Deep Neural Network Models for Protein Struction Classification

Xuewei Wu

Abstract

Motivation: Given a protein of unknown function, searching for its homologous protein is an effective method to predict its function. The SCOP database classifies proteins according to the relationship between structure and evolution. From top to bottom, they are "Class", "Fold", "Super" and "Family". The motivation of Protein Struction Classification is to obtain the label of given protein struction which includes "Class", "Fold", "Super" and "Family".
Result: We use ContactLib to preprocess protein data to obtain structural information including coordinates, dihedral angles, etc. We have implemented a variety of neural networks for protein structure classification including LinearNet, CNN, ResNet, DenseNet, and TransNet. In the training process, we use the coordinate information between contacts as the input of the network, which is actually a manifestation of protein structure information. We observe that TransNet is the most effective among the above networks. The reason may be that TransNet effectively uses global structural information, which is exactly what the convolutional network ignores.
Implementation: https://github.com/xueweiwujxw/Bioinformatics_project2

1 Introduction

Homologous proteins have a common ancestor, and their functions and evolutionary relationships are similar. For protein biology research, finding homologous proteins is often very critical.

The Structural Classification of Proteins (SCOP[1]) database is a largely manual classification of protein structural domains based on similarities of their structures and amino acid sequences. The SCOP database classifies proteins according to the relationship between structure and evolution. From top to bottom, they are "Class", "Fold", "Super" and "Family". The broadest groups on SCOP version 1.75 are the protein fold classes. These classes group structures with similar secondary structure composition, but different overall tertiary structures and evolutionarily origins. Each class contains a number of distinct folds. This classification level indicates similar tertiary structure, but not necessarily evolutionary relatedness. Domains within a fold are further classified into superfamilies. This is a largest grouping of proteins for which structural similarity is sufficient to indicate evolutionary relatedness and therefore share a common ancestor. Protein families are more closely related than superfamilies. Domains are placed in the same family if that have either: 1.>30% sequence identity. 2.some sequence identity (e.g., 15%) and perform the same function.

Here, we have made different attempts mainly for the network based on the ContactLib-DNN[2] method. We divide the structure of the protein into multiple contacts based on the peptide bonds and hydrogen bonds of the protein, and extract the coordinates, dihedral angles and other information. According to the structure of the SCOP database, we have trained a neural network model to obtain the class, fold and super labels of the query protein. With this neural network model, we can classify a protein which function is unknown and find its homologous protein to study its evolutionary relationship and protein function.

2 Method

We divide the SCOP database into training and testing according to protein release for simulating known and unknown protein data sets, and divided training into training and validation for model training. The coordinates of contact can reflect the global structural information of the protein. However the coordinates will change with the rotation and translation of the protein, we convert the coordinates into a distance matrix for more general information. In order to speed up training, we divide the data into three different sizes: 40%, 95%, and 100%. After data processing, we use different networks including LinearNet, CNN, ResNet, DenseNet and TransNet for classification. At the same time, we use data sets of different sizes as the input of TransNet in order to verify the

impact of data sets of different sizes on the results. Finally, we test the performance of the model on the testing data set to verify the efficiency of the model on unknown proteins

2.1 LinearNet

LinearNet is a traditional fully connected neural network as a performance benchmark. Figure 1 is the network structure of LinearNet, including embed and encod. Embed unifies the dimensions of the input contacts because of the different number of contacts read from data each time. Encod contains multiple fully connected neural network. LinearNet gets three outputs corresponding to the three labels of class, fold, and super.

