-experimental evidence codes rely on indirect sources of evidence, such as computational predictions

and sequence similarities.

The objective of this project is to predict the function of a set of proteins based on their amino acid sequence using a neural network. The term-protein assignments used are based on experimentally determined data, where each protein is labeled with specific terms that represent its validated functions supported by experimental evidence. This assignment of terms serves as class labels for the proteins. By utilizing these annotated terms, a dataset can be generated, consisting of protein sequences and their corresponding ground truth labels for each term.

II. Methodology

The neural network model employed in this study is designed specifically for protein prediction in a multi-label classification problem. The model architecture and its rationale are outlined as follows.

A. Exploratory Data Analysis

The 2023-01-01 release of ontology data in GO graph structure is stored in *go-basic.obo*. The file is in OBO format and can be parsed by the `obonet` Python package. The data gives all the current protein function annotations available, their definition, and all of their related nodes. Each node is indexed by the term name with the format `'GO:XXXXXXX'`, where X is a single digit. The data allows for the visualization of nodes and the relationship ties between them like the example shown in Figure 1.

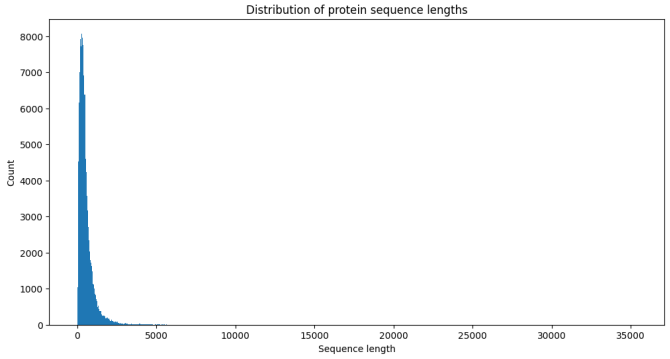

Fig. 2: Protein Sequence Length Distribution

The protein sequences for the training dataset are stored in *train_sequences.fasta*. The file is in the standard FASTA used for describing protein sequences. The data were retrieved from the European Bioinformatics Institute UniProt dataset. To read and parse the sequences, the `Biopython` package was utilized. A count of 142,246 sequences was recorded in total. The distribution of protein sequence length can be seen in Figure 2. The majority of proteins have sequence lengths of less than around 2700 amino acids.

Amino acids are denoted by uppercase letters of the alphabet. Figure 3 contains the amino acid composition for all protein sequences. The frequency distribution provides information on the protein's structure and function. Leucine (L), serine (S), alanine (A), and glycine (G) are the most abundant

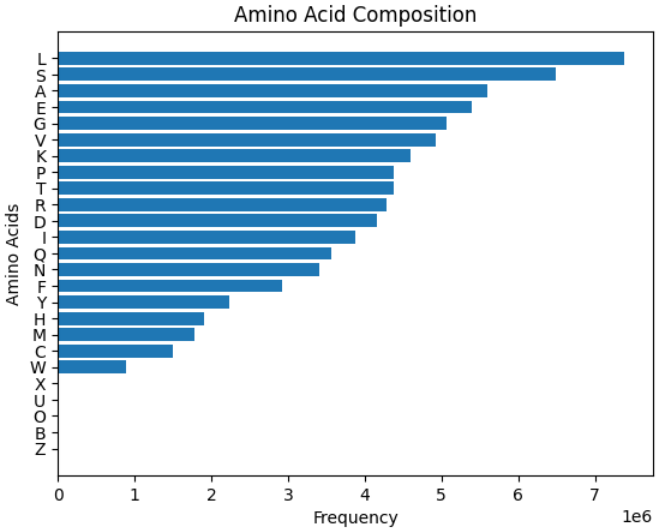

Fig. 3: Amino Acid Distribution

amino acid in the dataset while cysteine (C), methionine (M), tryptophan (W), and histidine (H) are the least common ones. Although there are typically less abundant amino acids, they play significant roles in specific protein functions. Their scarcity emphasizes their specialized importance in protein functionality. The presence of ambiguous amino acid symbols such as X, B, and Z, as well as the rare amino acids O and U, indicates that certain protein sequences may be incomplete or contain errors [5].

B. Data Pre-Processing

The annotated terms or ground truth labels for the protein sequences are stored in *train_terms.tsv*. The three columns in the file represent the protein ID, GO term ID, and the ontology aspect it belongs. The neural network model needs to predict the terms or functions of a protein sequence. Protein sequences are identified by protein ID and its function is denoted by the GO term ID. Each sequence can have many functions, thus the requirement is solving a multi-label classification problem.

The alphabetic protein sequences cannot be used directly in simple neural networks due to the variability in sequence lengths and the inability of the model to process text-based input. To train the model, the protein sequences are converted into vectors using embeddings, similar to the word embeddings used in NLP. A publicly available pre-trained protein embedding model based on Rost Lab's T5 protein language model was utilized by Fironov to generate the given training data stored in *train_embeds.npy* [6]. The protein IDs labels are separately stored in *train_ids.npy*.

