Predicting secondary structure of proteins: a comparison between different neural network methods

Zhouhui Qi 2024.3.20

1.Introduction

Protein secondary structure prediction is a critical problem in bioinformatics, serving as the foundation for predicting the complex structures of proteins. This project aims to train neural network models with PSSM of proteins as input and DSSP as labels, predict protein secondary structure, and evaluate the model's performance. This project employs four neural network architectures: MLP, RNN, CNN, and Transformer. Additionally, multiple parameters of the MLP model are adjusted for hyperparameter tuning and ablation experiments.

2.Materials and methods

2.1 Training and test data

DSSP (Dictionary of Secondary Structure of Proteins) is a standard method for describing the secondary structure of proteins (Kabsch & Sander, 1983). It defines a set of symbols to represent the secondary structure state of each amino acid residue, including 'H' for α-helix, 'E' for β-strand and '-' indicates an undetermined secondary structure type. PSSM (Position-Specific Scoring Matrix) is a matrix used to describe the score of an amino acid appearing at a specific position for each residue.

The data of this experiment come from the study of Drozdetskiy et al. (2015). It includes two data sets: training set and test set (Supplementary material).

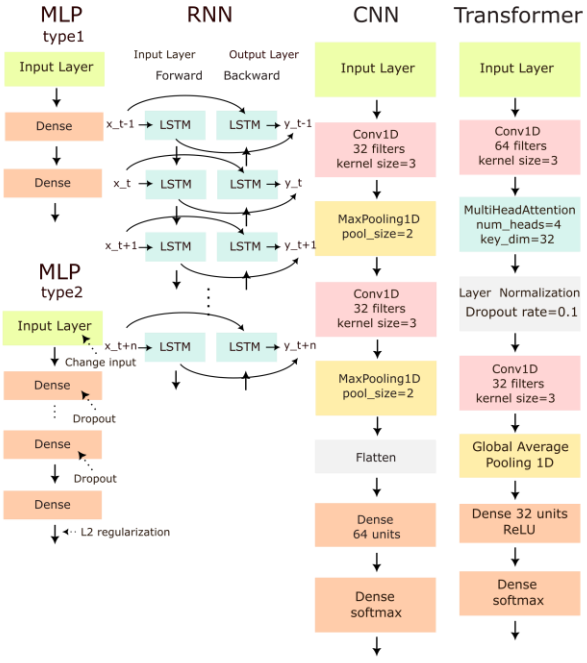

Figure 1. Structural diagrams of different models in this experiment (different types of layers are represented by different colors). MLP is mainly composed of fully connected layers; RNN is composed of multiple LSTM (Long Short-Term Memory) units; CNN is composed of convolution layers, pooling layers, flatten layer and fully connected layers; and Transformer is composed of convolution layers, multihead attention, pooling layer and fully connected layers.

2.2 MLP(Multilayer Perceptron)

The MLP model's inputs are the feature vectors extracted from PSSM files, constructed with the features of all residues within a specific window size. Given that the PSSM files contain two matrices, PSSM and MSA frequency, only one is selected as input. The labels are the DSSP files containing the secondary structure class for each residue. Then, the predictions are compared with labels to compute the loss function.

Additionally, this project also conducts ablation studies to systematically evaluate the impact of some specific components of neural networks by removing or altering them (Meyes et al., 2019). In this experiment, ablation studies and hyperparameter tuning are primarily divided into five parts (Figure 1):

1) Comparison of input variations, contrasting the MLP models based on PSSM input with those based on MSA frequency matrix input.

2) Comparison of models with different maximum iterations (Table 1).

3) Comparison of models with different numbers of neurons and models with different numbers of hidden layers (Table 1).

4) Evaluation of the impact of Dropout on model performance.

5) Evaluation of the impact of L2 regularization on model performance.

2.3 Other methods

The RNN model includes a bidirectional LSTM layer, which processes input sequences considering past and future information with 64 memory cells, and a time-distributed dense layer transforming LSTM output into a probability distribution for each time step, facilitating classification (Figure 1). The CNN (Convolutional Neural Network) model comprises a one-dimensional convolutional layer, a max-pooling layer, a flatten layer, and a fully connected dense layer (Figure 1). The Transformer Model incorporates multi-head self-attention layers and Dropout(Figure 1.).

These models employ the same method of parsing PSSM and DSSP files as the MLP model. However, there are slight differences in the encoding of inputs due to the different types of models employed. Please refer to the code in the supplementary material for further details.

2.4 Scoring indexes

The formulas for accuracy, weighted F1 score, and Matthews correlation coefficient (MCC) are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

3.Results and conclusion

Regarding computational efficiency, the MLP model significantly outperforms the CNN, RNN, and Transformer models.

Regarding performance, most of the models exhibit similar performance levels, achieving around 0.73 accuracy (Figure 2). Among them, the single-hidden-layer MLP model with parameters set to 30 iterations and 40 neurons performs the best, achieving an accuracy of 0.7478. On the other hand, the RNN and Transformer models perform the worst, failing to achieve an accuracy of 0.7.

