GeTP: Drug Molecular Toxicity Prediction Based on Graph-embedding Deep Learning

Yulong Hui
Shanghai Jiao Tong University
qinchuanhuiyulong@sjtu.edu.cn

Abstract

Drugs have become ubiquitous in our life, which has encouraged studies in the field of toxicity prediction with computational methods. Some researches have used DNN or CNN to deal with the string-format of drugs, however this method will lose the natural graph structure of the drug molecule. To tackle this issue, I transform the representation of drugs into graph format and construct some models based on some graph neural networks (GCN, TAG and GAT) respectively. After analyzing their capacities both theoretically and experimentally, I propose a **Graph-embedding Toxicity Prediction (GeTP)** Model, which fully utilizes and creatively combines the advantages of the graph neural networks above. I finally implement and test GeTP model on the data-set of kaggle, which shows significant improvement from the basic models and gains a pleasant performance.

1 Introduction

Nowadays, everyone is exposed to different chemicals through different sources, including food, household cleaning products, and medicines, while in some cases, these chemicals may be toxic and affect human health. Considering that the clinical toxicity prediction trials in the real world are very time-consuming, therefore, there rapidly raised the task to conduct the drug molecular toxicity assessment with computational methods (Raies and Bajic, 2016).

On the other hand, Deep neural network has become a hot research topic in machine learning in recent years. Compared to other methods, deep learning has shown its advantages in handling large amount of data and achieving better performance (Johnson and Khoshgoftaar, 2019).

Several deep learning methods have been proposed for the toxicity prediction task (Zhang et al., 2018; Mayr et al., 2016). However, in the most of the traditional state-of-the-art deep learning models (like DNN), drugs are represented as strings, which is not a natural representation thus the structural information of molecule-format is lost.

In this paper, I propose Graph-embedding Toxicity Prediction (GeTP) to predict the toxicity of the drugs with several common graph neural network models combined. I present the drugs by assigning the atoms as graph-nodes and assigning the bonding of atoms as graph-edges. My model is trained and tested respectively with the corresponding datasets, and the results show that my method gains a pleasant performance.

2 Related Work

2.1 Drug Representation

The most readable representation of molecules is called SMILES (Simplified Molecular Input Line Entry System)(Weininger, 1988). It is a linear representation for molecular structure using 1D ASCII

Submitted to CS410, Shanghai Jiao Tong University, China. Do not distribute.

strings. Because of its text-like feature, some natural language processing method can be used to solve related task. Alternatively, these strings can be seen as 1D representation, input into a convolution neural network (CNN).

However, in this paper, I want to fully utilize the natural graph representation of drugs. There are several studies finishing this work in other research fields about the drug molecules. For example, Nguyen et al. (2019) and Jiang et al. (2020) get the corresponding graph from the SMILES, they then use this information to solve the interaction between drugs and proteins. Inspired by this, I choose the similar way to embed the drug information into graphs.

2.2 Prediction Methods

With the task of toxicity prediction raising, a variety of computational methods have been used to develop the models, such as deep learning, support vector machines(Darnag et al., 2010), random forests(Polishchuk et al., 2009), structural alerts, read-across, molecular modeling (Zhang et al., 2018). Among them all, deep learning has rapidly emerged as a highly successful method because of the rapid development of the computational capacity (Schmidhuber, 2015). Moreover, in the specific molecule field, deep learning also has already been excellently applied to predict the outcome of biological assays (Ma et al., 2015), which makes it my prime structure for toxicity prediction.

There exists some related work based on deep learning. For instance, Liu et al. (2019) has accurately predicted the various of properties of the drug by using multi-task learning and DNN-GCN network. Öztürk et al. (2018) uses the representation of SMILE and analyzes the drug-target binding affinity using CNN, since the information is 1D. In this paper, I partly absorb the model from (Nguyen et al., 2019) and combine some other common graph neutral networks to solve the toxicity prediction task.

