Predicting Protein Family Classification using Deep Learning Models

Zhehui B. Liu

June 19, 2024

Abstract

The sequence of protein determines its function. The relationship between the sequence and function is a fundamental problem in biology and remains to be investigated. This study employs deep learning models for predicting protein family classifications using unaligned amino acid sequences from the Pfam database. Two models, Bidirectional LSTM and ProtCNN, were developed and evaluated. ProtCNN, inspired by ResNet architecture with dilated convolutions, demonstrated superior performance,reaching prediction accuracy of approximately 0.90, using only top 50 family classes. Future research may explore transformer-based models, like ProtBERT, which could better capture sequence variability and contextual relationships.

1 Introduction

This report examines deep learning methods for predicting protein family classifications based on amino acid sequences. The dataset is obtained via a random split of the Pfam dataset. The dataset contains labels of protein sequences and protein family classes, as well as aligned sequences using hidden Markov models [3]. However, this report focuses on learning from unaligned sequence information to infer classes.

2 Data Analysis and Preparation

The dataset is separated into three sets: train, dev, and test, with sizes of 1,086,741 (81%), 126,171 (9%), and 126,171 (9%), respectively. There are 17,929 unique families in the training dataset, but the distribution of protein families is skewed towards a few classes with large quantities of sequences (see Figure 1). To manage this imbalance and reduce computational load, a subset of the top 50 protein families was selected, reducing the dataset size by 95%. Sequence lengths were capped at 183 amino acids, which is at the 80% threshold of the sequence length distribution (see Figure 2), and sequences were padded to this

length. Amino acid codes were encoded into integers, and rare amino acids were marked as 0. Target labels were one-hot encoded using scikit-learn.

3 Model Selection and Explanation

3.1 Bidirectional LSTM

LSTM networks [4] are a specialized type of Recurrent Neural Network (RNN) capable of learning long-term dependencies, making them particularly suitable for sequence data like protein sequences. LSTMs are designed to remember information over extended periods, addressing the vanishing gradient problem that traditional RNNs face. In this study, a Bidirectional LSTM was chosen due to its ability to process sequences in both forward and backward directions, which are properties of protein sequences, thus preserving context from both past and future. This model is proposed for protein sequence learning because the context of an amino acid within the sequence can impact its role in the protein's function and thereby determine the protein family it is classified into.

The Bidirectional LSTM architecture was built from these several components. First, an embedding layer converts amino acid indices into dense vectors of fixed size, providing a meaningful representation of each amino acid. The number of unique tokens which represent each unique amino acids in this case is set at 21 for total 20 amino acids and 1 rare amino acid or padding token as 0. Following this, the Bidirectional LSTM layer processes the sequence in both directions, capturing dependencies across the entire sequence. Subsequent fully connected layers apply transformations to produce the final output, which is then passed through a softmax layer to generate a probability distribution over the protein families. To optimise this model, several hyperparameters were tuned, including the number embeddings, dropout rates, and learning rates. Dropout [6] and Early stopping was implemented to prevent overfitting.

3.2 ProtCNN

ProtCNN, used in this article [1], is based on the ResNet architecture [5], which employs residual blocks and dilated convolutions to effectively learn hierarchical features from protein sequences. Residual blocks address the vanishing gradient problem by providing alternate pathways for the gradient to flow through, ensuring stable training of deeper networks. Dilated convolutions expand the receptive field without increasing the number of parameters, allowing the model to capture larger contexts efficiently. The rationale for choosing CNNs for its demonstrated ability to learn complex patterns and hierarchical features, in various domains in addition to computer vision tasks.

The ProtCNN architecture begins with an initial convolution layer that extracts basic features from the input sequences. This is followed by multiple residual blocks, each consisting of two convolution layers with batch normalization, ReLU activation, and dilation. The dilated convolutions enhance the

model's ability to capture larger contexts within the sequences. After the residual blocks, a pooling layer reduces the spatial dimensions, retaining the most important features. Fully connected layers then transform the feature maps into the final predictions, which are passed through a softmax layer to produce the probability distribution over the protein families. The optimisation process for ProtCNN involved extensive hyperparameter tuning, including varying the dilation rates, kernel sizes, and the number of residual blocks. Regularisation techniques, such as dropout, were applied to prevent overfitting, and adaptive learning rate scheduling ensured efficient training.

