BS6207 Project:

Molecular Recognition Between Proteins and Ligands with Deep Learning

Jiao Yan, May 21

1. Problem definition

Molecular recognition between proteins and ligands plays an important role in many biological processes, such as membrane receptor signaling and enzyme catalysis. Predicting the structures of protein-ligand complexes and finding ligands by virtual screening of small molecule databases are two long-standing goals in molecular biophysics and medicinal chemistry [1, 2].

Deep learning has been relatively widely used by the bioinformatics and computational biology communities. Deep learning can be used for structure-based methods to predict protein-ligand binding affinity, which can play a role in the early stages of drug discovery.[3]

Given 3000 protein-ligand complexes with known 3D structures containing (x, y, z) coordinates and atom type ('C - Carbon', 'O – Oxygen', 'N – Nitrogen', etc)of the proteins and ligands respectively, the problem is to train a neural network that takes in the (x, y, z) coordinates and atom type of a pair of protein and the ligand, and then predict if they bind at the output of the network.

For testing each protein in the test data set, 10 ligands should be predicted to be the candidates that bind the protein. For each protein, only one ligand will bind to it. Hope that among the 10 candidates, you identify the correct one. Final score of the project is the number of proteins with a correct prediction for binding.

2. Highlights (new algorithms, insights from the experiments)

The most tough part in this deep learning classification part is the data preprocessing part mentioned in part 3. The difficulty we are facing is to convert the randomly shaped input into uniform structure, namely here, 3×48×48×48.

The following are the network structure:

Layers	Parameters	Shape
Inputs		3,48,48,48
Layer1	nn.Conv3d(3, 32, kernel_size=5, stride=1, padding=0)	32,22,22,22
Layer2	nn.Conv3d(32, 64, kernel_size=5, stride=1, padding=0)	64,9,9,9
Layer3	nn.Conv3d(128, 256, kernel_size=5, stride=1, padding=0)	128,9,9,9
Layer4	nn.Conv3d(64, 128, kernel_size=1, stride=1, padding=0)	256,2,2,2
Reshape	Dimension reduction into 1-dimension	2048
Fully connected Layer 1	nn.Linear(2048, 100) with dropout = 0.5	20,100
Fully connected Layer 2	nn.Linear(100, 1)	20,1
Sigmoid	Binary Classification	
Output		20,1

The input voxel is followed by 4 consecutive convolutional layers with ReLu activation functions. The output from the 4th convolutional layer, 256 feature maps of size 2,2,2, are passed through 2 the fully connected layer. The fully connected layers add up the weights of the previous layers to determine the exact blend of features to achieve a specific target output.

Since our input voxel size of 3×48×48×48 is significantly larger than the typical image data used in computer vision, such a deep network architecture would be computationally demanding, requiring GPU computation power.

2 models were compared in section 5, model 1 is as above, the other model 2 is all the same but without layer 4.

3. Dataset pre-processing description

The three-dimensional structure of the protein-ligand complex requires specific conversion and coding to be used by neural networks.

The input is represented as a 4D tensor, where each point consists of (x, y, z) coordinates (the first 3 dimensions of the tensor) and an atom type vector (the last dimension, converted from several atom types: B, C, N, O, P, S, Se, halogen and metal to 0 /1, 0 means polar, 1 means hydrophobic) definition.

The protein and ligand are labelled with 4-digit number as (0001_pro_cg_.pdb) ligand (0011_lig_cg_.pdb). If the number is equal, then the pro and ligand make a pair, if not, then not a pair.

Then we create 3000*3000 (9,000,000) pairs of data consisting of:

	Definition	Number of pairs	Label
Positive set	the pair of protein and ligand with same number code	3,000	[1,0]
Negative set	the pair of protein and ligand with different number code	8,997,000	[0,1]

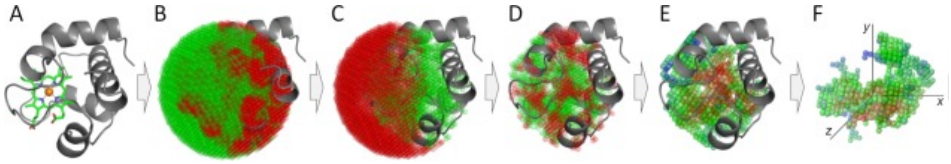
Since positive / negative datasets are very unbalanced, they are randomly shuffled and split into training and validation set with ratio=1:9:

	Positive	Negative	Total
Training set	2,700	8,097,300	8,100,000
Validation Set	300		900,000

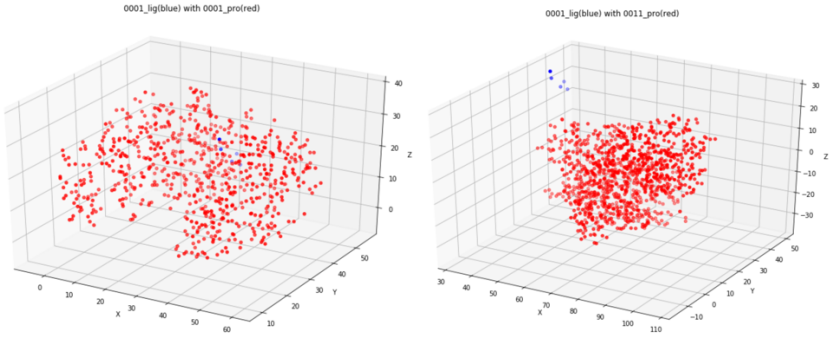
Testing data contains 824 protein and 824 ligand files that are named from 0001 to 0824; however, same 4-difits codes do not imply the protein and ligand are binding. Similarly paired up as training dataset does, we have 678976 data in testing set.