3 Model

3.1 Overview

Model GeTP can be separated into two main parts: the model of graph-data embedding (including molecule-feature extracting and format-transformation) and the model of predicting network.

Moreover, in the predicting part, I respectively utilize 3 basic graph-learning structures and analyze their disadvantages and capacities (Figure 2) at first. After that, I combine some components from the basic models and construct my advanced model (Figure 4).

3.2 Graph Embedding

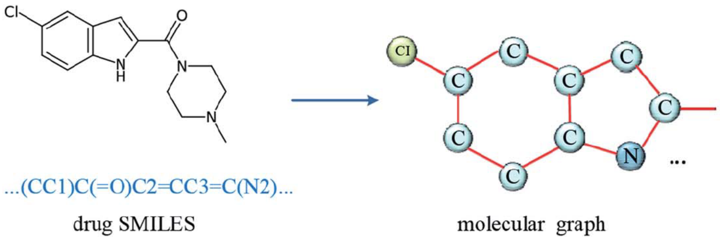

Figure 1. Graph embedding[1]

The drug compounds can be described as a graph of interactions between atoms. Therefore, handling input compounds in the form of graph representation, and subsequently applying learning algorithms on graphs may fit the task well. To this end, from each input drug compound string (SMILES), I construct a corresponding graph reflecting the interactions between the atoms. The formed graph will obey the original drug-format (Figure 1[1]), which means if there is a bonding between any two atoms (including self loop but except the hydrogen) then there must be a edge in the graph. More

[1] Due to the limited space, the constructed graph here is showed partly.

detailed, we can reconstruct the graph from the SMILE format by using RDKit (an open-source chemical informatics package (Landrum, 2013).

When it comes to the features of the nodes, I try to construct the feature vector with 5 atom features in the perspective of chemistry. The features include: the symbol of atom, the degree of the atom which describes the total number of the other connected atoms, the number of Hydrogen on the atom, the valence of the atom, and whether the atom is aromatic (Ramsundar et al., 2019). All of this chemical extraction can be implemented with Chem package in RDKit.

Moreover, in order to regularize and normalize the features which are in different representations, I transform their different formats into several 0-1 values and then do the normalization on this 0-1 vector. As a result, an indirect graph with regularly featured nodes is built for each SMILES string.

3.3 Basic Models

In this part, I **respectively** utilize 3 kinds of basic graph-learning structures (GCN, TAG and GAT) **one by one** [1] , and analyze their disadvantages and capacities in order to construct an advanced model by combination (Figure 2)

More specifically, after the input of the graphs and their features, there are 3 layers of graph neural networks activated by a ReLU function at the bottom, then a global max pooling layer is added to aggregate the whole graph representation. At the top there are 2 layers of fully connected networks, from which we can finally get the output result.

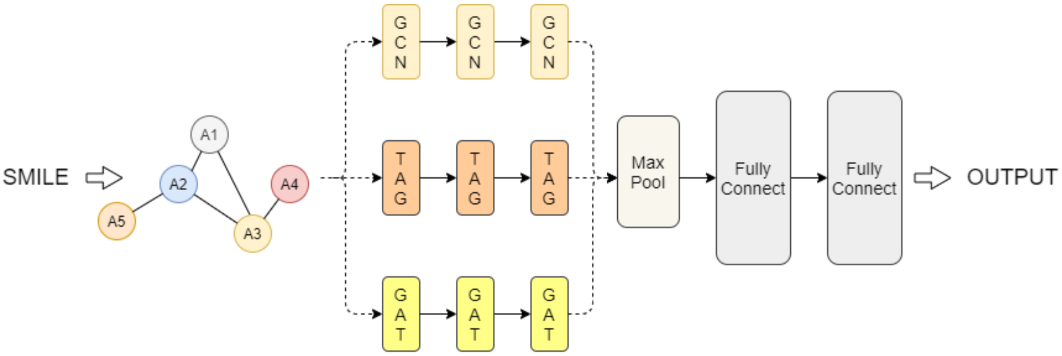

Figure 2. Basic Models

3.3.1 Graph Convolutional Network

GCN model (Kipf and Welling, 2016) was firstly designed to solve the problems of semi-supervised node classification. The model enables to learn hidden layer representations that capture both local graph structures and features of nodes. This advantage fits my indirect molecule-graphs quite well.

