Drug Analysis by Using Machine Learning And Deep Learning

Xueli Zhou
Columbian College of Arts
and Sciences
The George Washington University
Washington D.C.
Email: xueli81993@gmail.com

Siyuan Hu
Columbian College of Arts
and Sciences
The George Washington University
Email: ....com

Ruoyu Wang
Columbian College of Arts
and Sciences
The George Washington University
Email: wry@gwu.edu

***Abstract*—This project aims to find some useful informations from medical data. At first, we give an introduction about this project, and talk about some related works. Next, we use some traditional method to analyze these data. Then, we use some deep learning method to figure out our problem. And, we try to implement our model with more sensible models, and do some ensemble. At last, we talk about our comparison and conclusions.**

***Keywords*—Drug analysis, Machine learning, Deep learning, ECFP**

I. INTRODUCTION

Our Project aims to find some relations from medical informations. Since the average price tag for getting a new medicine has risen up to 2.5 billion dollars. And the estimated time that need cost is 10 to 15 years. So it is meaningful if we can get some information before we start the experiments. And hope those information can drive some decision for us.

Furthermore, the number of potential medicines is pretty huge while the number we can try is strictly limited by the time and finance cost.

A. Data Description

The datasets have the target, smile, fingerprint, occp, value and measurement.

Target is the object influenced by the medicine. In the CH dataset we have only one target. As for the D2 dataset, we have many target, which we need to predict with different bucket.

Smile is the formula for the different medicine that is the origin information we have.

Fingerprint is the import information we generate from the structure with technology method. We have 1024 different binary value. From the knowledge of the technology, we can not get the fixed strides from this fingerprint, so the information is pretty hard to extract from it.

OCCP is another import information, that is also 1024 columns. We think this may stands for occupancy, i.e., how many times that each of the 1024 features(certain substructure) occurred in the drug compound. At this time we just take them as something correlate to the fingerprint.

Value is the target we have, and the part we want to use the machine learning to do some regression or classification.

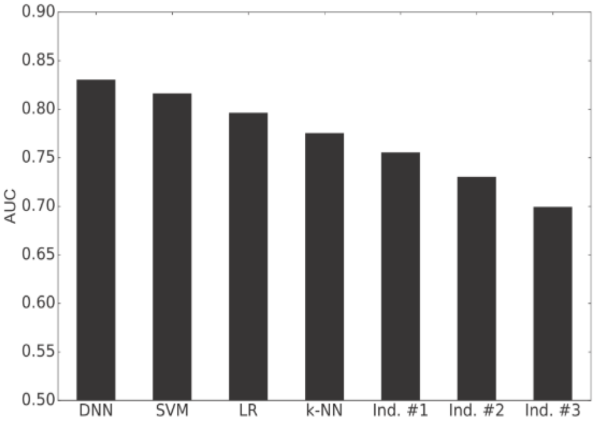
Measurement is the way how we get the value, there are several measurement we used here, at first we take them the same and then put them into different bucket to get a better performance.

II. RELATED WORK

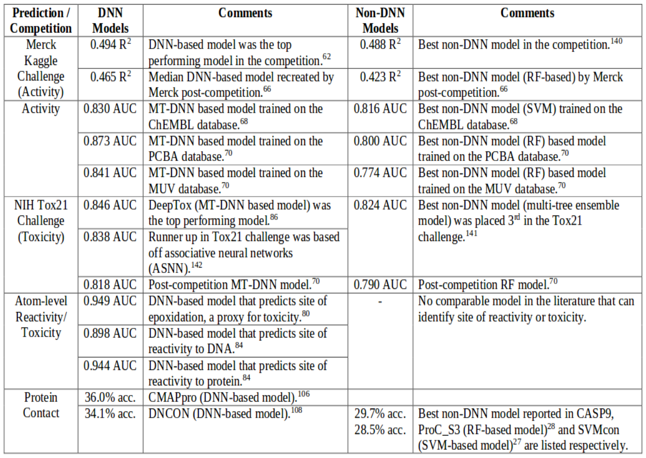
Up to now, deep learning has become the dominant algorithm among many data driven fields. Such as speech recognition, writing recognition, and medical informatics. In 2012, deep learning was introduced by Hinton and his team. And his team win the kaggle competition with their DNN model as a error rate of 16.4 %. This is the first time of deep learning into QSAR in Merck challenge in 2012. While the second team which based on traditional models only achieved the error rate of 26.2%.

In 2014, Dahl used another multi-task neural networks for this QSAR applications based on the kaggle algorithms.

Same year, Hochreiter and co-workers published a peer-review paper. In that essay, the authors used the ChEMBL database including 743,336 compounds, and 5069 targets. They used ECFP4 fingerprints and have a AUC of 0.83.



