Final Project

BS6207

ZHOU RUIMENG

Problem Definition

This project aims to predict the structures of protein-ligand complexes and find ligands by virtual screening of small molecule databases. The training and testing dataset has been given and we need to find out which two files can be bound together. The result is evaluated by picking out the top 10 potential ligands and if the correct one is in these 10 ligands, the prediction is correct.

Platform: Jupyter Notebook
CPU: Intel(R) Core (TM) i7-10750H
RAM: 24G
Language: Python 3.7
Framework: pyTorch

Highlights

Self-constructed the model and new ways to process the data instead of using built-in libraries.

Data Analysis and Pre-processing:

In the dataset there are several features, I draw the 3d scatter plot to see the distribution of each X, Y, and Z. I find that most of the values are combined. It makes sense because since it is a complex the structure would not be sparse.

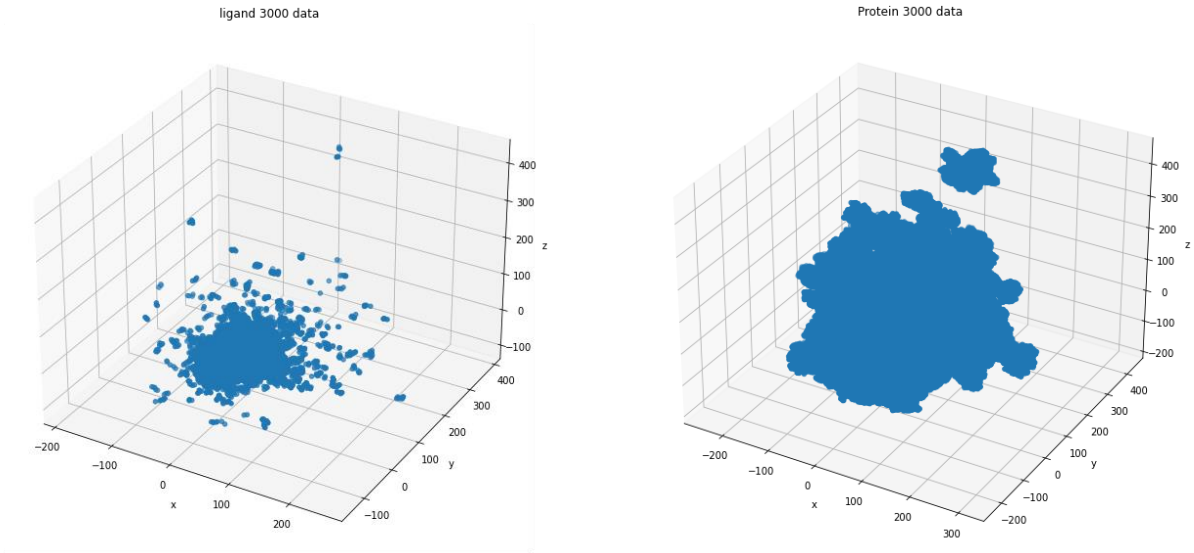


Figure 1 Distribution of coordinate

After I read the protein and ligands data, I store them into a list. There are 3000 proteins and ligands in total. From above (**Figure 1**) the ligand data is sparser than proteins, so I calculate the centroid of each ligand based on X, Y, Z, and minus by protein which can make the distance between protein and ligand much closer. Also, I minus the ligands as well so that the ligands can be closer as well. I split 80% of the data as training data and 20% as validation data.

Then I set two lists to store all the normalized training part and validation part. And here is the description of the training data. The mean value of x, y, z is quite close which is 39.99, 40.37, and 42.85 respectively. From DataFrame we can find the mean value close to 50%.

	max_X	max_Y	max_Z	min_X	min_Y	min_Z
count	3000.000000	3000.000000	3000.000000	3000.000000	3000.000000	3000.000000
mean	39.992332	40.369756	42.849595	0.030064	0.025421	0.031162
std	18.789359	19.764227	21.467782	0.102038	0.030708	0.060157
min	8.642000	10.116000	10.380000	0.000000	0.000000	0.000000
25%	27.066750	27.489750	28.384250	0.006000	0.006000	0.007000
50%	34.487500	34.577000	36.072000	0.016000	0.015000	0.017000
75%	47.493000	47.796250	50.304500	0.035000	0.033000	0.037000
max	282.240000	261.039000	271.938000	4.103000	0.336000	1.463000

Table 1. Description of Normalized Data

Here is the distribution plot, most of the protein sequence lengths are around 800 to 1500 and most of the ligand length is from 4 to 5. The length between protein and ligand is so different **(Figure 2)**, it will be hard for Convolutional Neural Network to pass the constant number through channels. Meanwhile, due to the lack of CPU performance, data need to be pruned. I divided the coordinate by 4 and the shift of pairing size. Since I set the 3D structure as 25*25*25, I pruned the coordinates if they were equal or larger than the pairing size.

