Final Project: Molecular Property Prediction

CS598 Deep Learning for Healthcare

Github: https://github.com/tonymazn/CS598

Zhouning Ma
University of Illinois
zm11@illinois.edu

ABSTRACT

Drug discovery could take a lengthy process that usually takes more than 5 years, and it is costly as well with billions of dollars on bills **[1]**. There are a few research tasks under drug discovery**,** the first one is molecular property prediction task which the input is a drug molecule and the output is the drug's properties. The second task is drug reposition with the input of drug proteins and molecules, and deep learning (DL) model predicts the affinity score, and the third task is drug-drug interaction with the input of different drugs and we want to predict the interactions between two drugs by deep learning. The lab will test drug-drug. The fourth one is de novo design which is the reversion of molecular property prediction **[2]**.

Deep Learning usually can be very helpful in those tasks. Due to the time limitations for the final project, we will only focus on one task: Molecular Property Prediction.

We propose a **Conditional GANS** (Generative Adversarial Networks) **[3]** to learn the dataset QM9 **[4] [5]** with its chemical space to discover molecular property prediction. The main idea came from one of the models in Yuemin Bian and Xiang-Qun Xie's paper **[6]**, We will demonstrate the detail in the following sections.

Keywords

Drug Discovery, Molecular Property Prediction, Deep Learning, Healthcare

1. INTRODUCTION

Drug discovery consists of the stages of molecule property prediction, preclinical studies, and clinical trials. Molecular property prediction is the core task of cheminformatics fields [7]. Usually researchers screen a large number of compounds to find the potential matching biological molecules, it costs very much money and time [8]. We found the drugs US Food and Drug Administration approved in these couple of years are small-molecule drugs.

To find the molecular property is searching the chemical space, and output the matching molecular property. Deep learning is playing a key role to do the job of molecular property prediction, drug discovery at the current date. Deep learning generative models are the fresh models of the drug discovery, one of the Generative Adversarial Network (GAN) [3]'s strengths is domain transfer: molecular property and chemical space, we propose a GANs model (called it GAN19) for molecular property prediction.

1.1 SMILES vs Graph

SMILES is the string to represent the molecule and Graph representations are another one.

Figure 1: Example of the two representations of the same molecule, the left-hand side is SMILES representation, the right-hand side is the Graph representation.

Most recurrent neural networks (RNN) work on SMILES type of encoding, the reason is RNN could solve the problem like Natural Language Processing (NLP), the disadvantages are some of the RNNs have to learn the rules and the order of representation [9]. Other models are using Graph-based representation of molecules such as MolGAN [9] which is a likelihood-free method and directly works on the graph.

With a focus on SMILES, in particular, a string-based representation generative adversarial network (GAN) [3], we will let GAN learn the rules and order of representation, and we will demonstrate that GAN19 model can offset the RNN disadvantages, and GAN19 model can work better on the molecular property prediction, and experimental results show GAN19 outperformed.

2. RELATED WORK

Deep Learning transfers one domain to another. And we have fast computers and big data to run deep learning networks especially large neural networks. The performance of deep learning algorithms is superior in comparison to traditional machine learning algorithms, and deep learning algorithms get better when using more and more data and traditional machine learning is not scalable.

There are a few traditional machine learning model such as the quantitative structure-activity relationship (QSAR) [10] model which was one of the early methods to solve the molecular property prediction problem, this model has good performance in this field, the machine learning models included support vector machines (SVM) [11], random forest (RF) [12] which has been used in chemical property prediction too.

Deep learning is a new way to solve the same problem, and LSTM (Long short-term memory) [13] is an example of generative chemistry method in Recurrent Neural Network (RNN). More and more deep learning networks are using for generative chemistry, such as variational autoencoder (VAE) [14], Gated Recurrent Unit (GRU) [15], adversarial autoencoder (AAE) [16], and the one we are using in this paper: Generative Adversarial Network (GAN) [3].

Generative adversarial network (GAN) is designed by Ian Goodfellow in 2014 [3]. It has included at least two neural networks: one is called generator and another is discriminator. The generator randomly generates examples, and the discriminator classified the examples as real or fake, this is very intelligent.

