Review: Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model

Overview of Model

The authors create a residual net with two main components: first, a series of one dimensional convolutions of the one-hot encoded protein sequence, and second, a series of 2D convolutions on the features learned from the first network, along with coevolution information. An overview of the model is shown in Figure 1.

The input to the first part of the model consists of a $L \times 26$ one-hot encoded matrix of the protein sequence, where L is the length of the protein. Each of these L positions are convolved with some number of filters with dimension 1×17. Consequently, each 1-dimensional convolution produces a $L \times m$ activation matrix where m is the number of neurons per position (can be thought of as the number of different features). As in residual network literature, the $L \times m$ output of the 1d convolution is concatenated with the $L \times 26$ original input and fed into the following layer. This allows a mixture of lower and higher level features to rise through the feed-forward architecture. The authors fix the number of 1d convolutions somewhat arbitrarily at six.

This entire first module yields an $L \times n$ matrix, where n is a hyperparameter. This matrix, which consists of sequential information, is transformed to yield pairwise information as follows: for every position i and j in the sequence, the n features for position i are concatenated with the n features for position j and the n features for position $(i + j)/2$. This yields a matrix that is $L \times 3n$, and which is stacked with the external pairwise coevolution information. Importantly, we assume that that this concatenations step was done mainly to allow mixtures of long and short range features to be included in the second module of the network.

The second set of convolutional layers takes the pairwise features as input and convolves them with a series of two-dimensional kernels. Again, the output of the convolutional filters is periodically concatenated with the original data, and fed to the subsequent convolutional layer. This special characteristic of deep residual nets means that the layers are learning functions with reference to the original layer inputs, $f(X)+X$, rather than just $f(X)$. These deep residual nets have been empirically shown to be more computationally stable and thus easier to train in comparison to other deep networks [1].

Model performance and evaluation

We note that the authors demonstrate greater accuracy on predicting long range protein contacts rather than medium range contacts, which is both surprising and commendable. The authors additionally put test their predictions through several proof of concepts. In particular, the authors used their predicted contacts as constraints in *ab initio* folding programs, and

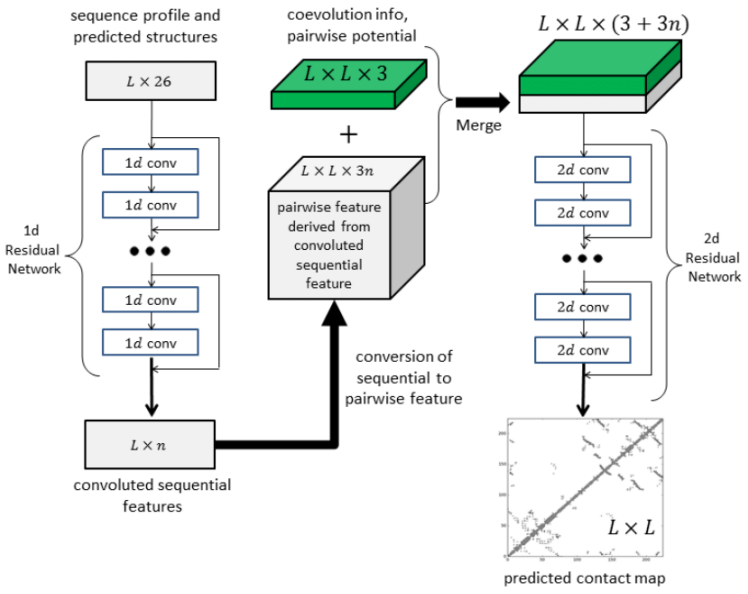

Figure 1: Overview of Residual Net Architecture

show twenty to thirty percent improvements over existing methods, in folding proteins from the widely-used CASP and CAMEO benchmarks.

In particular, the authors report stunning generalization ability, as they are accurately able to predict contact maps for membrane proteins with significant improvement over existing methods, even though the training set contained very few membrane proteins. The authors are to be commended for providing fairly unbiased data given that their training set included only 100 membrane proteins. Additionally, their experimental results suggest that the model may also work well in a transfer learning setting, where even for small datasets, initialization with the weights given by the residual net demonstrated by the authors may actually provide good performance in other closely-related domains.

Remarks

Residual Neural Networks. The author's usage of a residual net is commendable because of the apt application of convolutional neural networks that afford flexibility in providing both lower and higher level features to contribute to overall prediction performance. In particular, the authors are inspired by ResNet, a model built by Micorosoft Research that showed high accuracy in the ImageNet challenge. However we would have been interested in the application of *recurrent* architectures in this problem, either in conjunction or separate from the convolutional architecture. Recurrent networks allow varible input size, fixed hidden state size, and recent work on gated recurrent units (GRUs) allow both long range and short range information to emerge at prediction time. Thus, RNNs appear to be more natural architectures for the problem of learning informative features from protein sequences, and their greater flexibility in how the input is represented could allow for the network to learn more expressive features

for this problem.

Additionally, since the researchers are interested in building a platform that is usable for computational biologists, they should consider the computational complexity of a forward pass through their architecture. Indeed, recent theoretical work shows that very deep ResNets are essentially equivalent to shallow recurrent neural networks with weight sharing [2]. The benefit with RNNs is that since fewer parameters are required, it is more computationally efficient and potentially a more generalizable modeling regime.

Architectural improvements. In general, the authors underspecify their model design. The choice of architecture as presented in the paper is unmotivated and should be accompanied by further description of optimization procedures and hyperparameter search. What version of stochastic gradient descent was used to set the convolutional kernels? How many epochs were used to train the network? Was early stopping used in the training process? The choice of the depth in the architecture is also unclear, and should be explained. We would also like to know if grid search was used in hyperparameter tuning. In particular, this allows a much strong In addition, the choice of concatenation with pairwise features should be explained in terms of providing a mix of both short range and long-range information. We comment that their notation could be improved in general; in the figure they provide of the model, they do not make it clear that n in the $L \times n$ output from the 1d convolutional portion is a hyperparameter. They also do not provide any values for what they set n to be in the final model. Making the code open-source would solve many of these notational issues, and would make their model more clear and transparent for use in future research.

Regression formulation. Additionally, we would have been interested to see how the model would have performed had this problem been formulated as a regression over the distance map of residue-residue contact. While the classification problem is a canonical formulation of the problem and has associated state-of-the-art benchmarking available, it seems as though classifying would potentially cause information loss. We note, however, that including real valued distances would require a slightly modified 3D model design protocol that would jointly optimize over the energetics of a putative structure and the ℓ^2 deviation from the distance map output by the deep residual network. In addition, there might be some error when training the model on the distances from the crystallized protein, as the distances of amino-acids not in contact would tend to be more variable in solution versus in crystallized form, which could introduce some bias into the model for predicting structures which are similar to those of crystallized proteins. How much the *ab initio* folding programs should be constrained by the predictions from such a network would be interesting to investigate.

References

[1] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *CoRR* abs/1512.03385 (2015). URL: http://arxiv.org/abs/1512.03385.

[2] Qianli Liao and Tomaso A. Poggio. “Bridging the Gaps Between Residual Learning, Recurrent Neural Networks and Visual Cortex”. In: *CoRR* abs/1604.03640 (2016). URL: http://arxiv.org/abs/1604.03640.