Project 1: Drug Molecular Toxicity Prediction

Yuxiang Lu
518021911194
luyuxiang_2018@sjtu.edu.cn

Abstract

Deep learning can be used to predict molecule toxicity in the discovery and trial of new drugs. In this project, I implement a convolution neural network (CNN) to learn local biochemical features from the one-hot SMILES representation, and train it to predict the drug toxicity. The final result on the test is 0.86727 AUC.

1 Introduction

When we are sick, we often turn to doctors and ask for some medicines. Since everyone is bound to have different kinds of drugs during his or her lifetime, it is critical to ensure that the chemicals and substances used in the drugs will never be toxic or harmful to human beings. Therefore, it may take a long period of time to do multiple rounds of human clinical trials before a drug discovered in laboratory can be widely used by patients. Besides the time-consuming of clinical trails, it may also affect the health of the subjects. Research points out that more than 30% of drug candidates failing in clinical trials because of the undetected toxic effects [1].

Due to the problems in real-world clinical trials, drug toxicity assessment method is demanded to quickly and simply predict drug toxicity according to its molecular. Deep neural networks is the hottest topic in Artificial Intelligence in recent years, deep learning performs excellently in fields like natural language processing (NLP), computer vision (CV), since it is expert in learning and recognizing features automatically from large amounts of data. Many recent researches have shown that it is possible to predict drug toxicity using deep neural networks (DNN), including convolutional neural network (CNN), recurrent neural network (RNN), long short-term memory (LSTM), graph convolutional network (GCN), etc. Some of these researches will be talked about in the next section.

2 Related Work

The common method to test the toxicity of chemicals depend on High-Throughput Screening (HTS). HTS experiments can investigate whether a chemical at a given concentration exhibits a certain type of toxicity. These experiments are repeated with varying concentrations of the chemical, which allows to reliably determine whether a compound activated a given pathway or receptor, inhibited it or did not interact at all[1]. With the HTS experiments, many datasets of chemical toxicity are generated, such as "Toxicology in the 21st Century" (Tox21). However, these experiments take a lot of time and costs. Tox21 project costs millions of dollar, but could test only a few thousands of compounds for as few as twelve toxicity. Therefore, computational methods are invented for accurate prediction of chemical toxicity.

Common computational methods can be divided into two classes, structured-base and ligand-based. The structured-based ones simulate physical interactions between the compound and a biomolecular target but are limited to completed known 3D structure of all interacting molecules[3]. Ligand-based approaches use previous measurements to predict the molecular interactions[4].

Before deep learning appears in recent years, traditional machine learning methods in toxicity predictions are mostly based on ligand features, such as scoring approaches like Naive Bayes statistics[5], density estimation[6], nearest neighbour, support vector machines (SVM), and shallow forward neural networks[7].

Then deep learning becomes a hot topic in artificial intelligence, in 2012, Dahl et al.[8] won the Merck Kaggle Challenge using deep neural networks, which showed that multi-mask learning can help to predict biochemical activities on single proteins. Therefore, many researchers were inspired to use deep learning in toxicity and bioactivity prediction.

Mayr A et al.[10] proposed a deep neural network (DNN) model named DeepTox to predict the compound toxicity. DeepTox first normalizes the chemical representations of the compounds, then computes a large number of chemical descriptors as the input to DNN. Unterthiner et al.[12] also used deep neural network to automatically learn features resembling well-established toxicophores and predict several different types of toxic effects at the same time. Differently, they used a high-dimensional binary representation produced by Extended Connectivity FingerPrint (ECFP4) features, which is a common compound description in drug design applications.

Ryu et al.[11] proposed an attention- and gate-augmented graph convolutional network (GCN) for the prediction of molecular structure-property. The attention mechanism helps to identify atoms in different environments, and the gated skip-connection further improves by updating feature maps at an appropriate rate. They demonstrated that their model could extract better structural features related to a target molecular property such as solubility, polarity, synthetic accessibility and photovoltaic efficiency.

Wallach et al.[2] introduced AtomNet, a structured-based, deep convolutional neural network (DCNN) designed to predict the bioactivity of small molecules. Their model expertises in apply the covolutional concepts of feature locality and hierarchical composition to the modeling of bioactivity and chemical interactions. Chen et al.[9] proposed a noval convolutional neural network (CNN) regression model named BESTox, to predict the acute oral toxicity of chemical compounds. Their model learns the compositional and chemical properties of compounds from their two-dimensional binary matrices. Each matrix encodes the occurrences of certain atom types, number of bonded hydrogens, atom charge, valence, ring, degree, aromaticity, chirality, and hybridization from the SMILES string of a given chemical.