3. METHOD

We will present the problem setting and our solution in this section. GANs are unsupervised learning task, which involves automatically learning and discovering the rules, the generator generates data without the verification of the result, but the discriminator will identify the result as 'Real' or 'Fake', therefore, the generator will 'learn' to generate a better result to fool the discriminator. Section 3.1 is the problem settings, and in section 3.2 we will expound on the detail of our model Gan19 (Figure 2).

3.1 Problem Settings

The task of this paper is to explore a model which can yield an output of the molecules with inputs of molecule property. For example, the baseline of 'mu' property is 2.5, we train the model based on the 'mu' property is greater than 2.5 or less than 2.5, when the model is fine tune trained, we should be able to expect that the model will provide a list of which 'mu' greater than 2.5 or less than 2.5(the rest items are less). This feature will help with the process of molecular property prediction, and save the time and cost to discover similar molecules.

3.2 Generative Adversarial Networks

Algorithm: GAN [17]

Initialize O_d , O_g for D, G

for each iteration **do**

Sample examples $\{x^1,x^2,\ldots,x^n\}$ from data

Sample noise $\{n^1, n^2, \ldots n^n\}$

Obtaining generated data $\{x^1, x^2, \ldots x^n\}$ and $\{n^1, n^2, \ldots n^n\}$

Update discriminator parameters O_d to maximize

Update generator parameter O_g to minimize

End

The generator's dataset is difference from the discriminator's dataset, the SMILES in the generator may or may not exist in the discriminator, and the generator's label is random.

3.3 SMILES Generator

Generate SMILES (Simplified Molecular Input Line Entry System) [18] is beyond this project's scope, but the discriminator 's input is SMILES vectors, and the vectors are coming from SMILES. One solution is to generate a random string that looks like SMILES because we don't have enough computation to enumerate all the strings (10^{60} [19]) which included all the SMILES. We find another easy way to 'generate' SMILES, the converted qm9 dataset included SMILES information, we split qm9 dataset into two: one is the real example for discriminator input, and another one is the output of the generator. Then we have SMILES as output for generator, we also build some relationship between the generator output to that part of the dataset. The generator input is a random number. With generator's Convolutional Neural Network, the output is an index that can locate the position of the item in the SMILES half dataset, then we have the SMILES.

This solution will use half the data for the generator to output SMILES, which means only another half data to be used for discriminator, it is difference from non-GANs models, the non-GANs model usually keeps most data for training, and uses few data for validation and small amount data for testing.

3.4 Generator and Discriminator

The generator [20] 's input is random data, and the output is SMILES. To make this task simpler, we split the data (qm9) into two sets, one is the real data for the discriminator, another is the output of generator, the benefits are:

- When we do the evaluation, we need to know the random data 's true label for metrics;
- Our task is molecular property prediction and spending too much time on SMILES generator is not preferable. By search the dataset applicable for the generator, we should able to find the SMILES as output, which makes the task easier.

The discriminator [21] 's task is to indicate generator's result as 'Real' or 'False' by the discriminant function.

- The discriminator does not know generate random data;
- Real data are used to find a discriminant function;
- Data are Classified as real or fake;
- Output is usually binary (Real/Fake);

GANs modeling is to solve the unsupervised problem, and it is two player's game. The generator is like a 'counterfeiter' making the fake money, and the discriminator is like the police to detect and catch counterfeit money. The counterfeit must make like real money to fool discriminator, and that means generator network has to learn to create samples from the same distribution of training data. [3]

We have trained a great number of models, and the 19th GAN model of our training returned the best result, so we called this model 'GAN19'. What GAN19 accomplished is: the generator generates random SMILES with its random property, and they are passed to the discriminator. Then the discriminator checks and communicates to generator with the outcome of 'True' or 'False' by learning from real data's distribution, we owe credits the books "GANS in action" [22] and "Make your first GAN with Pytorch" credits [23] which talk about some technical details.

