BS6207 Final Project LONG JINGYU 03/02/2022

Predicting Protein – Ligand Interaction by using Deep Learning Models

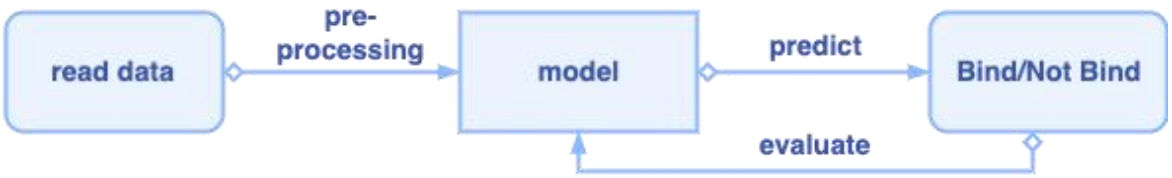
Problem definition

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:

(1) Train a neural network that takes in as input, the (x, y, z) coordinates of atoms and atom type of a pair of protein and the ligand, and then predict if they bind at the output of the network.

(2) For the metric of grading, for each protein in the test data set, you are required to identify 10 ligands that are predicted to bind the protein. For each protein, only one ligand will bind to it. Final score of the project is the number of proteins with a correct prediction for binding.

(3) For your project output, you need to provide .csv file specifying which protein random index files match to which of ligand random index files.



Highlights

1.AdaBound Optimizer: As fast as Adam, as good as SGD optimizer

Disadvantages of SGD: SGD is often used in late tuning, but the problem with SGD is that the convergence speed is slow in the early stage. The reason for the slow convergence of SGD in the early stage is that SGD has the same scale and contraction of gradients in all dimensions when updating parameters, and the training effect is very poor when the training data distribution is very uneven. Due to the problem of slow convergence, adaptive optimization algorithms such as Adam. Although these adaptive optimization algorithms can show fast convergence speed in the early training, their performance in the test set will soon stagger and eventually be surpassed by SGD.

Disadvantages of Adam : Adam in the late period, which is caused by the unstable extreme learning rate in the late period of adaptive method training. In other words, when the self-adaptive learning rate is trained in the later stage, the learning rate is extreme. When the parameters are updated, the learning rate in some dimensions is particularly large, while that in some dimensions is particularly small.The learning rate of sampling parameters. Each cell contains a value obtained through numerical calculation of the learning rate. The lighter the color, the smaller the learning rate. We can see that when the model approaches convergence, there are a large number of extreme values (including many cases less than 0.01 and greater than 1000) in the learning rate. This phenomenon shows that extreme learning rate actually exists in practical training. The

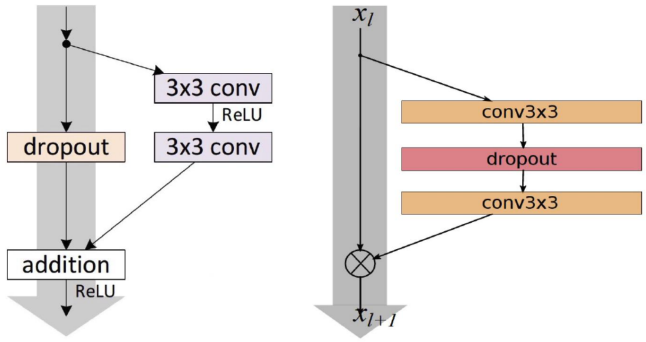
specific approach is to dynamically cut the learning rate,

Under this setting, the algorithm is closer to Adam in the early training period because the upper and lower bounds have little influence on the learning rate. As time goes by, the clipping range becomes tighter and tighter, and the learning rate of the model gradually tends to be stable and closer to SGD at the end of the model. In other words, Adam and SGD are a special case of AdaBound.