The data are still in different shape, with 4 times different width, however, deep learning requires ligand-protein complexes to be converted to fixed-size 3D voxel structures. A three-step refinement procedure is then applied in order to obtain a precise representation of the actual pocket by removing unnecessary grid points.

First, excluded volume points, defined as those located within 2 Å from any protein atom, are eliminated (red points in Fig B). Second, points outside of the convex hull of the protein structure, which is considered as the union of envelopes of all protein atoms, are removed (red points in Fig C). Third, discarding points detached from the largest connected component in the pocket grid (red points in Fig D) creates the final, continuous grid structure in the shape of [3,48,48,48].[4]



The examples of a PL complex and non-paired PL are shown below:



4. Training and testing procedure

In the training procedure, the parameters are batch_size = 20, num_epochs = 20, lr = 0.001.

The loss function is MSELoss and the optimizer is Adam optimizer that outperform stochastic gradient descending by avoiding local minimum with adaptive learning rate. The code below shows the process of training and evaluate the performance on validation set with accuracy.

```
correct = 0
total = 0
train_loss = []
for step, (inp,target) in enumerate(train_loader):
    target = target.float()
    inp, target = inp.to(device), target.to(device)
    outputs = model(inp)
    loss = criterion(outputs, target)
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
    total += target.size(0)
    print(step, loss)
    correct += (outputs.ge(0.5).reshape(20, 1) == target.reshape(20, 1)).sum().item()
    acc = correct / total
```

In the testing procedure, the parameters are batch_size = 1, log_interval = 5000. The testing result is the sigmoid probability of 824*824 pairs, then select the top 10 probability ligands as the candidates.

5. Experimental study

The experiment results of 2 models are shown as below:
The loss and accuracy of model 1 in 20 epochs:
First, the extra layer of layer 4 was supposed to increases the amount of model parameters and better fits the feature distribution of the data. Secondly, it can better extract the features in the data, making the model more robust.

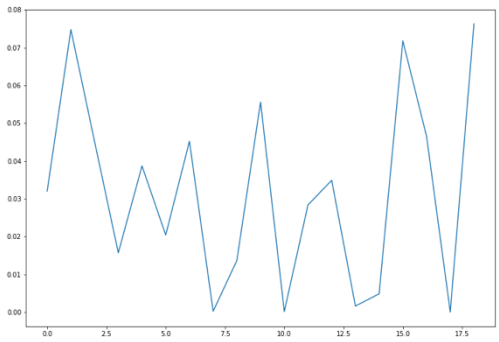


Figure 5.1: Loss of model 1

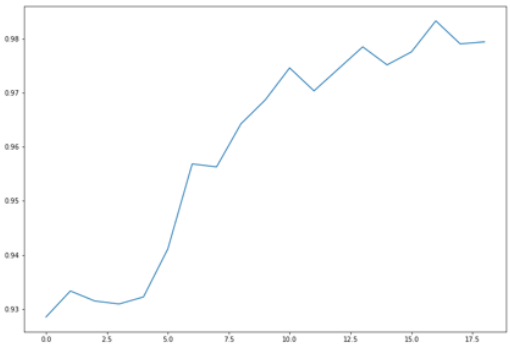


Figure 5.2: Accuracy of model 1

The loss and accuracy of model 2 that without the 4th conv3d layer in 20 epochs:

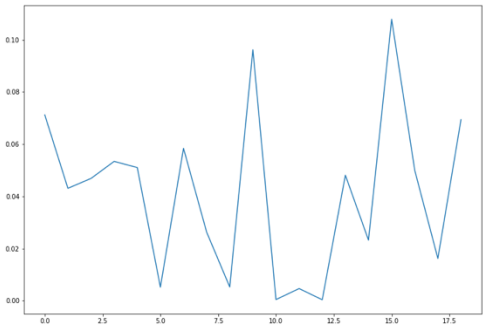


Figure 5.3: Loss of model 2

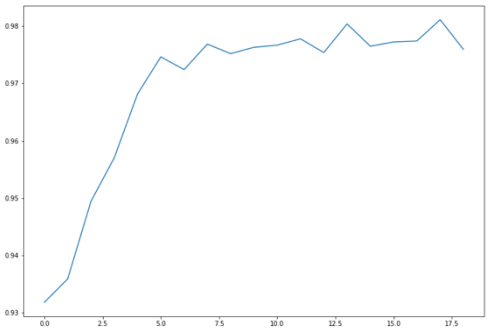


Figure 5.4: Accuracy of model 2

According to the results, the loss of 2 models are very small at the beginning, about 0.7-0.8, and there is no sign of convergence after training many times. This is generally a problem with the data set or unreasonable network settings. If there is a problem with the network settings, you need to change the model, such as changing the loss parameter type.

Secondly, both accuracies are very high, beginning from 93% to 98%. However, the accuracy of model 2 increases faster than model 1, This is different from the initial guess, that more layer more robust. On the contrary, too complex model could be a overkill that overfitting the training data.

Reference:

1) https://en.wikipedia.org/wiki/Protein
2) https://en.wikipedia.org/wiki/Protein_structure
3) Stepniewska-Dziubinska, Marta M., Piotr Zielenkiewicz, and Pawel Siedlecki. "Development and evaluation of a deep learning model for protein–ligand binding affinity prediction." *Bioinformatics* 34.21 (2018): 3666-3674.
4) Pu L, Govindaraj RG, Lemoine JM, Wu HC, Brylinski M. DeepDrug3D: Classification of ligand-binding pockets in proteins with a convolutional neural network. PLoS Comput Biol. 2019;15(2):e1006718. Published 2019 Feb 4. doi:10.1371/journal.pcbi.1006718