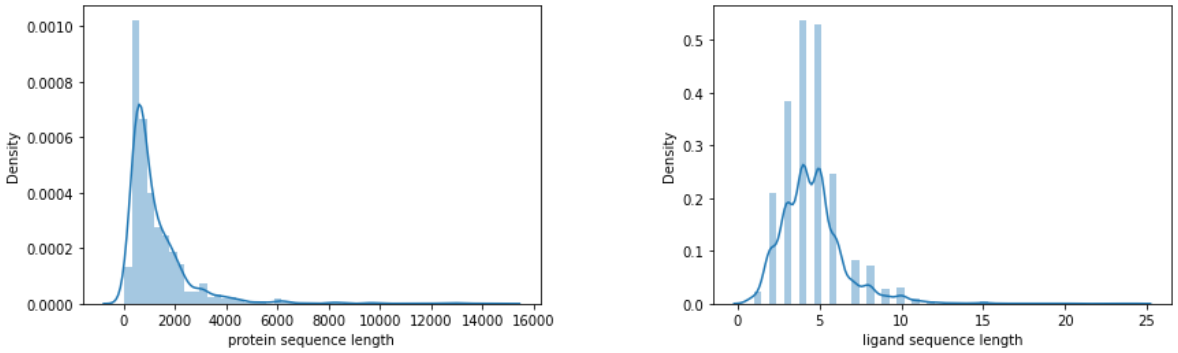
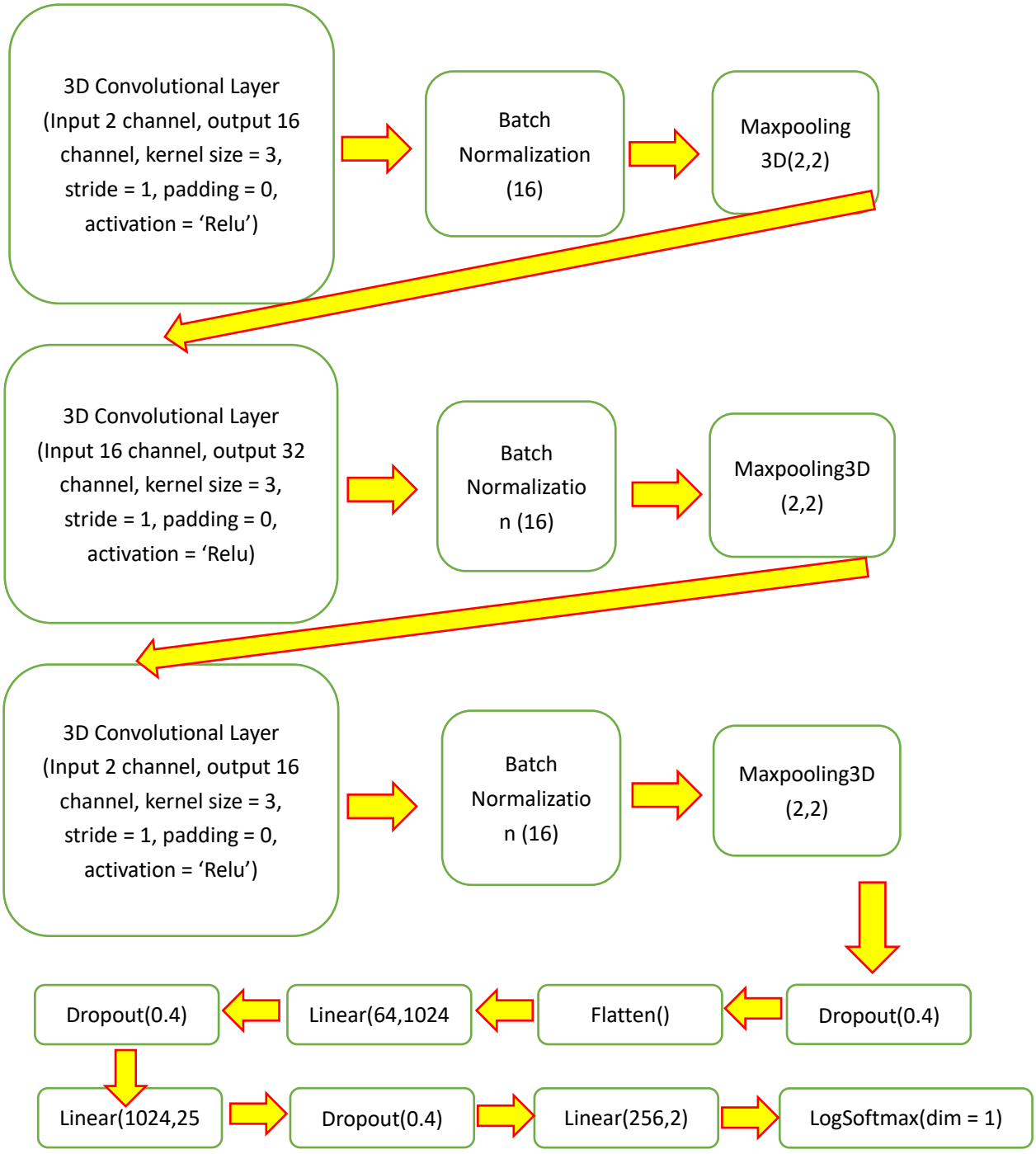


Figure 2. Distribution of Sequence Length

Instead of pairing ordered data, I also pair the disorder data used for validation. In generate_disorder_pair function, I randomly choose the ligands to pair with the protein so that I can validate the performance of the model.