2. WRNs — Wide Residual Networks

By widening Residual Network (ResNet), the network can be shallower with the same accuracy or improved accuracy. Shallower network means:

- Number of layers can be reduced.
- Training time can be shorter as well.

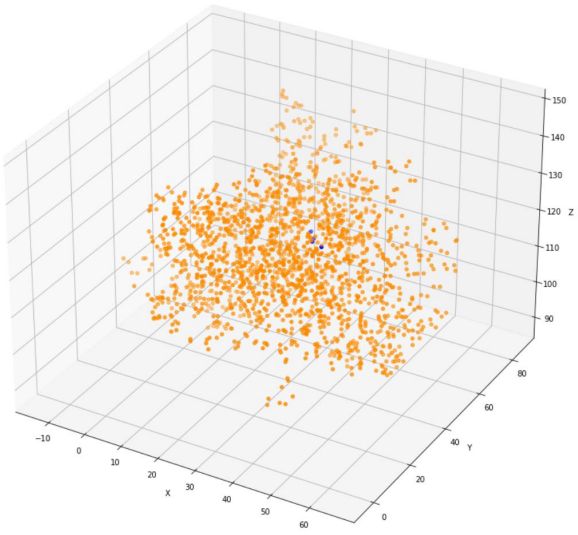


Dropout in Original ResNet (Left) and Dropout in WRNs (Right)

Dataset pre-processing description

We can see protein contain 100s to 1000s of atoms but ligand just contain much less atoms,which is hard to design a neural network, so we have to build a 3D matrix.

0004_pro_cg.pdb, 0004_lig_cg.pdb



(1) Return a sparse matrix representation of the voxel. After merge protein and ligand, we center all atoms around the center of the protein, we will add feature list to

identify the atom h/p and pro/lig,move all atoms to the nearest grid point
(2) **Return a tuple containg the training data and corresponding labels** specifies the number of negative and positive training examples .

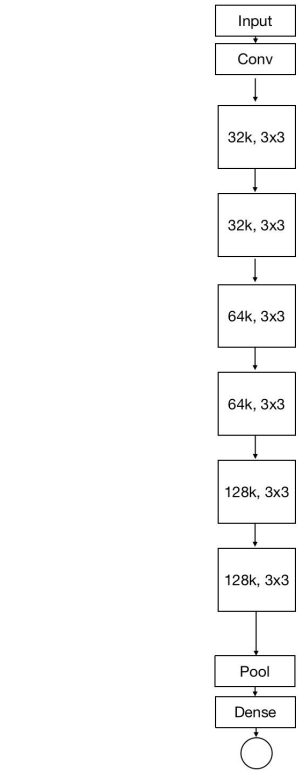
(3) Training and testing procedure

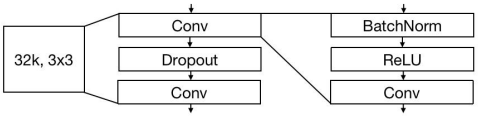
The constructed model structure is as shown in the below diagram, which add 5 widening Residual Network (ResNet):