Formally, the precise process of GCN can be mathematically defined as follows:

$$h_i^{(l+1)} = \sigma(b^{(l)} + \sum_{j\in\mathcal{N}(i)} \frac{1}{c_{ij}} h_j^{(l)} W^{(l)})$$

where $\mathcal{N}(i)$ is the set of neighbors of node i, c_{ij} is the product of the square root of node degrees (i.e., $c_{ij} = \sqrt{|\mathcal{N}(i)|}\sqrt{|\mathcal{N}(j)|}$). W denotes the weight of parameters and σ is an activation function (I use ReLU in this paper) .

[1] Because of the limited space, I present 3 kinds networks in a single figure, while I utilize them one by one indeed.

3.3.2 Topology Adaptive Graph Convolutional Networks

Original graph convolutional neural networks (GCNs) require approximation to the convolution to alleviate the computational complexity, resulting in performance loss. Du et al. (2017); Kipf and Welling (2016) proposes the topology adaptive graph convolutional network (TAGCN), which provides a systematic way to design a set of fixed-size learn-able filters to perform convolutions on graphs. It no longer amounts to mere averaging of neighboring nodes like traditional GCNs, and can be mathematically defined as follows:

$$H^{(l+1)} = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

$\hat{A} = A + I$, where A denotes the adjacency matrix and I is the identity matrix. $\hat{D}$ is the diagonal node degree matrix of $\hat{A}$.

3.3.3 Graph Attention Network

Both of the traditional GCN and TAG models are universally used, while there exists a disadvantage that they handle the neighbours in a same way without considering the different relations between a nodes and its various neighbours.

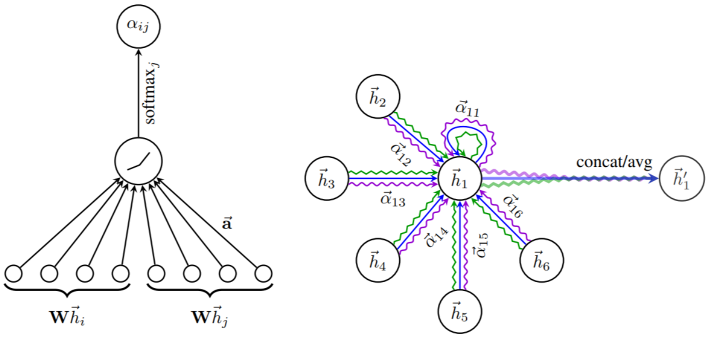

Figure 3. GAT Illustration

Therefore, I adopt graph attention network (GAT) (Veličković et al., 2017) in my model , which proposes an attention-based architecture to learn hidden representations of nodes in a graph by applying a self-attention mechanism (as Figure 3 shows). It can be mathematically defined as follows:

$$h_i^{(l+1)} = \sum_{j \in \mathcal{N}(i)} \alpha_{i,j} W^{(l)} h_j^{(l)}$$

where $\alpha_{i,j}$ is an attention score between node i and node j, while W denotes the weight of parameters:

$$\alpha_{ij}^{l} = \mathrm{softmax_i}(e_{ij}^{l})$$

$$e_{ij}^{l} = \mathrm{LeakyReLU}\left(\vec{a}^{T}[Wh_i \| Wh_j]\right)$$

3.4 Advanced model (GeTP)

Combining the above theoretical analysis, we can get the following points, which are proved to be correct in Section 4.1:

- Traditional GCN provides an idea of graph convolution, but its original format is not used widely in practice.
- TAG uses a topology graph which provides a symmetric normalization, which makes it perform and converge better in universal tasks than basic GCN.