4 Results

4.1 Optimisation and Testing

The optimisation of both models involved a systematic and thorough approach. For the Bidirectional LSTM, initial tests were conducted to determine the optimal number of embeddings, experimenting with values ranging from 64 to 256. Dropout rates were varied between 0.2 and 0.5 to assess their impact on preventing overfitting. Learning rates were tested within the range of 0.001 to 0.01, with the final model using a rate of 0.001. Early stopping was implemented based on validation loss, halting training when no improvement was observed over five consecutive epochs. Batch sizes of 64, 128, and 256 were tested, with 256 providing the best balance between training time and performance. The number of epochs was initially set to 100, but early stopping typically concluded training between 60 and 70 epochs.

For ProtCNN, the optimization process was similarly rigorous. Kernel sizes of 3x3 and 5x5 were compared, with 3x3 kernels providing a good balance between capturing local patterns and computational efficiency. The number of residual blocks was varied from two to five, with four blocks yielding the best results. Dropout rates between 0.2 and 0.5 were tested, with 0.2 chosen to prevent overfitting while retaining high accuracy. Learning rates were initially set at 0.001, with adaptive scheduling reducing the rate by a factor of 0.1 upon plateauing of the validation loss. The final ProtCNN model trained for up to 50 epochs, with early stopping typically concluding training around 10-15 epochs.

4.2 Model Comparison

The Bidirectional LSTM model achieved an approximate accuracy of 0.80 on the test data, demonstrating a relatively good capability to learn sequence context without overfitting. In contrast, the ProtCNN model outperformed the LSTM, achieving around 0.90 accuracy. This superior performance can be attributed to the advanced architecture of ProtCNN, which efficiently captures complex patterns through its hierarchical feature learning. Additionally, ProtCNN converged faster, making it more suitable for large datasets with intricate patterns.

Comparing the two models (see Figure 3), ProtCNN demonstrated superior performance, achieving higher accuracy and faster convergence. While the Bidirectional LSTM was effective in capturing sequence context, it was computationally intensive and slower to converge, costing 130 seconds to finish the training whereas ProtCNN only took about 40 seconds. However, ProtCNN is shown to be prone to be overfitting, which was mitigated through regularization and early stopping. Both models, however, showed the potential of deep learning approaches in predicting protein family classifications based on amino acid sequences.

5 Conclusion

This study explored two deep learning models, Bidirectional LSTM and ProtCNN, for predicting protein family classifications based on amino acid sequences. While both models exhibited strong performance, ProtCNN's advanced architecture and efficient training process led to superior results. Future research will explore transformer-based models like ProtBERT [2], which offer improved accuracy by effectively capturing sequence variability and contextual relationships. Addressing class imbalance through techniques such as oversampling, undersampling, and class weighting will be employed to mitigate the effects of class imbalance. These advancements will contribute to protein function prediction and hopefully accurately expand the Pfam family database.

References

[1] Maxwell L Bileschi, David Belanger, Drew H Bryant, Theo Sanderson, Brandon Carter, D Sculley, Alex Bateman, Mark A DePristo, and Lucy J Colwell. Using deep learning to annotate the protein universe. *Nature Biotechnology*, 40(6):932–937, 2022.

[2] Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 2022.

[3] Sean R Eddy. Accelerated profile hmm searches. *PLoS computational biology*, 7(10):e1002195, 2011.