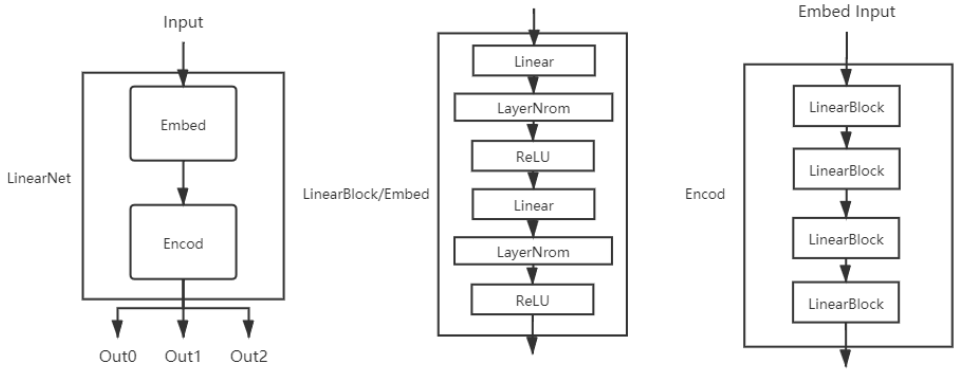


Fig. 1. LinearNet network structure. The left one is the global structure of LinearNet which contains Embed and Encod, the right one is the structure of Encod which contains multiple LinearBlocks, the middle one is the structure of Embed and LinearBlock which contains multiple combinations including Linear, LayerNorm, and ReLU.

2.2 ConvolutionNet

CNN[3] is similar to LinearNet, replacing multiple fully connected layers in Encod with multiple convolutional layers. Figure 2 is the network structure of CNN.

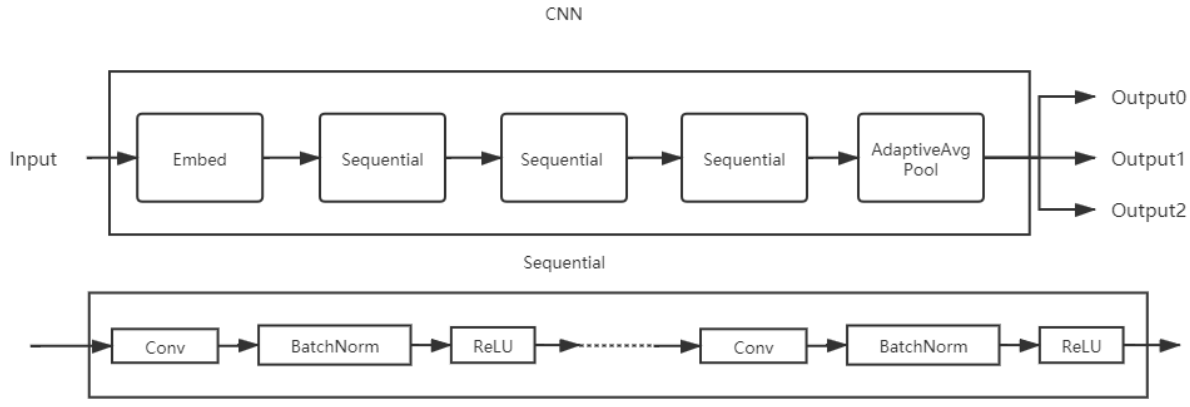


Fig. 2. CNN network structure. CNN replaces encod with multiple sequential, which contains multiple combinations including conv, batchNorm, and ReLU.

2.3 ResNet

ResNet[4] adds a residual unit on the basis of CNN to solve the gradient disappearance caused by increasing depth. Figure 3 is the network structure of ResNet.