A. *Other related models*



Prediction / Competition	DNN Models	Comments	Non-DNN Models	Comments
Merck Kaggle Challenge (Activity)	0.494 R^2	DNN-based model was the top performing model in the competition.[62]	0.488 R^2	Best non-DNN model in the competition.[140]
	0.465 R^2	Median DNN-based model recreated by Merck post-competition.[66]	0.423 R^2	Best non-DNN model (RF-based) by Merck post-competition.[66]
Activity	0.830 AUC	MT-DNN based model trained on the ChEMBL database.[68]	0.816 AUC	Best non-DNN model (SVM) trained on the ChEMBL database.[68]
	0.873 AUC	MT-DNN based model trained on the PCBA database.[70]	0.800 AUC	Best non-DNN model (RF) based model trained on the PCBA database.[70]
	0.841 AUC	MT-DNN based model trained on the MUV database.[70]	0.774 AUC	Best non-DNN model (RF) based model trained on the MUV database.[70]
NIH Tox21 Challenge (Toxicity)	0.846 AUC	DeepTox (MT-DNN based model) was the top performing model.[86]	0.824 AUC	Best non-DNN model (multi-tree ensemble model) was placed 3rd in the Tox21 challenge.[141]
	0.838 AUC	Runner up in Tox21 challenge was based off associative neural networks (ASNN).[142]		
	0.818 AUC	Post-competition MT-DNN model.[70]	0.790 AUC	Post-competition RF model.[70]
Atom-level Reactivity/ Toxicity	0.949 AUC	DNN-based model that predicts site of epoxidation, a proxy for toxicity.[80]	-	No comparable model in the literature that can identify site of reactivity or toxicity.
	0.898 AUC	DNN-based model that predicts site of reactivity to DNA.[84]		
	0.944 AUC	DNN-based model that predicts site of reactivity to protein.[84]		
Protein Contact	36.0% acc.	CMAPpro (DNN-based model).[106]		
	34.1% acc.	DNCON (DNN-based model).[108]	29.7% acc. 28.5% acc.	Best non-DNN model reported in CASP9, ProC_S3 (RF-based model)[28] and SVMcon (SVM-based model)[27] are listed respectively.

In 2015, Merch published a study paper, talking about the comparison between DNNs and RF-based models.

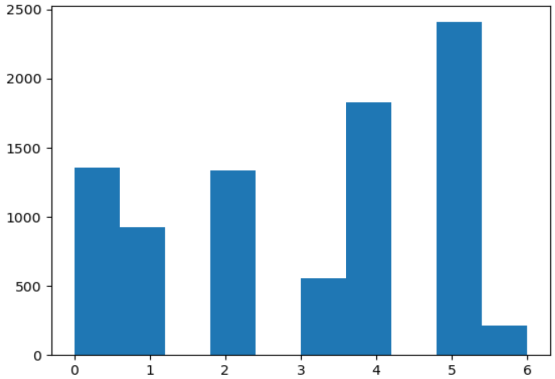
III. Traditional Method

Since the dataset do not obey some order. We split the data simple.

$X\_train, X\_test = X[: int((0.8*len(X)))], X[int((0.8*len(X))) :]$

$y\_train, y\_test = y[: int((0.8*len(X)))], y[int((0.8*len(X))) :]$

The target distribution seem like this:



A. *Evaluation results*

We use F1 Score to check performance. In statistical analysis of classification, the F1 score is a pretty good measure of the test evaluation. This macro F1 score including both precision and recall. Precision is the number of true positive divided by all positive results, and recall is the number of true positive divided by the positive results that should have been returned. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst at 0.

$F1 = 2 * (precision * recall)/(precision + recall)$

The reason why we do not use R square or AUC is that sometimes the increasing of AUC doesn't really reflect a better classifier.

B. *traditional results*

We used Support Vector Machine, Random Forest(number of decision tree, depth of decision tree), Naive Bayes to predict the results. And the results are shown below.

Method	SVM	NB	RF(100,100)	RF(100,15)
Train_F1	0.5066	0.4913	0.9935	0.8467
Test_F1	0.1334	0.1362	0.1282	0.1211

TABLE I. Traditional method F1 Score

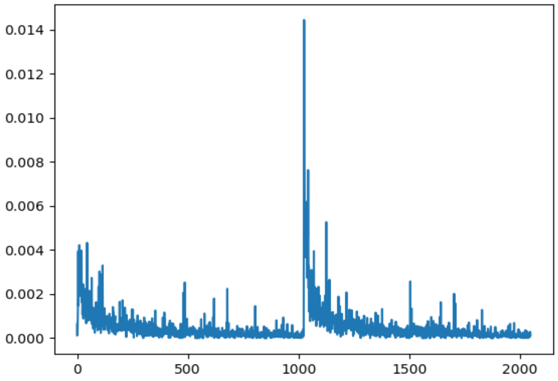
No one has the a better result for our result. Since one divided by seven equals to 0.14285.