- GAT considers the different attention between the various neighbours, which can get deeper information from the graph to learn and predict.

Therefore, I propose my advanced model GeTP with taking these points into consideration, and combines the basic components of TAG and GAT.

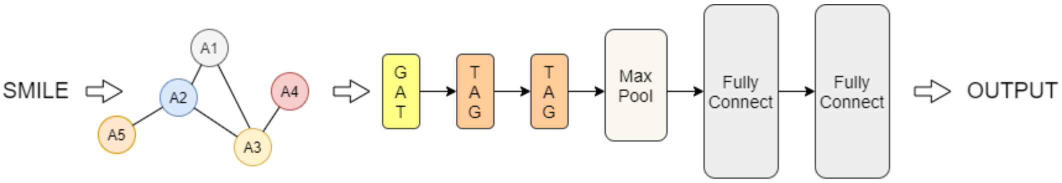

Figure 4. Advanced Model (GeTP)

As the Figure 4 shows, the SMILE is transformed into graph with features at first. Then I set a layer of GAT to train the molecule-graph, since GAT can extract deeper and more detailed information. After that, 2 layers of TAG are added because of its advantages in universal process and converging performance, while a max pooling is following to aggregate the whole graph representation from the nodes. Finally, there are 2 layers of fully connected networks, from which we can get the ultimate output.

4 Experiments

In my Experiments, the feature extracting is finished based on the Chem package in RDKit(Ramsundar et al., 2019), while the graph constructing and model training tasks are based on the DGL and Pytorch library.

I choose Adam as the optimizer and BCELoss as the loss function. The model is trained on a NVIDIA GeForce RTX 2080 GPU which is rented from Juchiyun platform.

4.1 Basic Attempt

To test and analyze the performances of baselines, I set the learning rate to 0.0008, and the batch size is set to be 32. The drop-out is used and its parameter is set to 0.5 in order to avoid the over-fitting.

The results of my basic attempt are as follows (Figure 5 6 7) :

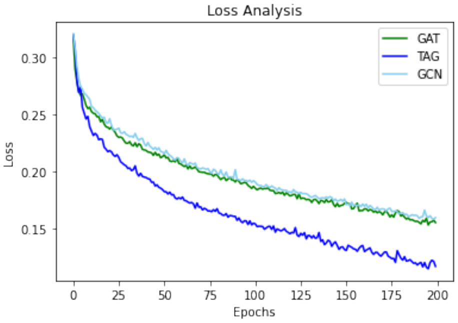

Figure 5. Basic Train-Loss

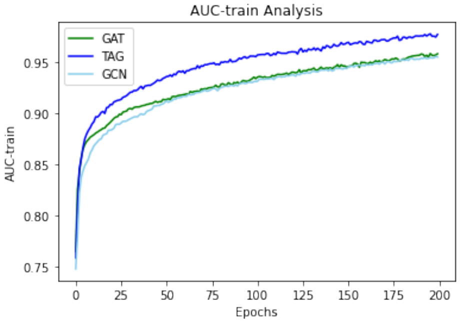

Figure 6. Basic Train-AUC

Figure 5 shows that when the model is trained, the TAG converges much better than other two models, which means it can converge faster and the value of loss is always much lower than others. Figure 6 shows the performance of training in another perspective. The TAG model can get to a higher AUC value in the train data-set with less steps.

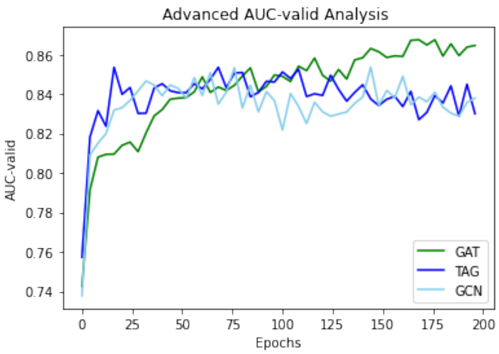

Figure 7. Basic Valid-AUC[1]

Figure 7 shows the performance on the validation data-set.[1] Apparently, the GAT model can get a higher AUC value than others when conducting the validation, which is mainly because the GAT model can extract deeper and more detailed information.