Wide ResNet Model

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 21, 21, 21,	0	
conv3d (Conv3D)	(None, 6, 6, 6, 16)	6928	input_1[0][0]
batch_normalization (BatchNorma	(None, 6, 6, 6, 16)	64	conv3d[0][0]
activation (Activation)	(None, 6, 6, 6, 16)	0	batch_normalization[0][0]
conv3d_2 (Conv3D)	(None, 6, 6, 6, 32)	13856	activation[0][0]
dropout (Dropout)	(None, 6, 6, 6, 32)	0	conv3d_2[0][0]
batch_normalization_1 (BatchNor	(None, 6, 6, 6, 32)	128	dropout[0][0]
activation_1 (Activation)	(None, 6, 6, 6, 32)	0	batch_normalization_1[0][0]
conv3d_3 (Conv3D)	(None, 6, 6, 6, 32)	27680	activation_1[0][0]
conv3d_1 (Conv3D)	(None, 6, 6, 6, 32)	544	activation[0][0]
add (Add)	(None, 6, 6, 6, 32)	0	conv3d_3[0][0] conv3d_1[0][0]
batch_normalization_2 (BatchNor	(None, 6, 6, 6, 32)	128	add[0][0]
activation_2 (Activation)	(None, 6, 6, 6, 32)	0	batch_normalization_2[0][0]
conv3d_4 (Conv3D)	(None, 6, 6, 6, 32)	27680	activation_2[0][0]
dropout_1 (Dropout)	(None, 6, 6, 6, 32)	0	conv3d_4[0][0]
batch_normalization_3 (BatchNor	(None, 6, 6, 6, 32)	128	dropout_1[0][0]
activation_3 (Activation)	(None, 6, 6, 6, 32)	0	batch_normalization_3[0][0]
conv3d_5 (Conv3D)	(None, 6, 6, 6, 32)	27680	activation_3[0][0]
add_5 (Add)	(None, 6, 6, 6, 128)	0	conv3d_15[0][0] add_4[0][0]
batch_normalization_12 (BatchNo	(None, 6, 6, 6, 128)	512	add_5[0][0]
activation_12 (Activation)	(None, 6, 6, 6, 128)	0	batch_normalization_12[0][0]
average_pooling3d (AveragePooli	(None, 3, 3, 3, 128)	0	activation_12[0][0]
flatten (Flatten)	(None, 3456)	0	average_pooling3d[0][0]
dense (Dense)	(None, 128)	442496	flatten[0][0]
dropout_6 (Dropout)	(None, 128)	0	dense[0][0]
dense_1 (Dense)	(None, 1)	129	dropout_6[0][0]

Total params: 2,497,201
Trainable params: 2,495,377
Non-trainable params: 1,824



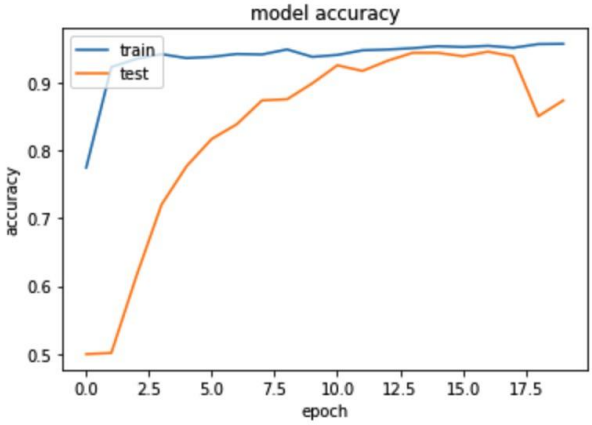


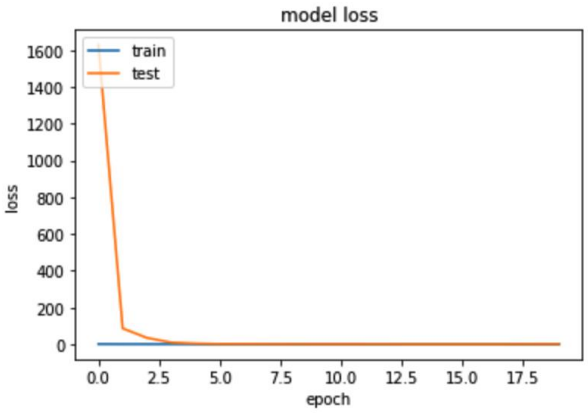
Training data are randomly shuffled and split into training and validation set with 1:9, and i rent a cloud GPU in website due to the performance. In the training procedure, the parameters are batch_size = 512, num_epochs = 30, optimizer=AdaBound(lr=1e-3, final_lr=0.1), loss='binary_crossentropy'.