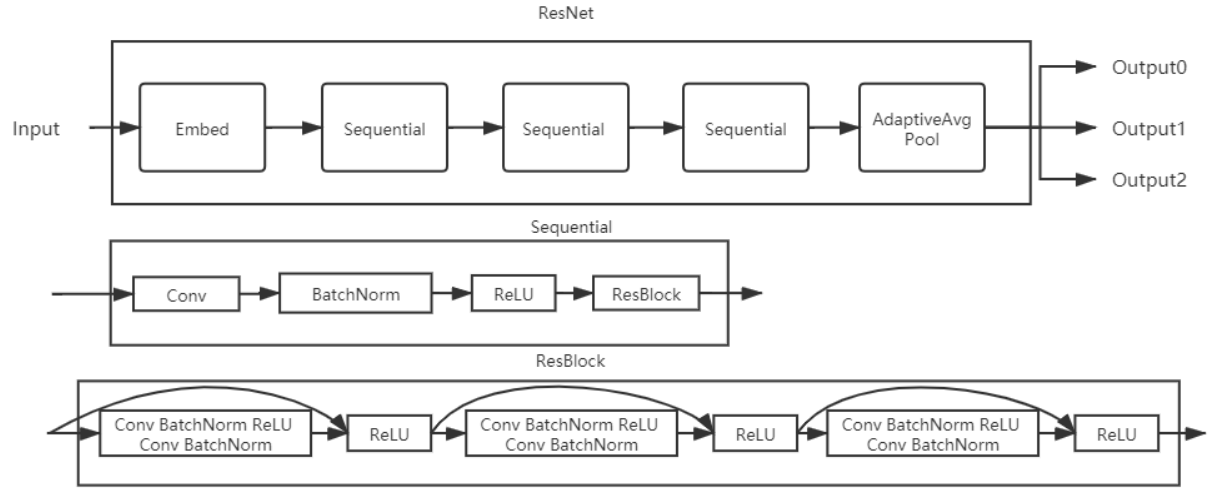


Fig. 3. ResNet network structure. ResNet adds the input and output of the previous layer, and activates it with the ReLU function as the input of the next layer.

2.4 DenseNet

Compared with ResNet, DenseNet[5] proposes a more radical dense connection mechanism that connects all layers to each other. Specifically, each layer will accept all the previous layers as its additional input. Figure 4 is the network structure of DenseNet.

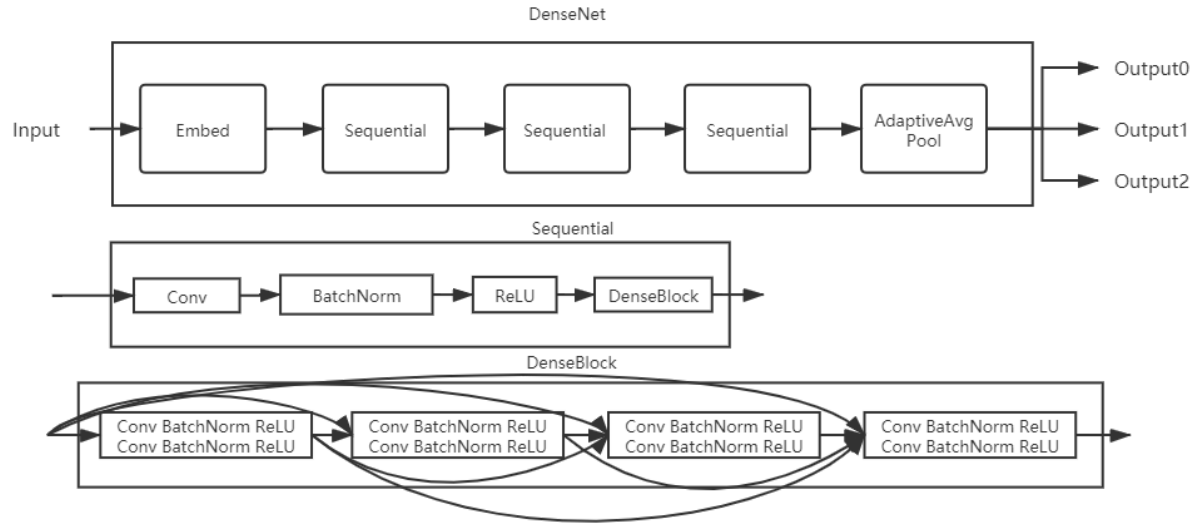


Fig. 4. DenseNet network structure. DenseNet passes the input of the current layer to each subsequent layer.

