Enhanced Molecular Representations for Property Prediction

Yuchun Dai and Yipei Xu *

Department of Computer Science, Fudan University

July 9, 2020

Abstract

Molecular property prediction has huge application scenarios in today's medical field. It can learn molecular property information from existing molecular pairs, so as to predict whether other molecular pairs work or not. With the outbreak of COVID-2019, the importance of molecular property prediction has become more prominent in the development of related vaccines.

In this project, we concentrate on the property prediction task that released recently by MIT[1]. We adopted a series of method for this task, including LSTM, Tree-LSTM, Chemprop, and a new method that we created based on the above methods. We achieve the result of 0.777 ± 0.230 for 10-fold CV ROC-AUC, and 0.139 ± 0.220 for 10-fold PRC-AUC on Tree-LSTM model with focalloss.[2]

Introduction

This is the final project for PRML courses, Spring 2020, Fudan University. In this project, our main task is to analyze the molecular properties and predict whether they will inhibit the virus. This task is also called molecular property prediction. Molecular property prediction has a long history, and there are some traditional statistical methods, such as QSAR, whose predictors consist of physico-chemical properties or theoretical molecular descriptors of chemicals, to solve this problem. After deep learning became popular, many neural network methods were also applied to this problem.The most common is graph convolutional neural network, which encodes the entire graph to a embedding and use the embedding for classification or regression prediction.[2]

However, there are still many difficulties remains in this field. Taking the task we are facing as an example, first of all, the way of modeling molecular structure is not perfect. SMILES is a commonly used molecular representation, it enters a chemical molecule in the form of a string, therefore, we can directly treat it as sequence information. We have made related attempts using RNN series networks. However, it is not enough to treat a molecule as a sequence. We need to use the structural information in the molecule, such as benzene ring, etc. So we can transform SMILE to a molecular structure diagram. At this time, the molecular information is Stored in the graph, we can use graph neural network to process. This is also the most common processing method at present. However, it is still not enough to treat the molecule as a planar graph. Because molecules are 3D substances and contain a lot of information in 3D space, such as orbital hybridization, chiral carbon, etc. the current model rarely considers the addition of these information.

In addition, the lack of task-related annotation data is also a major difficulty in this task. Compared with CV and NLP, the biological field is much more difficult, because the effect of the interaction between molecules must be verified in a professional laboratory, whose cost is very high. In our target data set, there are only 2098 examples in entire training set. What's more, the positive examples are even less, only 48 positive molecular appears in the data set, which also reflects the difficulty of finding the target drug. The imbalance of

*Equal Contributions. Our Student ID is 18302010047 and 18307130103 respectively.

[1]Details can be seen on: https://www.aicures.mit.edu/tasks

[2]Our code and model can be seen on https://github.com/cilebritain/PRML_Final

data distribution in the data set also brings many difficulties to the convergence of training.

So in this project, we aims to answer this two question: How to enhance the information in graph embedding, and how to train our model in this imbalance data set. For the first question, we use a series of method: LSTM, Tree-LSTM as sequence encoder, MPNN (Message Passing Neural Network) as graph encoder. We also add some new 3D features as the input to MPNN, thus enrich the information in the graph embedding. For the second question, we adjust our sample strategy, force the positive example must maintain a certain level for each batch. Using these method, we gained better performance on the data set than the ordinary method.

Related Works

There are many methods that can be applied to this task, which can be classified by solving problems from a sequence perspective and solving problems from a graph perspective. Here we mainly select some introductions that are inspiring to our project.

LSTM[1]: The full name is Long Short Term Memory neural network, is a kind of widely used RNN. In order to solve the problem of gradient disappearance in RNN, it introduces the structure of a memory gate, which can effectively use long-distance memory, and it is applied to many classic sequence problems. So it can also be applied to this task as the sequence method baseline.

Tree-LSTM[4]: If the input data has the structure of a tree, such as syntax tree for a sentence, we can consider how to adjust the network architecture for the input. Tree-LSTM also have many network cells, while the traditional LSTM take the hidden state of previous cell as input, the Tree-LSTM cell take the hidden state of his son node on the tree as the input. The forward process are calculated from the leaf node to the root. We can take the hidden state of root node as the embedding for this tree.