The embedded data are loaded and transformed into a pandas dataframe for convenient manipulation. Each protein sequence embedding is a vector with 1024 elements, resulting in a dataframe with 142,246 sample rows with 1024 feature columns. The current labels are only representative of the protein identification and not its function. The protein ID and GO term ID need to be processed to be multi-label one-hot encoded data for it to be suitable in a classification task. A

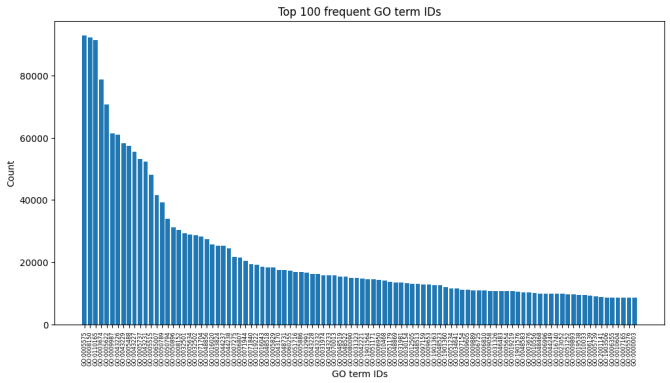

Fig. 4: Label Frequency

problem that can be encountered when processing the data labels is the huge number of available GO term IDs. To simplify the model, out of more than 40,000 terms, only 1,000 most frequent functions are selected. The plot of the 100 most frequent GO term IDs is in Figure 4. The training labels are now reduced to 142,246 protein sequences with their corresponding 1,000 possible functions. The distribution of the ontology aspect after term reduction can be seen in Figure 5, which shows a fraction that is very similar to the original data. In the labels dataframe, each element is assigned a value of 1 if the protein sequence is associated with the specific function, and 0 if it is not. The dataset is then subsequently divided into three sets, training, validation, and testing. The split is performed with a distribution of 64% for training, 16% for validation, and 20% for testing.

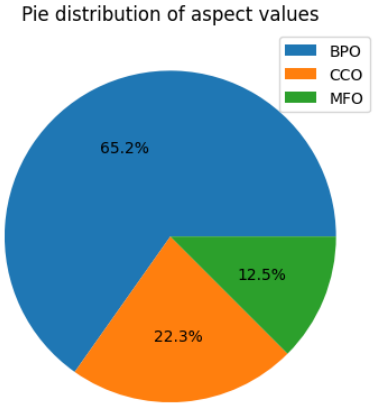

Fig. 5: Aspect Distribution

C. Model Architecture

The input shape of the model is determined by the number of features in the training data, which in this case is 1024 - the size of the embedding vector. This ensures that the input layer of the neural network is compatible with the shape of the input data.

The model consists of several densely connected layers. The first layer is a `BatchNormalization` layer, which normalizes the input features, aiding in stabilizing and accelerating the training process.

Subsequent layers are implemented as `Dense` layers. These layers learn complex patterns and representations from the input data. The `relu` activation function is used in each dense layer to effectively model non-linear relationships within the data.

The number of units in each dense layer is configured as follows: 2000, 1000, 500, and 2000. This architecture allows the network to extract and condense higher-level features while maintaining sufficient representation capacity. The last dense layer consists of `num_of_labels` units and utilizes the `sigmoid` activation function, enabling the model to perform multi-label classification by independently predicting the probability of each label.

D. Model Optimization

The model is compiled with the Adam optimizer, using a learning rate of 0.001. The binary cross-entropy loss function is chosen to suit the multi-label classification task. It measures the dissimilarity between the predicted probabilities and the actual labels for each protein. By optimizing this loss function, the model learns to minimize the discrepancy between its predictions and the ground truth, enabling it to make accurate predictions for multiple labels simultaneously.

Two evaluation metrics are defined: binary accuracy, which is suitable for multi-label classification tasks where each label is treated independently, and Area Under the Curve (AUC) which measures the ability of the model to rank the predicted probabilities correctly, indicating how well the model can discriminate between positive and negative instances for each label.

E. Training Process

During the training process, the model is fitted using the `fit` function. The training features and labels are provided, along with validation data for monitoring the model's performance on unseen data. A batch size of 2000 is set to determine the number of samples processed in each training iteration. The model is trained for 20 epochs, representing the number of complete passes through the training data.

During the training process, the model's performance is monitored by observing the values of the loss function and evaluation metrics on both the training and validation sets. In the provided epoch results, the training was stopped once it was noticed that the validation loss started to increase after reaching its lowest point.

The decision to stop the training was made when it was observed that the validation loss started to increase after reaching its lowest possible value. This step was taken to prevent overfitting, as further training might lead to a decline in generalization performance.