Table 1. Performance of each model on the test set

	Model	Accuracy	F1 Score	MCC
MLP	Default MLP with MSA(30 iterations, 100 neurons)	0.7332	0.7340	0.5935
	Default MLP (200 iterations, 100 neurons)	0.7194	0.7207	0.5729
	Default MLP (100 iterations, 100 neurons)	0.7266	0.7282	0.5838
	Default MLP (30 iterations, 100 neurons)	0.7392	0.7405	0.6042
	MLP (30 iterations, 40 neurons)	0.7478	0.7490	0.6163
	MLP (30 iterations, 60 neurons)	0.7442	0.7457	0.6113
	MLP (30 iterations, 80 neurons)	0.7429	0.7438	0.6077
	MLP (30 iterations, 200 neurons)	0.7198	0.7213	0.5741
	Deeper MLP (Three layers 128/64/32)	0.7358	0.7368	0.5967
	Deeper MLP (Three layers 64/32/16)	0.7418	0.7431	0.6092
	MLP (L2 Regularized, 30 iterations, 100 neurons)	0.7193	0.7205	0.5739
	Dropout MLP (Dense*3, Dropout0.5*2)	0.7346	0.7337	0.6034
RNN	RNN(Bidirectional LSTM, units=64)	0.6946	0.6916	0.5314
CNN	CNN(Conv1D*2, Pooling1D, Dense*2)	0.7375	0.7394	0.6035
Transformer	Transformer	0.6454	0.6459	0.4635

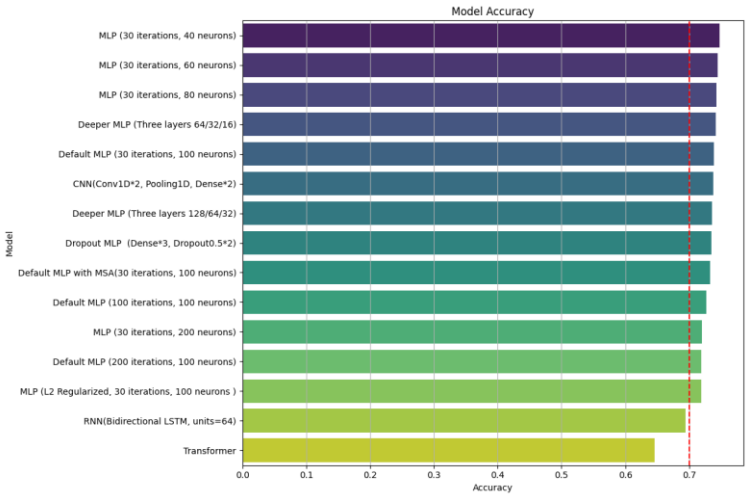

Figure 2. Performance of each model (ranking based on Accuracy). The performance of each model is relatively close. Among them, the simplest MLP with 40 neurons in a single hidden layer performs best, and the Transformer and RNN models perform the worst (0.7 is used as a reference line).

Furthermore, in comparing MLP models with a single hidden layer and multiple hidden layers, the result shows that more layers in the model do not mean better performance (Table 1); meanwhile, the model's performance with more neurons is slightly decreased (Figure 4). Wardah et al. (2019) mentioned that by using BLOSUM62 and a hidden layer with 19 neurons, it is possible to obtain a model that possesses a prediction accuracy of 78%. Hush. (1989) also mentioned that a MLP network with one hidden layer performs better than two hidden layers in network size. These all suggest that increased model complexity does not lead to better prediction accuracies.

Regarding the number of iterations, despite the optimisation still not converged at the maximum number of iterations, 200, the result shows that 30 iterations are sufficient for the model to perform well, and the extra iterations are likely to cause overfitting and thus decrease the accuracy of models (Figure 3).

The input of training data does affect the model's performance. The accuracy of the model using the MSA frequency matrix is 0.7332, whereas the accuracy of the model using the PSSM with the same parameters is 0.7392. This means that the additional physicochemical properties of the amino acids in the PSSM might helped in the prediction.

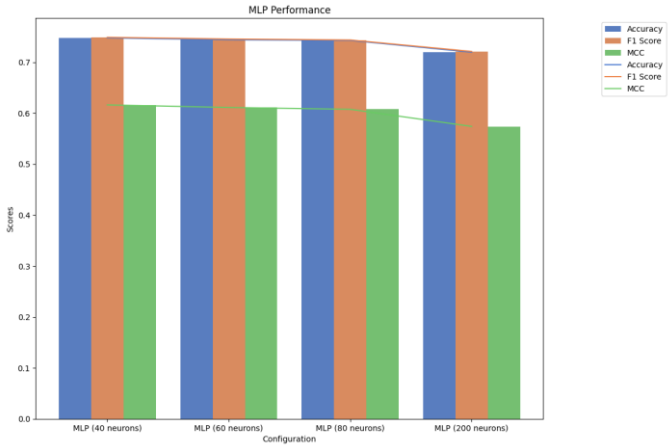

Figure 3. Model performance using different neurons in an MLP with the same maximum iterations. The single hidden layer model with 40 neurons has the best effect, and the increase after more than 40 neurons does not improve the model performance.