C. *Random Forest*

However, we still looked into the important feature of data. And plot one graph for these 2048 features. And choose the first 25 in fingerprint and the first 25 in occp.

RF accuracy : 0.130714677679 RF train accuracy: 0.982164744178

The result does not change anyway.



RF accuracy : 0.130714677679 RF train accuracy: 0.982164744178

The result does not change anyway.

IV. Deep Learning

A. *Basic Neural Network*

We will try to use several structures of Basic Neural Network to check the prediction.

1) 1 hidden layer 2048-1024-7:

convergence in 6000 iterations
Training set accuracy:0.6908
Testing set accuracy:0.2520
Cost:1.3206

2) 2 hidden layer 2048-1024-512-7:

convergence in 3200 iterations
Training set accuracy:0.43
Testing set accuracy:0.2485
Cost:1.3959

3) 3 hidden layer 2048-1024-512-256-7:

convergence in 2800 iterations
Training set accuracy:0.9628
Testing set accuracy:0.3042
Cost:1.1936

4) 10 hidden layers:

Training set accuracy:0.9941
Testing set accuracy:0.1882
Cost:1.16926

B. *Convolutional Neural Network*

We will try to use several structures of CNN to check the prediction.

1) CNN1: stride : 2 padding: 8

iterations 3500 Training set accuracy:0.9941
Testing set accuracy:0.2625
Cost:0.02

2) CNN2: stride : 2 padding: 16

As the time I change channels into a small size and increase the padding scale. This CNN achieve a 98% train accuracy after 500 iterations. With the cost less than 0.1.

iterations 1100 Training set accuracy:0.9941
Testing set accuracy:0.2839
Cost:0.017

3) CNN3: stride : 4 padding: 16

Training set accuracy:0.9941
Testing set accuracy:0.2694
Cost:0.021

4) CNN4: stride : 8 padding: 32

iterations 2500 Training set accuracy:0.9942
Testing set accuracy:0.2451
Cost:0.01

C. *Recurrent Neural Network*

We will try to use several structures of CNN to check the prediction.

1) 1024-512-256:

iterations Training set accuracy:0.99
Testing set accuracy:0.33
Cost:0.99

2) 512-256-128:

iterations Training set accuracy:0.99
Testing set accuracy:0.33
Cost:0.99

3) 512-128-64:

iterations Training set accuracy:0.99
Testing set accuracy:0.33
Cost:0.99

4) 256-128-32:

iterations Training set accuracy:0.99
Testing set accuracy:0.33
Cost:0.99

D. *Combine Fingerprint and OCCP*

E. *confusion matrix for the first 100 rows in test set*

V. Conclusion

A. *summary*

In our project, we experiment several neural works with different parameters. Aims to analyze those information to get a direction of model adjustment. Such as :
1)the number of iterations which the loss function has been convergent.
2) The accuracy of training set
3) The accuracy of testing set
4) The loss function after convergent

B. *implement and future work*

We do not run the neural network more than 10,000 iterations. Since the cost is extremely huge. And the dataset is not big enough for our analysis.

Up to now, the best result is from ... . We can implement our model by using ensemble method. Adding more features into the dataset, or adding more instance into the dataset. And using cross-validation as well.

The most import thing we need to implement is feature engineer. In this project we still not find a good way to extract information from the fingerprints and occp.

Acknowledgment

Thanks to our professor Zeng Chen who give us many help on our project.

References

[1] Deep Learning for Computational Chemistry,Garrett B. Goh,Nathan O., Hodas, Abhinav Vishnu

[2] Ajay; Walters, W.p.;Murcko, M.A.J.Med.Chem. 1998,31,3314

[3] Burden, F.R.; Ford, M.G.; Whitley, D.C.; Winkler, D.a.J.Chen=m. inf. Comput. Sci. 2000, 40,1423

[4] Cramatica, P.QSAR Comb. Sci. 2007, 26, 694

[5] Verma, J.; Khedkar, V.M.; Coutinho, E.C.Curr. Top. Med Chem. 2010, 10,95

[6] Kaggle

[7] Dahl, G.E.; Jaitly, N.; Salakhutdinov, R.arXin:1406.1231 2014

[8] Ma,J.;Sheridan, R.P.; Liaw, A.;Dahl, G.E.; Svetnik, V.J. Chem. Inf. Model. 2015m 55m 263