From the above result, the points in Section 3.4 can be proved to be correct. Therefore, we can fully utilize the advantages of better converging in TAG and deeper information in GAT to construct and implement my advanced model GeTP.

4.2 Advanced Implementation (GeTP Model)

In the advanced GeTP model, we finally set some valid parameters after abundant experiments, where the learning rate is 0.0005 and the batch size is 32.[2]

The results of GeTP implementation are as follows (Figure 8 9 10) :

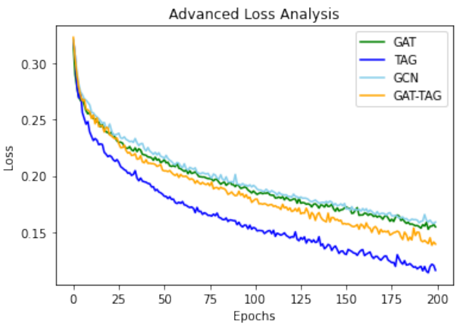

Figure 8. Advanced Train-Loss

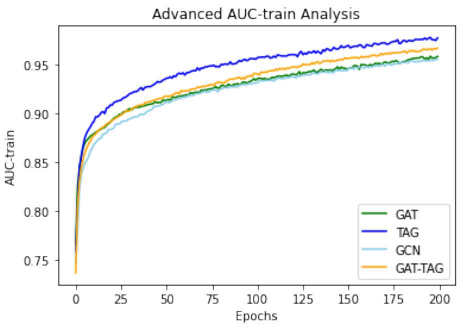

Figure 9. Advanced Train-AUC

Both Figure 8 and 9 show that the training converging performance of GeTP model is between the TAG model and the GAT&GCN models, which obeys the above analysis and estimation.

Figure 10 shows that the combination design of GeTP model outperforms on the valid data-set. Its best value of AUC is apparently higher than that in GAT model. One more thing to explain, GeTP model can converge faster, so it can converge to the best model around epoch 125, after which the redundant epoch causes the overfitting and decreasing of AUC.

[1] If I continuously increase the epochs over 200, the AUC-valid of GAT will still remain around 0.86 without constant increasing.

[2] Please check out my code, if you want to get other parameters.

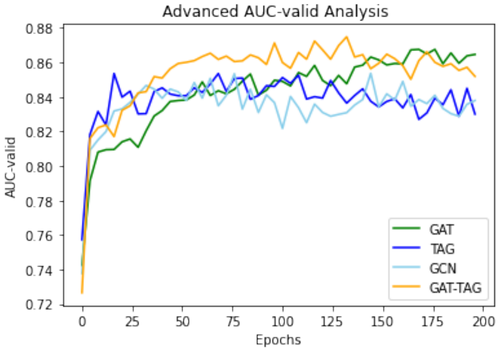

Figure 10. Advanced Valid-AUC

Table 1 shows the final result of the four models, where the presented data of each model is the best one selected from various implementations, with the Test AUC from the kaggle online test. Finally, the GeTP model gains a 8.45% improvement from the GAT model and achieves the AUC value of 0.88931.

Table 1. Experiment Results

Model	Valid AUC	Test AUC
TAG (baseline)	0.85435	0.87849
GCN (baseline)	0.85612	0.87926
GAT (baseline)	0.86157	0.88186
GeTP	**0.87018**	**0.88931**

5 Conclusions

In this paper, I propose an model GeTP for molecule-toxicty prediction by utilizing graph natural networks. I gain insight from the importance of natural molecular structure and transform the SMILE representation of drugs into graphs. Further, I construct and implement some models based on GCN, TAG and GAT respectively, and propose the GeTP model combined with GAT and TAG after analyzing their capacities.