Model Structure

The model structure is shown below. The data will first be passed into a 3D Convolutional Layer, with kernel equals 3 and stride equals 1. Followed by Relu activation, a 3D Batch Normalization, and a 3D Maxpooling layer. After that, the neural network will dropout 40% and the data will be flattened into one dimension. Then the data is fed into the fully connected layer and dropout the network again. Finally applies the LogSoftmax function to a 1-dimensional input Tensor.



Training Part

In our training process, I split the whole dataset as training and validation data by a ratio of 8:2. The batch size I set here is 64 and I use the SGD as my optimizer and the learning rate is set to 0.001.

The total epoch is 300 and I save the model once the accuracy is higher than 80% or 90%. I split training into several parts. Also, I print the accuracy and loss for the validation set during training.

The plot below illustrates the training and validation loss through the last few epochs. The training loss decreases steadily before the last 40 epochs and then decreases acutely. Finally, it maintains around 0.1

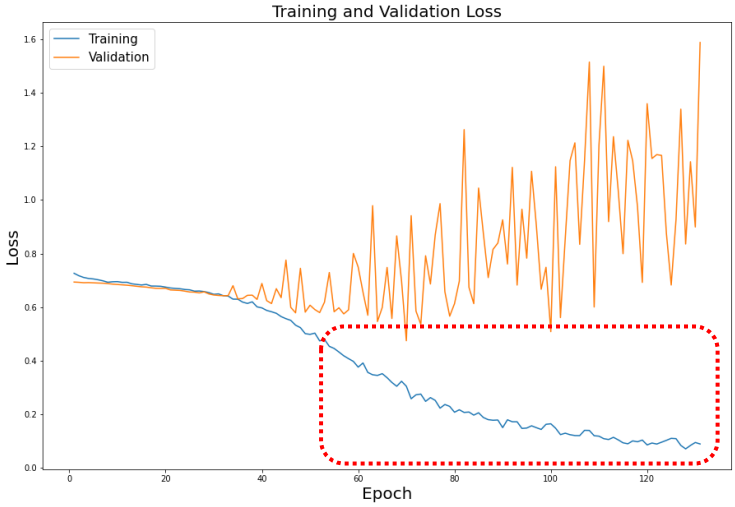


Figure 3. Huge Decrease

I then further plot the accuracy of each epoch during validations. It clearly shows that before the last 40 epochs, the accuracy increased to a stable status but later it showed a huge fluctuation. Finally, the accuracy reaches 85%.

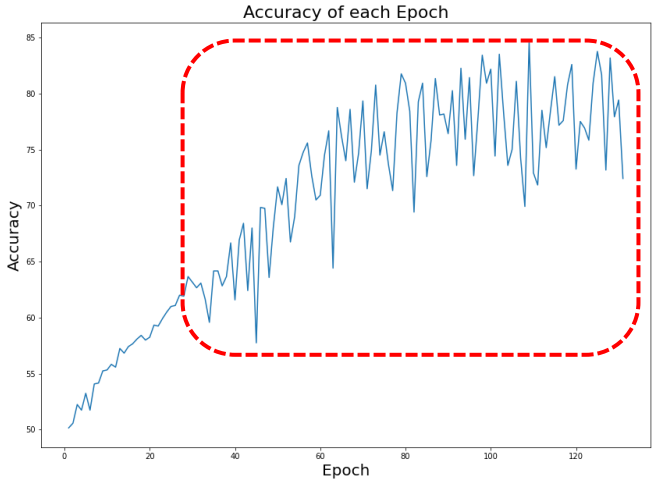


Figure 4. Huge Fluctuation

Experimental study & Conclusion & Future Work

So far I finish a project to predict the combination of protein and ligands. I first read the pdb file and normalize the data by calculating the distance between protein and ligands. Then I pair the data in order and in disorder, which the order pairing means these two data can be bound and the disorder pairing means these two data cannot be bound. Then I construct and train a Convolutional Neural Network on the training dataset and validation dataset. The performance is good and accuracy can reach 85%. Then I run the test dataset and output the top 10 potential results.

In the future, I think the performance can be improved a lot. The model can be re-constructed and data can be used in different ways to process the data. Perhaps the data augmentation can be done as well. Last but not least, the huge fluctuation makes the model not that stable. The reason may be owing to the layers in the model.