2.5 TransNet

TransNet[6] is taken from the Positional Encoding and Encoder of Transformer, and outputs the result from the Encoder. Figure 5 is the network structure of TransNet.

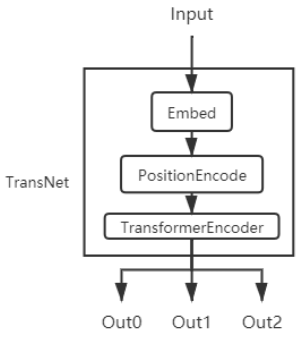


Fig. 5. TransNet network structure

2.6 Loss Function

We use the famous cross entropy function[7] as the loss function. In detail, we calculat the cross entropy of out0, out1, out2 and the corresponding class, fold, super and the rms value of the matrix before the output of the model, and multiply them by 1/3, 1/3, 1/3, 0.1. The loss function could be defined as:

$$Loss = \sum_{a=0}^{2} cross\_entropy(out_a, label_a) + \sqrt{\frac{1}{n}\sum_{x \in X} x^2}$$

where (out_a, $label_a$) corresponds to ($out0$, $class$), ($out1$, $fold$), ($out2$, $family$) and n is the number of elements of matrix X which is flattened to get $out0$, $out1$, $out2$.

3 Results

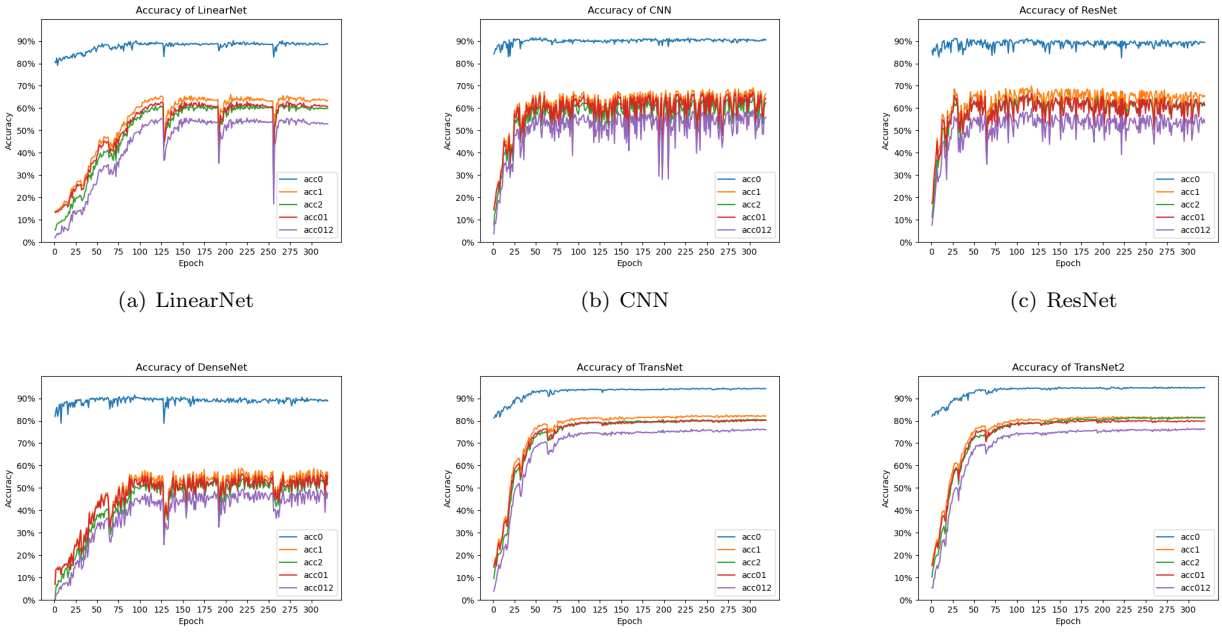
What we need is the prediction accuracy of class, fold, and super named acc0, acc1, acc2. Furthermore, the accuracy of predicting class, fold and the accuracy of predicting class, fold and super will be named acc01, acc012.