3 Dataset

The dataset used in this project is about the toxicity of some small molecules. It includes three parts, 8169 samples for training, 272 samples for validation, and 610 samples to test the model. There are three files for the training and validation data:

1. *names_smiles.txt*, each line contains a drug molecule's name and its SMILES expression;
2. *names_labels.txt*, each line contains a drug molecule's name and its toxicity, where 0 means non-toxic and 1 means toxic;
3. *names_onehots.npy*, a numpy file, contains two ndarrray, one is the names of the molecules, and the other is the one-hot representations of SMILES expressions of the drug molecules.

The test data only have *names_smiles.txt* and *names_onehots.npy*.

Simplified Molecular-Input Line-Entry System (SMILES) is a linear representation for molecular structure using 1D ASCII strings. The one-hot format of SMILES is a 2D binary matrix, where each column represents a symbol in the SMILES notation of the molecule, like the atoms, chemical bonds, while each row is one ASCII character appeared in the whole SMILES dictionary, and does not have repeated contents. If the content of a column matches that of a row, the cell will be one, otherwise it will be zero, which means that there is one and only one 'one' in each column.

For example, aspirin, a common drug, its SMILES is

$$\mathrm{CC(=O)OC1=CC=CC=C1C(=O)O}$$

and its one-hot matrix is

	C	C	(	=	O	)	O	C	1	=	C	C	=	C	C	=	C	1	C	(	=	O	)	O
C	■	■						■			■	■		■	■		■		■					
(			■																	■				
=				■						■			■			■					■			
O					■		■															■		■
)						■																	■	
1									■								■							

The size of each one-hot SMILES matrix is the length of the SMILES dictionary $\times$ the length of the longest SMILES of all molecules, specifically 73×398 in this dataset, which means there is zero-padding after the short molecules.

4 Methods

Since chemical groups are defined by the spatial arrangement and bonding of multiple of atoms in space, and these atoms are proximate to each other, biochemical interactions in molecules are predominantly local effects[2]. As convolutional neural network (CNN) has shown its advantage in feature detection, it is also able to detect the local features in molecules. Therefore, CNN is used in this project to automatically learn the local features from the one-hot SMILES matrix of the drug molecules and predict the drug toxicity.

Network architecture The main structure of my CNN model contains two sub-models, illustrated in Figure 1. Two sub-models do not interact with each other, they share the same original input, and the output is the arithmetic mean value of the outputs from Model 1 and Model 2. In the three convolution layers in Model 1, the L2 regularization is used to avoid overfitting.

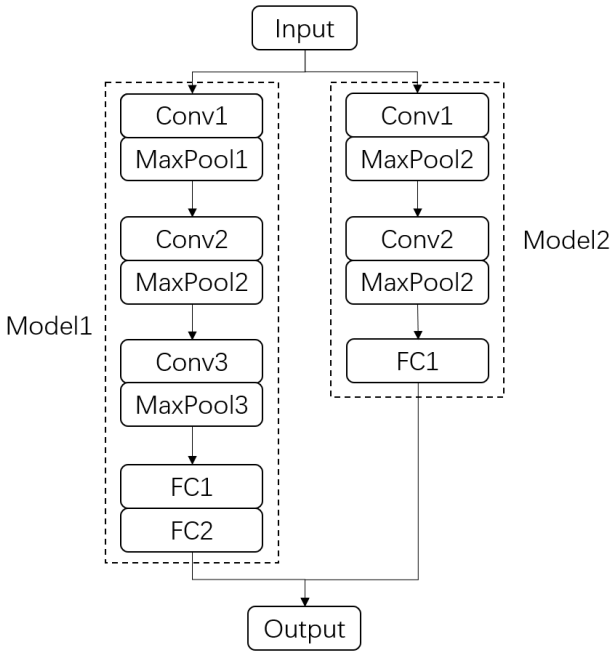

Figure 1: Structure of CNN Model

More detailed configurations of each layer are shown in Table 1.

Table 1: Configurations of each layer

model	layer	# of filters	kernel size	strides	padding	activation
Model1	Conv1	64	(5,5)	(1,1)	same	relu
	MaxPool1	64	(2,2)	(2,2)	valid	/
	Conv2	128	(3,3)	(1,2)	same	relu
	MaxPool2	128	(2,2)	(2,2)	valid	/
	Conv3	256	(3,3)	(1,2)	same	relu
	MaxPool3	256	(2,2)	(2,2)	valid	/
	FC1	32	/	/	/	relu
	FC2	2	/	/	/	none
Model2	Conv1	32	(5,5)	(1,1)	same	relu
	MaxPool1	32	(2,2)	(2,2)	valid	/
	Conv2	32	(3,3)	(1,2)	same	relu
	MaxPool2	32	(2,2)	(2,2)	valid	/
	FC1	2	/	/	/	none

Model training Two sub-models can be trained separately, and their weights are saved in independent files, for it is possible to get better performance of both models at the same time in the final prediction. The batch size of an iteration is 128, the loss function is binary cross entropy (BCE), and the optimizer is Adam with learning rate of 0.0001 for Model 1 and 0.001 for Model 2. As I observed that the raising of validation score usually converges within $20 \sim 30$ epochs, the epoch for training is set as 30. During training, the weights with best performance of validation data are saved for the test later.