Chemprop[5]: In this method, The author used D-MPNN(Directed Message Passing Neural Networks), to encode the entire graph. MPNN is Operated on a graph, and it will pass the message between moleculars by making a message integration from the surrounding moleculars. It can be represented by following formulas:

$$m_{vw}^{t+1} = \sum_{k \in \{N(v) \setminus w\}} M_t(x_v, x_k, h_{kv}^t)$$

$$h_{vw}^{t+1} = U_t(h_{vw}^t, m_{vw}^{t+1})$$

Where m_{vw}^{t+1} is the message from v to w in the t+1 step. As we can see, the message is calculated from the neighbor node information of v(despite w), by a specific message function M. Then we can get the hidden state of h_{vw}^{t+1} by union the message and the hidden state of last step. We iterate this process for many steps, and finally we can get the entire graph embedding by adding the hidden state of each molecular together. Furthermore, the author aggregated many molecular features as input in this paper, including atom type, hybridization, etc. This measure enrich the input information of the graph, rather than just a plain SMILE string.

Seq2Seq-Fingerprint[6]: This paper has a new idea for molecular modeling: it used a unsupervised seq2seq encoder to encode the molecular. That is, it first train an unsupervised encoder-decoder for the SMILE string: the model input the SMILE string to the encoder, and train the model to predict the original string. After training, the encoder can be used as embedding layer. We can directly take the encoder out, and place another layer after the encoder, then use the new network to train our task. The parameter in the embedding consist of structure information which will improve the performance.

Approach

In this project, we mainly adopted two method: sequence method and graph method. Here are the detailed information of how our models are build and run.

Sequence approach:

There are many approaches using fingerprint to classify or regress. However, these methods simply employed recurrent neural networks on fingerprint can't use the spatial structure of the molecular well, and may meet long-rang dependence problem for molecular with a large length. Therefore, we propose our Tree-LSTM model.

We Build a tree based on the branch structure of a molecular. For every branch, we create a node to describe the features of it. And the father of the node is the backbone of the branch. As the branch

is indicated by '(' and ')' in the smiles string, we can build this tree easily.

For every node, we learn the features of its branches first, which is stored in its children nodes. So we must obtains the feature of all children nodes before we capture the feature of the current node. It can be seen as a progress from bottom to top in the tree. After the work finished for every children, we scan the smiles string in the current from left to right and record the substring we get. Every time we meet a position where a branch appears, we use a LSTM network to get feature from this substring and reset the substring to a null string. We add this feature into a list, then add the feature of the branch into the list later. So we can obtain a feature list in every node. We employ another LSTM network to this list and get the final result as the feature of this node. The overall feature of the whole smile string is the feature stored in the root node of the tree.

Graph approach:

As mentioned above, the Chemprop network can reach a good performance to this job. Here, we advance this model for the job.

First, we change the sample strategy. We set a limitation on the proportion of positive samples in the subset we select because the number of positive samples is too small. Because there are few positive samples in dataset, we hope this selection can make the model obtain a good performance on identifying positive samples, although it will degenerate on identifying positive samples, as our purpose is to observe potential component with activity.

Second, as mentioned in the article, the 3D spatial information can't expressed in the additional features. We add three values, the distances between an atom to three fixed atom in the molecular. Similar to ensure a point in a three dimension by three distance between an atom to three fixed atoms. We assume these additional features can be help this model learn obtain spatial information of the molecular. In this way, our input have 130 atom feature dimensions and 12 bond feature dimensions.

FocalLoss:

The FocalLoss[3] is proposed to solve the class imbalance problem in positive and negative samples selection. In this assignment, we discover the number of positive samples(1) is much smaller than negative samples(0), which indicates that our model can tend to learn how to identify negative samples rather than classify postve and negative samples. And balance the number of positive and negative samples decrease the performance in classification on negative samples. Therefore, we employ FolcalLoss as the loss function of our tree model.