[4] Alex Graves and Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks

from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

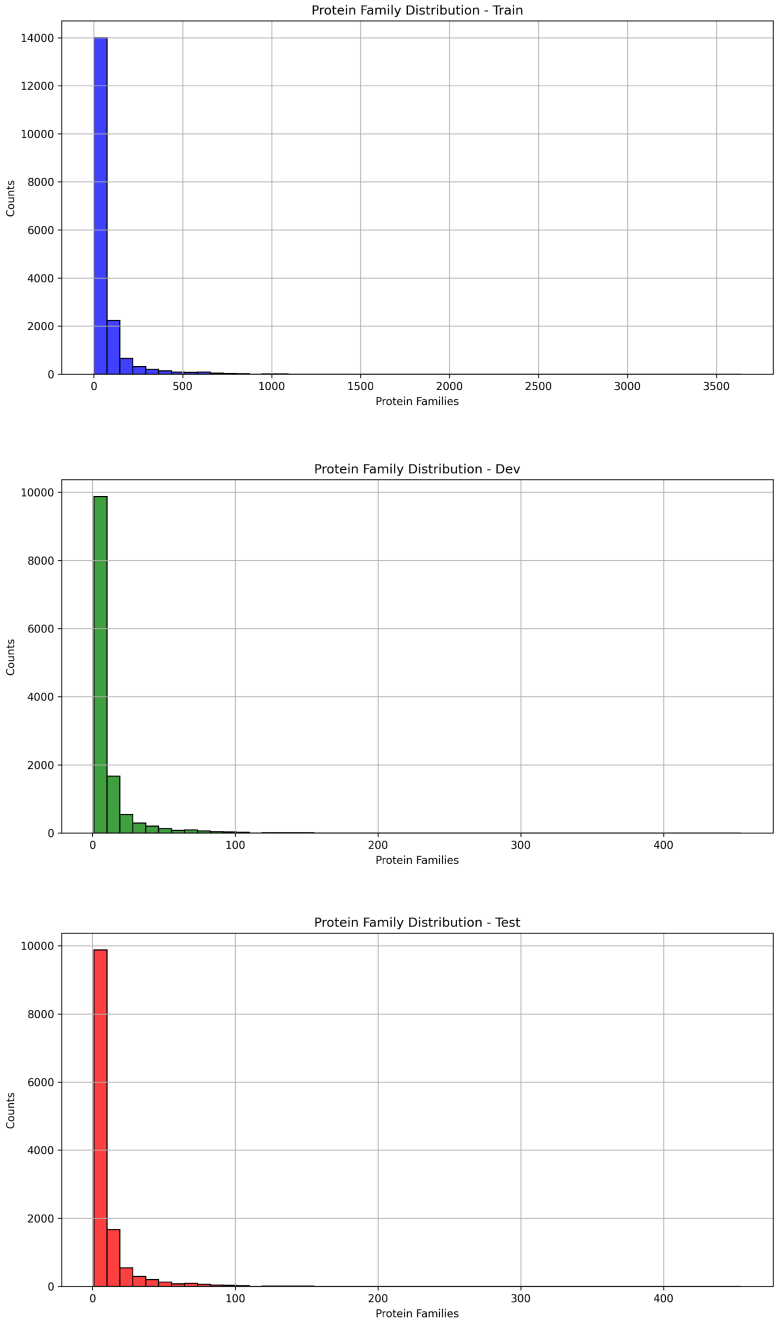

Figure 1: Protein family distribution for three split sets.

