A Differentiable Protein Loop Detector

Alexander E. Chu[1,2]
[1]Biophysics Program, [2]Department of Bioengineering
Stanford University, Stanford, CA, USA

Introduction

De novo protein design is a new field in protein engineering and computational biophysics, in which the complete sequence and structure of protein molecules can be specified *in silico*. However, current methods are limited to fairly simple protein architectures, and better tools for protein design are needed to access new protein folds and allow for faster iteration in the design process. One common challenge is the ability to detect and improve poorly designed protein "loops." Here, I use 2D pairwise distance matrices to represent 3D protein structures, and then try to predict whether each residue of the protein is part of a loop. One advantage of this approach is that these matrices are similar to images, for which many convolutional tools have been developed.

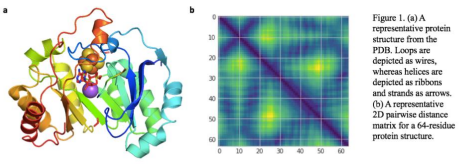

Figure 1. (a) A representative protein structure from the PDB. Loops are depicted as wires, whereas helices are depicted as ribbons and strands as arrows. (b) A representative 2D pairwise distance matrix for a 64-residue protein structure.

Experiments and Training

Two general approaches were tried: fully connected deep neural networks (FCNN), and deep convolutional neural networks (CNN). Three initial models (A, B, C) were trained using the fully-connected approach, and then a Final FCNN model was trained. An initial model was trained using the convolutional approach (D), and then a Final CNN model was trained.

Final Results

Training both models, I was able to achieve accuracies of 0.84 and 0.88 on the test set, which was not observed at any point during training, using the Final FCNN and Final CNN models, respectively. Interestingly, the performance of the models is consistent across the training, dev, and test sets, suggesting that there is perhaps some bias, but not much variance in these models. However, larger and more complex models did not perform better.

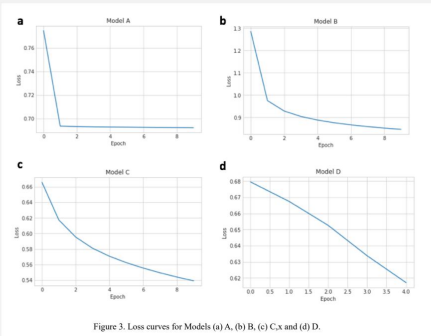

Figure 3. Loss curves for Models (a) A, (b) B, (c) C,x and (d) D.

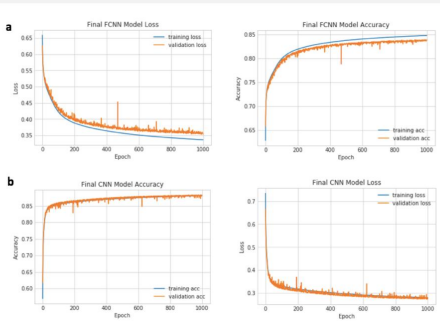

Figure 3. Loss and accuracy over training epochs for (a) the FCNN model and (b) the CNN model.

Table 1.	Architecture	No. of parameters	Epochs	Batch Normalization?	Training accuracy	Validation accuracy	Test accuracy
Model A	Dense: 256, 128, 64	1,089,984	10	No	0.5850	0.5835	-
Model B	Dense: 4096, 1024, 64	21,042,240	10	No	0.7100	0.7095	-
Model C	Dense: 256, 1024, 256, 64	1,590,848	10	No	0.7314	0.7292	-
Final FCNN	Dense: 256, 1024, 256, 64	1,590,848	1000	No	0.8412	0.8393	0.8405
Model D	7x7@32-Pool2, 4x4@64-Pool2, 4x4@128, 4x4@256-Pool2, 1x1@512-Pool2, 1x1@512-Pool2, Dense: 256,512,64, ReLU activations	1,773,376	5	No	0.6626	0.6679	-
Final CNN	4x4@8-s2, 4x4@16-s2, 4x4@32-s2, 4x4@64-s2, 1x1@128-s1, 1x1@128-s1, Dense: 256, 64, LeakyReLU[0.2] activations	610,344	1000	Yes	0.8805	0.8811	0.8822

Next Steps & Acknowledgments

In future work, I would like to explore a more thorough random hyperparameter search with more resources, and extend the function of these models to not only detect, but evaluate the quality of computationally designed protein loops.

I would like to acknowledge Raphael R. Eguchi and Ahmad Momeni for helpful guidance and discussions. Funding is acknowledged at bottom of the poster.