Experiments

Metrics

The metrics for this task are ROC-AUC and PRC-AUC. Their full name are Receiver Operating Characteristic - Area Under Curve and Precision Recall Curve - Area Under Curve. As we can see, the ROC-AUC related to False Positive Rate(FPN) and True Positive Rate(TPR), and their formulas are as below:

$$FPR = \frac{FP}{FP + TN}$$

$$TPR = \frac{TP}{TP + FN}$$

In the ROC-AUC, we take FPR as the x axis, and the TPR as the y axis. we all know that in the classification model we can set a threshold to help us predict, so as we adjust the threshold from 0 to 1, the FPR and TPR changes as well. For each threshold, we can get a point in the coordinate system. we connect these points into a line, then we get the ROC curve. And the ROC-AUC is the is the area enclosed by the x-axis.

For PRC-AUC, the situation is similar. the x-axis is recall rate, and the y-axis is precision. Their formulas are as below:

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

the TP, FN in the four formulas above have the meaning of True Positive and False Negative, and so on to other shorthands. Similarly, the PRC-AUC is the area enclosed by the x-axis and PRC curve.

Model Setting

There are many hyper parameter for each model, we did our experiment and get our result in the following setting:

Bi-LSTM model: hidden_size=32*2, learning_rate=0.05, epochs=200, batch_size=100.

Tree-LSTM model: hidden_size=32, learning_rate=0.05, epochs=200, batch_size=100.

Chemprop model: positive_sample_rate = 0.25, batch_size=50, learning_rate = 1e-4, epochs=30.

Our models are train on the Nvidia RTX2080 GPU, the training time for each model is less than one hour.

Result

We did a series of experiment with different parameters, and we will show our best results for each model.

Here are the LSTM results:

Fold	**ROC-AUC**	**PRC-AUC**
Fold_0	0.803	0.078
Fold_1	0.766	0.156
Fold_2	0.753	0.042
Fold_3	0.467	0.023
Fold_4	0.878	0.112
Fold_5	0.861	0.048
Fold_6	0.551	0.019
Fold_7	0.657	0.006
Fold_8	0.719	0.032
Fold_9	0.783	0.035
Over-all	**0.724 ± 0.256**	**0.055 ± 0.101**

Figure 1: LSTM Results

The ROC-AUC index is normal, but the PRC-AUC is very low in this model, which means the model's precision is very low. That may caused by the lack of long distance memory, so we adapted Tree-LSTM method.

Here are the Tree-LSTM results:

Fold	**ROC-AUC**	**PRC-AUC**
Fold_0	0.792	0.091
Fold_1	0.735	0.132
Fold_2	0.724	0.330
Fold_3	0.591	0.347
Fold_4	0.829	0.178
Fold_5	0.791	0.137
Fold_6	0.562	0.053
Fold_7	0.801	0.011
Fold_8	0.860	0.143
Fold_9	0.784	0.035
Over-all	**0.747 ± 0.185**	**0.146 ± 0.201**

Figure 2: Tree-LSTM Result

As we can see, with the help of tree structure, the model performed better on both ROC-AUC and PRC-AUC. Though the PRC-AUC index is still not very high, it performed much better than the previous model. So we can conclude that the Tree-LSTM model can use the structure information to make a more precise prediction.

The imbalance of the data set is a big problem to us. In order to answer this question, we adopted focalloss to the model, and Here are the results:

Fold	**ROC-AUC**	**PRC-AUC**
Fold_0	0.726	0.234
Fold_1	0.904	0.290
Fold_2	0.813	0.101
Fold_3	0.669	0.068
Fold_4	0.909	0.157
Fold_5	0.772	0.359
Fold_6	0.546	0.021
Fold_7	0.889	0.020
Fold_8	0.872	0.117
Fold_9	0.666	0.020
Over-all	**0.777 ± 0.230**	**0.139 ± 0.220**

Figure 3: Tree-LSTM With Focalloss Result

The focalloss is work here. ROC-AUC even increase than the previous result. In this result, we can predict almost every positive example, so the ROC-AUC is a little bit higher, while we also predict many negative example to true, cause the low PRC-AUC.