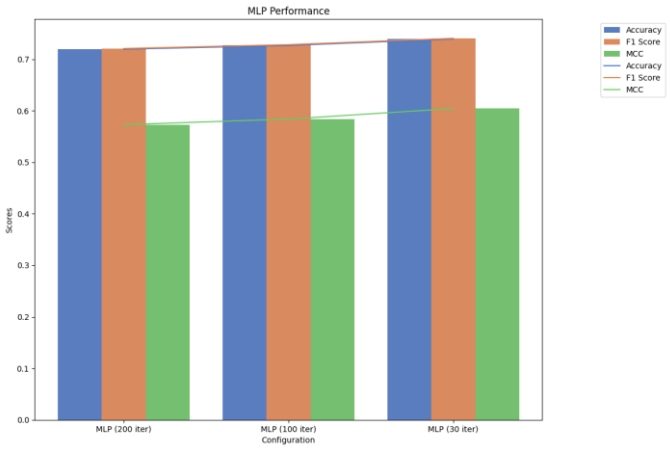

Figure 4. Model performance using different maximum iterations in an MLP with only one hidden layer of 100 neurons. The vertical axis is the score of model performance, the horizontal axis represents different models, and the three colored columns represent accuracy, F1 score and MCC respectively. As the number of iterations increases, the performance of the model decreases slightly.

Regarding the prediction results, the following is an example of the prediction results for the protein primary sequence, the predicted secondary structure, and the confidence corresponding to each position (Figure 5):

MNLTELKNTPVSELITLGENMGLENLARMRKQDIIFAILKQHAKSGE

----HHH---HHHHHHHHHH----H--H--HHHHHHHHHHHHHH---

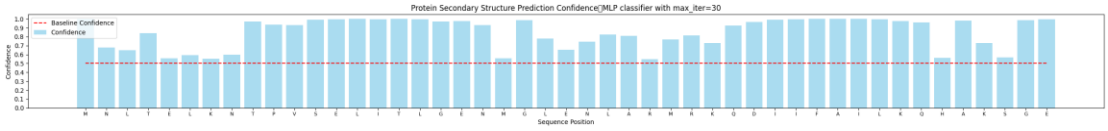

Figure 5. Confidence histogram of model predictions (based on a single hidden layer model with 30 iterations and 40 neurons). Each prediction has a corresponding confidence, and 0.5 is set as the baseline of confidence. The model outputs three categories (H, E, -) for each residue, representing three different secondary structures. The model predicts the probability of each category and then selects the category with the highest probability as the prediction result and uses its probability as the confidence for that prediction

4. Discussion

The improvement in protein secondary structure prediction accuracy originates from three main factors: larger training datasets, enhanced features, and improved machine learning models (Smolarczyk et al., 2020). The accuracy limit of secondary structure prediction currently stands at 88% (for a three-class classification) (Rost, 2001). Hence, the results of these models in the experiment are deemed satisfactory. It is conceivable that superior outcomes could be attained with larger datasets. For Transformer, RNN, and CNN models, the lack of extensive hyperparameter tuning and insufficient training data may be the reasons for their relatively poorer performance despite their more complex structures. However, theoretically, complex structures are expected to yield better results——the convolution and pooling layers in CNN models can encompass and integrate more contextual information, and the memory and attention mechanisms in LSTM and Transformer models can aid in decision-making.

This report only explores some of the hyperparameter tuning for different MLP models. Numerous combinations of model hyperparameters remain to be explored.

5.Supplementary material

The experimental data comes from https://github.com/katarinaelez/protein-ss-pred.git. Code can be found at: https://colab.research.google.com/drive/1_PIcELY-1x0CFbegKCL8yT1t4JcOIil_#scrollTo=EixFMfOqJteh

References

Drozdetskiy, A., Cole, C., Procter, J., & Barton, G. J. (2015). JPred4: a protein secondary structure prediction server. *Nucleic acids research*, *43*(W1), W389-W394.

Hush. (1989, August). Classification with neural networks: a performance analysis. In *IEEE 1989 International Conference on Systems Engineering* (pp. 277-280). IEEE.

Kabsch, W., & Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen‐bonded and geometrical features. *Biopolymers: Original Research on Biomolecules*, *22*(12), 2577-2637.

Meyes, R., Lu, M., de Puiseau, C. W., & Meisen, T. (2019). Ablation studies in artificial neural

networks. *arXiv preprint arXiv:1901.08644*.

Rost, B. (2001). Protein secondary structure prediction continues to rise. *Journal of structural biology*, *134*(2-3), 204-218.

Smolarczyk, T., Roterman-Konieczna, I., & Stapor, K. (2020). Protein secondary structure prediction: a review of progress and directions. *Current Bioinformatics*, *15*(2), 90-107.

Wardah, W., Khan, M. G., Sharma, A., & Rashid, M. A. (2019). Protein secondary structure prediction using neural networks and deep learning: A review. *Computational biology and chemistry*, *81*, 1-8.