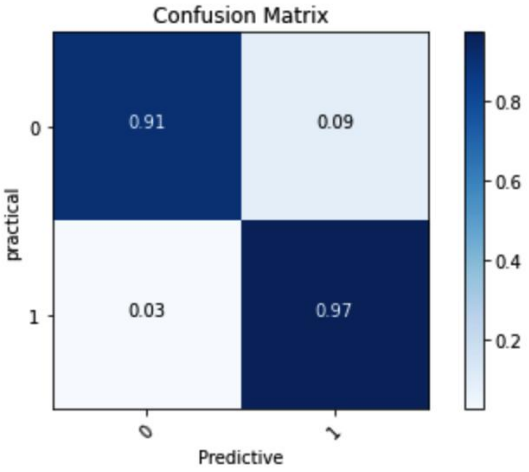
Experimental study

I trained two models by Adam and Adaboost. According to the results, the loss of 2 models are very small at the beginning between 75%-80% , both accuracies are over 90%. However, the accuracy of model 2 increases faster than model 1, the fitting effect is better.

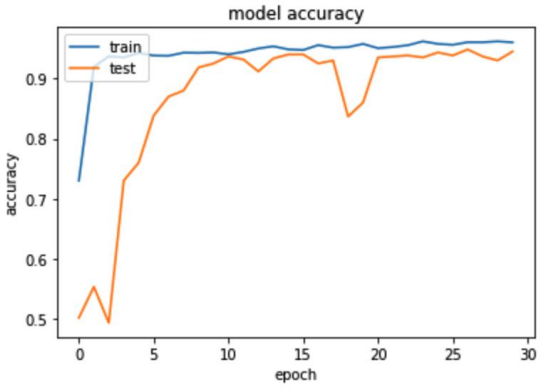
Adam optimizer result

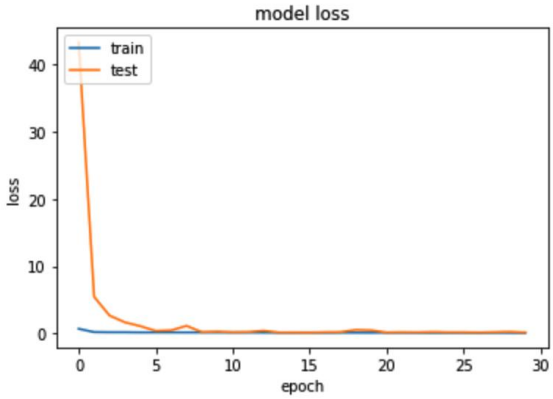


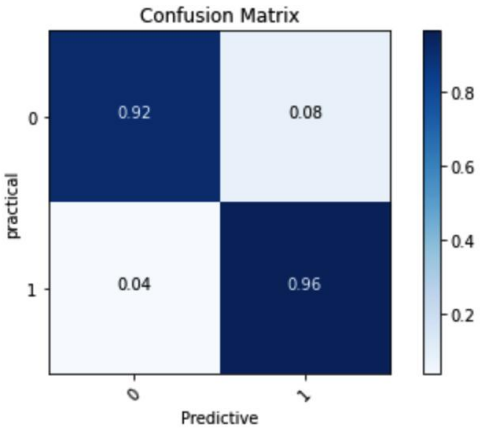




Adaboost optimizer result







References

1) https://en.wikipedia.org/wiki/Protein
2) https://en.wikipedia.org/wiki/Protein_structure
3) https://en.wikipedia.org/wiki/Drug_design
4) ftp://ftp.wwpdb.org/pub/pdb/doc/format_descriptions/Format_v33_A4.pdf
5) https://pymol.org/2/#download
6) http://pymol.sourceforge.net/newman/userman.pdf
7) https://chemrxiv.org/articles/DLSCORE_A_Deep_Learning_Model_for_Predicting_Protein-Ligand_Binding_Affinities/6159143
8) https://academic.oup.com/bioinformatics/article/34/21/3666/4994792
9) https://arxiv.org/abs/1612.02751