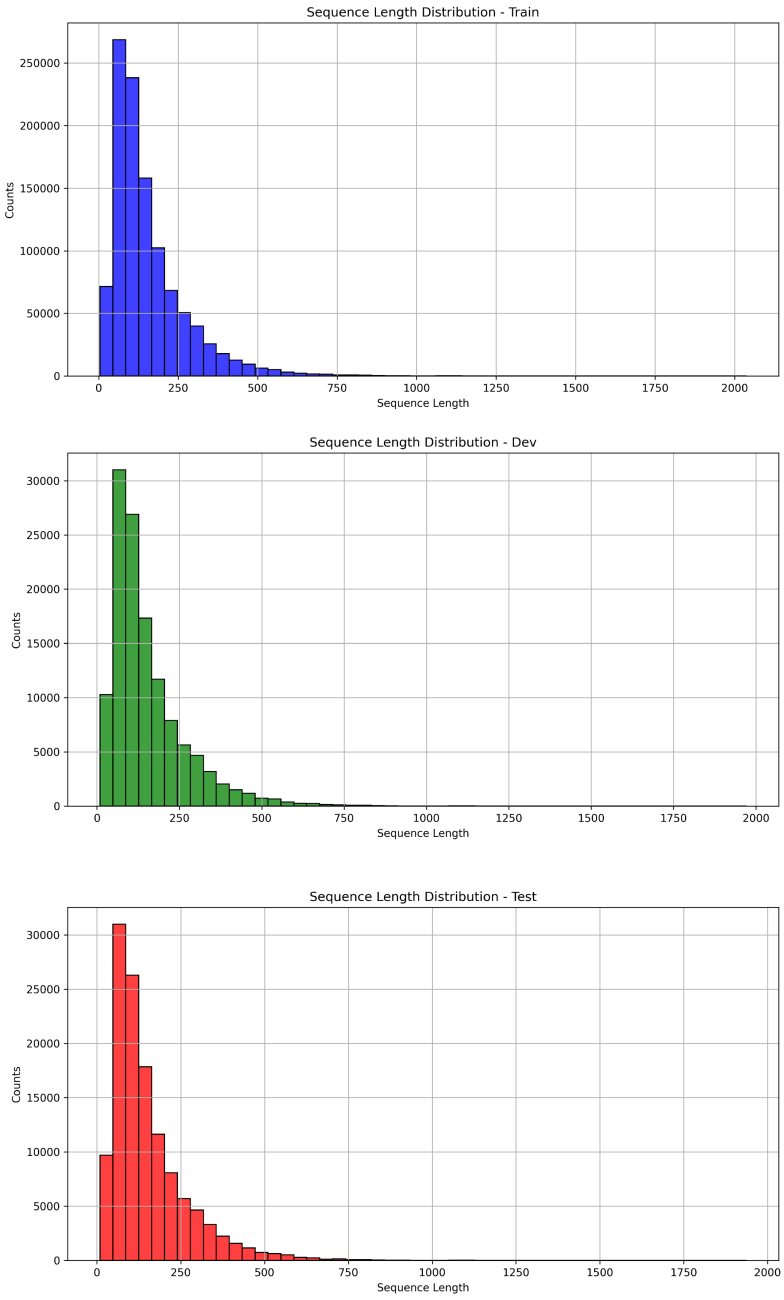

Figure 2: Sequence length distribution for three split sets.

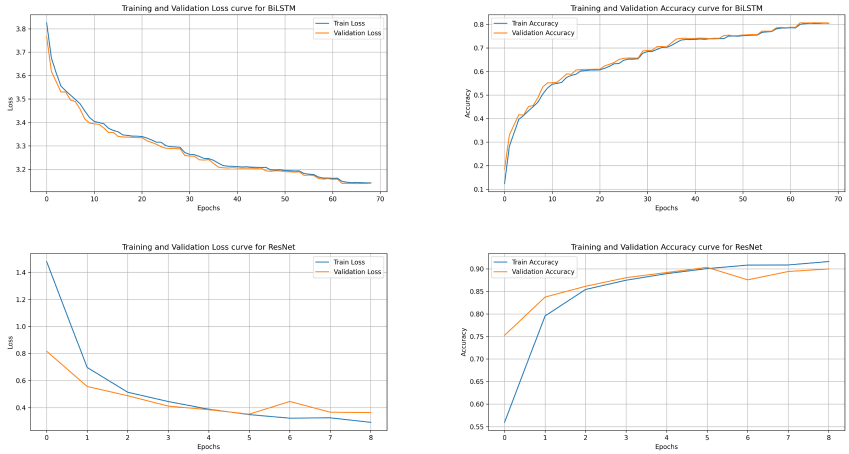

Figure 3: Model comparison between BiLSTM and ProtCNN, based on loss and accuracy curves during training.