5 Results

The prediction of the test dataset is submitted to the online Kaggle competition, and the result is reported in the area under the receiver operating characteristic (AUC), which is a common evaluation method for binary classification problems. First, I test Model 1 or Model 2 alone, and get 0.86653 and 0.85448 respectively. Then I submit the arithmetic mean of the outputs from two sub-models, which is the final prediction of my model, and get the score of **0.86727**. It shows that it is effective to combine the outputs from different models together, which is actually a common method used in deep learning.

6 Discussions

The most time-consuming part of this project is trying different network architectures, tuning the hyper-parameters in network layers like number of filters, size of filters, padding, units in fully-connected layer, and etc. As shown in Table 2, when I designed the structure for the dense layers after three convolution layers in Model 1, I have tried different sizes, for just one layer, or two layers. I run the model for several times , and evaluate it by the average validation score. Finally, I choose the one-32 layer. I have also tried to add batch normalization layers between convolution layers and pooling layers, but it seems that they do not work.

Table 2: Trials on FC layers

1 layer		2 layers	
units	valid score	units	valid score
4	0.8091	64+32	0.8035
8	0.8014	128+32	0.8116
16	0.8144	256+32	0.8100
32	0.8181	32+16	0.7961
64	0.8062	128+64	0.8107
128	0.8070		
256	0.8007		

The early version of my model only contains the Model 1 part, when doing some tests, I found that the demo network from TA actually performed very well in the online test, so I decided to put it in my model and calculate the mean value of two sub-models.

Ryu et al.[11] point out that graph convolutional network (GCN) can outperform CNN, as it can make better use of the molecular graph representation, and identify important features by analyzing the relations between neighboring atoms and bonds. From the discussion with other students in this course, I know that GCN indeed has a better result than CNN. So I tried to realize a GCN model by myself, but I failed to train it well. Due to the time limit, finally I returned to my CNN model, this is a pity for this project.

7 Conclusion

In this project, I use a convolutional neural network with two sub-models to learn features from the one-hot SMILES representation of molecules, and predict the drug toxicity of the test dataset. I get 0.86727 AUC score in the competition. I have also learned how to build a deep neural network, how to train the model and tune the hyper-parameters to make my model perform better. This is the first artificial intelligence programming project for me, which benefits me a lot. Sincere thanks to Professor Li and TAs for the tutorials and helps in this project!

References

[1] Kola I, Landis J. Can the pharmaceutical industry reduce attrition rates? *Nature reviews Drug discovery*, 2004, 3(8): 711-716.

[2] Wallach I, Dzamba M, Heifets A. AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv preprint arXiv*:1510.02855, 2015.

[3] Kitchen, D., Decornez, H., Furr, J., and Bajorath, J. Docking and scoring in virtual screening for drug discovery: methods and applications. *Nature Reviews Drug discovery*, 3(11):935949, 2004.

[4] Jenkins, J., Bender, A., and Davies, J. In silico target fishing: Predicting biological targets from chemical structure. *Drug Discovery Today: Technologies*, 3(4):413421, 2007.

[5] Xia, X., Maliski, E., Gallant, P ., and Rogers, D. Classification of Kinase Inhibitors Using a Bayesian Model. *Journal of Medicinal Chemistry*, 47(18):44634470, August 2004.

[6] Harper, G., Bradshaw, J., Gittins, J., Green, D., and Leach, A. Prediction of biological activity for high-throughput screening using binary kernel discrimination. *Journal of Chemical Information and Computer Sciences*, 41(5): 12951300, 2001.

[7] Byvatov, E., Fechner, U., Sadowski, J., and Schneider, G. Comparison of Support Vector Machine and Artificial Neural Network Systems for Drug/Nondrug Classification. *Journal of Chemical Information and Computer Sciences*, 43(6):18821889, September 2003.

[8] Dahl, G., Jaitly, N., and Salakhutdinov, R. Multi-task neural networks for QSAR predictions. *CoRR*, abs/1406.1231, 2014.

[9] Chen, Jiarui, Hong-Hin Cheong, and Shirley Weng In Siu. BESTox: A Convolutional Neural Network Regression Model Based on Binary-Encoded SMILES for Acute Oral Toxicity Prediction of Chemical Compounds. *International Conference on Algorithms for Computational Biology*. Springer, Cham, 2020.

[10] Mayr A, Klambauer G, Unterthiner T, et al. DeepTox: toxicity prediction using deep learning. *Frontiers in Environmental Science*, 2016, 3: 80.

[11] Ryu, Seongok, et al. Deeply learning molecular structure-property relationships using attention-and gate-augmented graph convolutional network. *arXiv preprint arXiv*:1805.10988 (2018).

[12] Unterthiner, Thomas, et al. Toxicity prediction using deep learning. *arXiv preprint arXiv*:1503.01445 (2015).