3.1 Accuracy

Figure 6 shows the curves of the accuracy of different network models during training. Table 1 shows the max accuracy of different network models during training. Figure 7 shows the curves of the acc012 of different network models during training. The difference between TransNet and TransNet2 is the length of positional encoding, which is 1000 and 800 respectively. We observe that the accuracy of TransNet and TransNet2 are higher than other models. Although we use the distance of contact as input, which reflects the global structure of the protein, the convolutional network does not effectively use the global structure because the convolution operation focuses more on local information. On the contrary, TransNet can effectively utilize the global structure of the protein by encoding the distance between contacts.

Model Name	Acc0	Acc1	Acc2	Acc01	Acc012
LinearNet	89.35%	65.5%	61.37%	62.92%	55.57%
CNN	90.42%	69.12%	64.47%	67.28%	59.15%
ResNet	90.61%	69.89%	65.25%	66.60%	59.83%
DenseNet	90.71%	62.92%	56.34%	61.47%	52.37%
TransNet	94.19%	**82.19%**	80.35%	80.25%	76.28%
TransNet	**94.87%**	81.10%	**81.22%**	**80.25%**	**76.57%**

Table 1. Max accuracy of models



(a) LinearNet (b) CNN (c) ResNet

(d) DenseNet (e) TransNet (f) TransNet2

Fig. 6. Accuracy of models

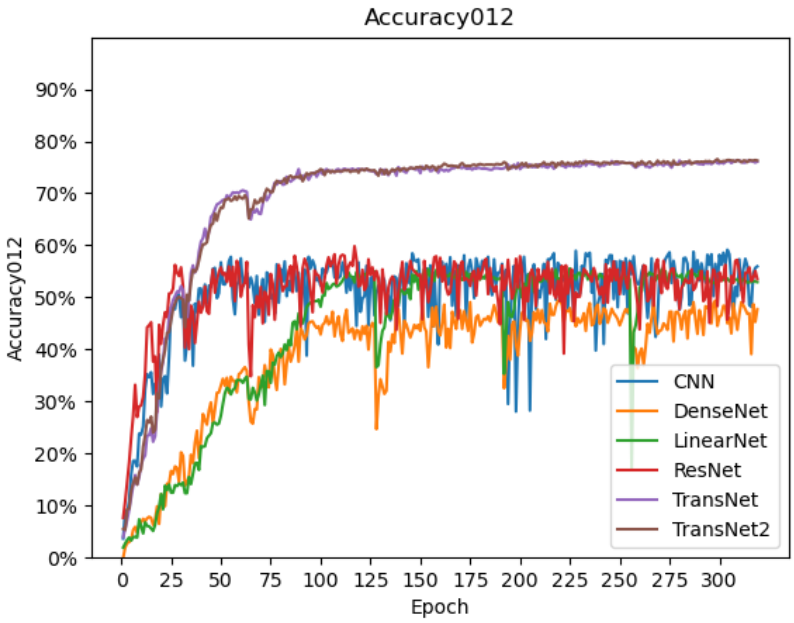
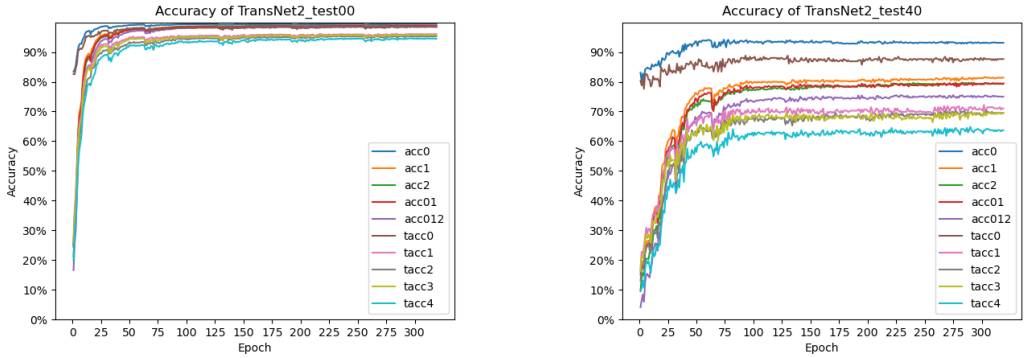