Then, we use the Chemprop model, however, this model are not performed as well as we expected. That may caused by the unawared bugs in our code. Due to the time limit, we have not get well performed result on this model yet. And here are our results:

Fold	**ROC-AUC**	**PRC-AUC**
Fold_0	0.504	0.224
Fold_1	0.391	0.054
Fold_2	0.142	0.010
Fold_3	0.557	0.024
Fold_4	0.554	0.114
Fold_5	0.547	0.181
Fold_6	0.346	0.010
Fold_7	0.425	0.004
Fold_8	0.436	0.142
Fold_9	0.889	0.376
Over-all	**0.424 ± 0.452**	**0.114 ± 0.262**

Figure 4: Chemprop Result

As we can see, the ROC-AUC is much lower than the previous models. We check the result in

our model, and find that the prediction probability is less than 0.2 in the most cases, and many probability are around 0.18. We analyzed the results and think that the data set is too small to fully train this model. We also drew the curve to verify our conjecture, and the curves are as below:

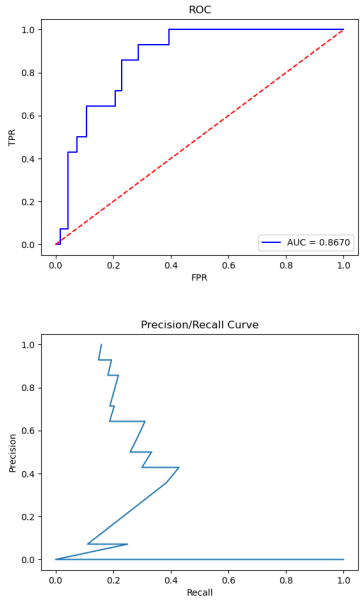

Figure 5: ROC curve and PRC curve for fold_0 by Tree-LSTM

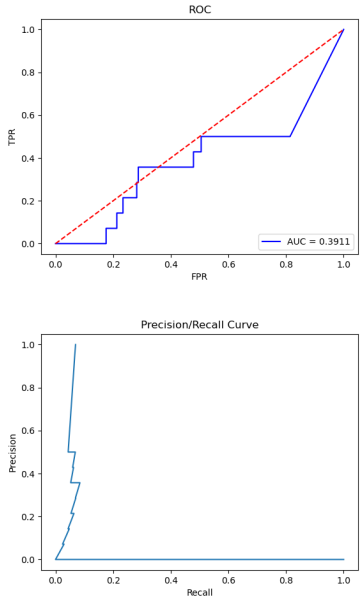

Figure 6: ROC curve and PRC curve for fold_0 by Chemprop

Comparing this four figures of two methods, we can find that for chemprop, it is not sensitive to the threshold, causing its FPR is not high even when threshold come to 0.8. But for the Tree-LSTM, it's FPR reached 0.8 when threshold is 0.2. In another word, Chemprop didn't do a good learning on the data set, so its prediction probability is not normal in this result.

We also made some attempts to improve chemprop's effect. We think that adding some 3D information can further improve the model effect. So we did the experiment as mentioned above. Here are its results:

Fold	**ROC-AUC**	**PRC-AUC**
Fold_0	0.731	0.334
Fold_1	0.400	0.056
Fold_2	0.269	0.137
Fold_3	0.575	0.146
Fold_4	0.486	0.110
Fold_5	0.576	0.183
Fold_6	0.364	0.010
Fold_7	0.135	0.002
Fold_8	0.503	0.145
Fold_9	0.755	0.361
Over-all	**0.479 ± 0.334**	**0.148 ± 0.213**

Figure 7: Refined Chemprop Result

Indeed, adding 3D information can improve some model effects, both on ROC-AUC and PRC-AUC. However, the final effect is still not satisfactory. We also noticed that the results on the official website claiming to use the chemprop method are far better than the results we obtained. We guess that they should use ensemble methods or transfer learning. For time reasons, we have not optimized this chemprop method.

Conclusion

In this project, We have studied and improved some methods in the field of property prediction, and made some explorations on both sequence methods and graph methods. For the sequence method, we used Tree-LSTM with focalloss, we get our best results. For the graph method, we added more features to chemprop. Although the overall result is not good, it is still improved from the previous result. We hope that in the future, we can further study this issue and make further improvements.

References

[1] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[2] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608, 2016.

[3] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

[4] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.

[5] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, 2019. PMID: 31361484.

[6] Feiyun Zhu Zheng Xu, Sheng Wang and Junzhou Huang. Seq2seq fingerprint: An unsupervised deep molecular embedding for drug discovery. *BCB'17, Aug 2017, Boston, Massachusetts USA*, 2017.