I illustrate the performance gains of our model on the train,valid and test dataset. I also perform ample experiment results and their analysis to understand the effect of each module proposed in the paper and show how they are brought together to get the best performing model. In the final test and scoring, my model GeTP outperforms the baseline models and achieves the AUC value of 0.88931.

References

Darnag, R., Mazouz, E. M., Schmitzer, A., Villemin, D., Jarid, A., and Cherqaoui, D. (2010). "Support vector machines: development of qsar models for predicting anti-hiv-1 activity of tibo derivatives." *European journal of medicinal chemistry*, 45(4), 1590–1597.

Du, J., Zhang, S., Wu, G., Moura, J. M., and Kar, S. (2017). "Topology adaptive graph convolutional networks." *arXiv preprint arXiv:1710.10370*.

Jiang, M., Li, Z., Zhang, S., Wang, S., Wang, X., Yuan, Q., and Wei, Z. (2020). "Drug–target affinity prediction using graph neural network and contact maps." *RSC Advances*, 10(35), 20701–20712.

Johnson, J. M. and Khoshgoftaar, T. M. (2019). "Survey on deep learning with class imbalance." *Journal of Big Data*, 6(1), 27.

Kipf, T. N. and Welling, M. (2016). “Semi-supervised classification with graph convolutional networks.” *arXiv preprint arXiv:1609.02907*.

Landrum, G. (2013). “Rdkit documentation.” *Release*, 1, 1–79.

Liu, K., Sun, X., Jia, L., Ma, J., Xing, H., Wu, J., Gao, H., Sun, Y., Boulnois, F., and Fan, J. (2019). “Chemi-net: a molecular graph convolutional network for accurate drug property prediction.” *International journal of molecular sciences*, 20(14), 3389.

Ma, J., Sheridan, R. P., Liaw, A., Dahl, G. E., and Svetnik, V. (2015). “Deep neural nets as a method for quantitative structure–activity relationships.” *Journal of chemical information and modeling*, 55(2), 263–274.

Mayr, A., Klambauer, G., Unterthiner, T., and Hochreiter, S. (2016). “Deeptox: toxicity prediction using deep learning.” *Frontiers in Environmental Science*, 3, 80.

Nguyen, T., Le, H., and Venkatesh, S. (2019). “Graphdta: prediction of drug–target binding affinity using graph convolutional networks.” *BioRxiv*, 684662.

Öztürk, H., Özgür, A., and Ozkirimli, E. (2018). “Deepdta: deep drug–target binding affinity prediction.” *Bioinformatics*, 34(17), i821–i829.

Polishchuk, P. G., Muratov, E. N., Artemenko, A. G., Kolumbin, O. G., Muratov, N. N., and Kuzmin, V. E. (2009). “Application of random forest approach to qsar prediction of aquatic toxicity.” *Journal of chemical information and modeling*, 49(11), 2481–2488.

Raies, A. B. and Bajic, V. B. (2016). “In silico toxicology: computational methods for the prediction of chemical toxicity.” *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 6(2), 147–172.

Ramsundar, B., Eastman, P., Walters, P., and Pande, V. (2019). *Deep learning for the life sciences: applying deep learning to genomics, microscopy, drug discovery, and more*. " O’Reilly Media, Inc.".

Schmidhuber, J. (2015). “Deep learning in neural networks: An overview.” *Neural networks*, 61, 85–117.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2017). “Graph attention networks.” *arXiv preprint arXiv:1710.10903*.

Weininger, D. (1988). “Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules.” *Journal of chemical information and computer sciences*, 28(1), 31–36.

Zhang, L., Zhang, H., Ai, H., Hu, H., Li, S., Zhao, J., and Liu, H. (2018). “Applications of machine learning methods in drug toxicity prediction.” *Current topics in medicinal chemistry*, 18(12), 987–997.