Fig. 7. Accuracy012 of models

Figure 8 shows the curves of accuracy of different sizes of data. Table 2 shows the max accuracy of different sizes of data. We have observed that the accuracy of the same model for different sizes of data is significantly different. The network model trained with 40% data achieved an accuracy of 76.57% on the validation set, but only achieved an accuracy of 64.86% on the testing set. The network model trained with 100% data achieves 98.49% accuracy on the validation set and 94.74% accuracy on the testing set, which is close to the efficiency on the validation set. The latter has a higher accuracy rate and is more efficient in predicting unknown proteins. This demonstrates the effect of different input data sets on the efficiency of the model is intense.

Model Name	Acc0	Acc1	Acc2	Acc01	Acc012
TransNet (100% data)	93.22%	98.92%	98.75%	98.78%	98.49%
TransNet (40% data)	93.22%	81.03%	79.38%	79.19%	75.02%
Model Name	Tacc0	Tacc1	Tacc2	Tacc01	Tacc012
TransNet (100% data)	98.83%	96.07%	95.50%	95.85%	94.74%
TransNet (40% data)	87.64%	71.43%	69.50%	69.50%	64.86%

Table 2. Max accuracy on different sizes of data.



(a) 00 (b) 40

Fig. 8. Accuracy on different sizes of data. The input on the left is 100% data, and the input on the right is 40% data. Acc0, acc1, acc2, etc. are the accuracy rates of the validation data set, and tacc0, tacc1, tacc2, etc. are the accuracy rates of the test data set.

Figure 9 shows the curve of the accuracy with family prediction. We observe that acc0123 is almost 0%. Although the distance matrix can reflect the structure of the protein, it is not enough to predict the family of the protein. This requires the sequence information of the protein, which is not used as the input of the model.

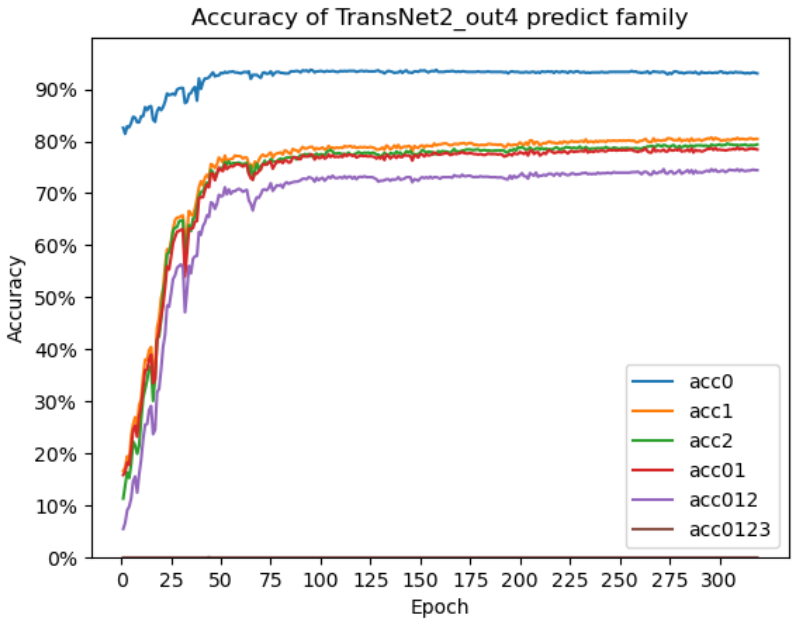


Fig. 9. We try to anticipate family through the use of TransNet, but the results are almost unavailable, the accuracy rate has been hovering around 0%.