III. Results and Discussion

In the final epoch of training, the neural network model achieved a loss of 0.0584, indicating the extent of dissimilarity

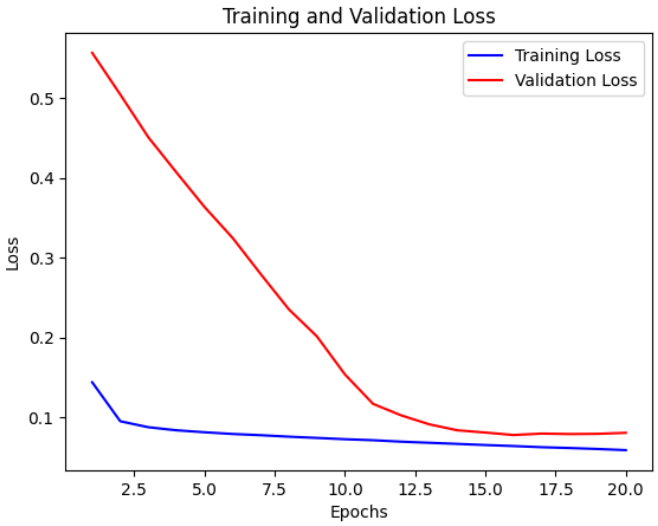

Fig. 6: Training and Validation loss

between the predicted and actual labels. The model's binary accuracy reached 0.9793, indicating the proportion of correctly predicted labels out of all the labels. Moreover, the Area Under the Curve (AUC) metric attained a value of 0.9651, demonstrating the model's ability to rank predicted probabilities accurately. These metrics reflect the high performance of the model in capturing protein-related patterns and relationships. The validation results from the last epoch yielded a validation loss of 0.0801, a validation binary accuracy of 0.9738, and a validation AUC of 0.9228. These metrics showcase the model's generalization capability on unseen data, further validating its effectiveness in protein prediction tasks.

The neural network model was tested and evaluated on a dataset consisting of 28,450 protein samples. After the testing process, the model achieved a final loss of 0.0808, a binary accuracy of 0.9736, and an AUC of 0.9224 on the testing set. These metrics indicate the model's effectiveness in capturing protein-related patterns and relationships, as well as its ability to accurately predict the presence or absence of protein labels. The summary of all the results is in TABLE 1.

Metrics	Loss	Accuracy	AUC
Train	0.0584	0.9793	0.9651
Validation	0.0801	0.9738	0.9228
Test	0.0808	0.9736	0.9224

TABLE I: Neural Network Performance

The achieved performance highlights the effectiveness of the neural network model in protein prediction tasks. The low loss and high binary accuracy indicate the model's ability to accurately classify proteins with multiple labels. Moreover, the AUC score suggests that the model can effectively rank predicted probabilities, further enhancing its predictive capabilities.

The results obtained from this study support the utilization of the neural network model for protein prediction. The model's ability to capture intricate relationships within the protein data and make accurate multi-label predictions holds great promise for various applications, such as protein function prediction and drug discovery.

Overall, the results demonstrate the effectiveness and potential of using neural networks in protein prediction tasks. Further research and experimentation can be conducted to explore different architectures, hyperparameter tuning, and additional datasets, leading to even more accurate and robust protein prediction models.

IV. Conclusion

In this study, we developed and evaluated a neural network model for protein prediction in a multi-label classification setting. The model demonstrated strong performance, achieving low loss and high accuracy on both the training and validation sets. These results highlight the model's effectiveness in capturing complex protein relationships and accurately predicting protein labels.

The utilization of the binary cross-entropy loss function, along with binary accuracy and AUC as evaluation metrics, proved suitable for the protein prediction task. The model's ability to accurately predict protein labels holds great promise for various applications, such as bioinformatics and drug discovery.

While this study had limitations, including a subset of labels and dataset size, the success of the neural network model suggests the potential for further advancements. Future research should focus on expanding label coverage and considering larger and more diverse protein datasets to validate and generalize the findings.

In conclusion, the neural network model presented in this study shows promise in accurately predicting protein labels. With further research and improvements, such models have the potential to contribute significantly to the analysis and understanding of proteins, leading to advancements in various biological and medical applications.

References

[1] M. E. M. Elhaj-Abdou, H. El-Dib, A. El-Helw, and M. El-Habrouk, "Deep_CNN_LSTM_GO: Protein function prediction from amino-acid sequences," *Computational Biology and Chemistry*, vol. 95, p. 107584, 2021. doi:10.1016/j.compbiolchem.2021.107584

[2] M. Kulmanov, M. A. Khan, and R. Hoehndorf, "DeepGO: Predicting protein functions from sequence and interactions using a deep ontology-aware classifier," *Bioinformatics*, vol. 34, no. 4, pp. 660–668, 2017. doi:10.1093/bioinformatics/btx624

[3] M. Ashburner et al., "Gene ontology: Tool for the unification of biology," *Nature Genetics*, vol. 25, no. 1, pp. 25–29, 2000. doi:10.1038/75556

[4] "Gene ontology overview," Gene Ontology Resource, http://geneontology.org/docs/ontology-documentation/ (accessed Jun. 13, 2023).

[5] "3AA-20 to 3AA-21," 3AA-20 and 3AA-21, https://iupac.qmul.ac.uk/AminoAcid/A2021.html (accessed Jun. 13, 2023).

[6] Sergeifironov, "T5embeds calculation [only few samples]," Kaggle, https://www.kaggle.com/code/sergeifironov/t5embeds-calculation-only-few-samples (accessed Jun. 13, 2023).