3.2 Loss

Figure 10 shows the loss curves of different network models during training. The loss of all networks shows a cyclical upward and downward trend. The reason for this is the dynamic learning rate we use, which changes periodically named as CosineAnnealingWarmRestarts[8].

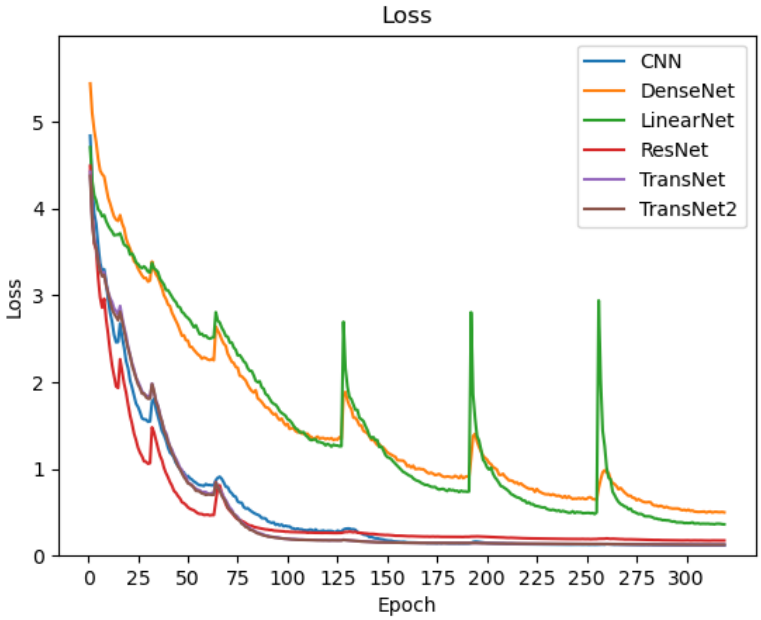


Fig. 10. Loss of models

4 Conclusion

Homologous proteins have a common ancestor, and their functions and evolutionary relationships are similar. For protein biology research, finding homologous proteins is often very critical. In order to find homologous proteins, we need to obtain the labels of class, fold, super, and family according to the structure of the protein. We divide the structure of the protein into multiple contacts based on the peptide bonds and hydrogen bonds ofthe protein, and extract the coordinates, dihedral angles and other information. We convert the coordinate information of contact into a distance matrix as the input of the model, and get three outputs of class, fold, and super. We try to use different neural network models which include LinearNet, CNN, ResNet, DenseNet, TransNet to classify proteins, and TransNet achieved the best efficiency. TransNet focuses on the global information of proteins, while convolutional networks focus on the local information of proteins, which is why TransNet is more efficient. Further, we try to use different size data sets for model training. The network model trained with 100% data achieves 98.49% accuracy on the validation set and 94.74% accuracy on the testing set, which is close to the efficiency on the validation set. This demonstrates that modifying the input data will also affect the efficiency of the model besides modifying the model

So far, we have only used the coordinate information of contact, but the protein also has other information such as dihedral angles. Combining dihedral angles and coordinates as input to the model will be something to try in the future.

References

[1] Wikipedia contributors. Structural classification of proteins database — Wikipedia, the free encyclopedia, 2020. [Online; accessed 27-December-2020].

[2] Y. Min, S. Liu, C. Lou, and X. Cui. Learning protein structural fingerprints under the label-free supervision of domain knowledge. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 69–74, 2018.

[3] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[4] 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 (CVPR)*, June 2016.

[5] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv e-prints*, page arXiv:2010.11929, October 2020.

[7] G.E. Nasr, E. Badr, and C. Joun. Cross entropy error function in neural networks: Forecasting gasoline demand. pages 381–384, 01 2002.

[8] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. *arXiv e-prints*, page arXiv:1608.03983, August 2016.