3.5 Conditional GANS

Conditional GANS was introduced by Mehdi Mirza and Flickr [24]. The conditional GANS involve using some additional information during the training. The conditional GANS are capable to determine what kind of data will be generated as the answer, the detail is conditional GANS direct the Generator to synthesize the 'random' data, same as what we have designed [22]. The famous conditional GANS examples are the one to ten number classes, the result could produce the number 6, and another example is to generate the images with the properties such as gender, age, or facial expression.

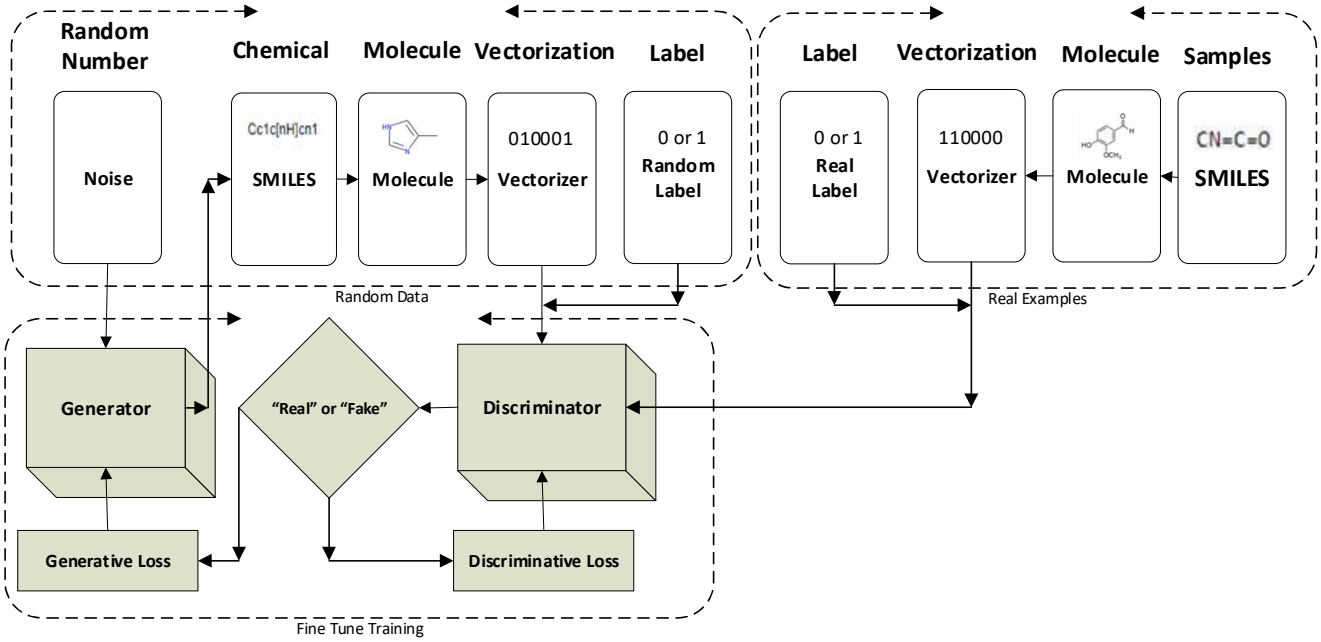

Figure 2: GAN19 architecture of the generative adversarial network. The networks comprises two components: generator and discriminator. And the generator will create some random data which is the partial of the discriminator's input, and the real examples are another input for the discriminator.

Our task is molecular property prediction, and we like to recommend some SMILES whose molecular properties are great than baseline, or opposition: less or equal to baseline. The Generator learns the random samples for each label and the Discriminator only accepts the real matching pairs or reject the mismatching pairs:

The fake sample from Generator is similar as the real sample. The Discriminator is trained by real samples (x, y) and fake samples (x*|y, y), the Discriminator learns to distinguish the real and fake data and then accept or reject the samples from Generator.

Algorithm: GAN19 Conditional GANS

for each iteration **do**

Generator:

Noise vector z, label y

G(z, y) = x*|y (output: SMILES|y)

Update Generator's gradient by descending:

$$\nabla_{\theta g} \frac{1}{m} \sum_{i=0}^{m} \log\left(1 - D(G(z^{(i)}))\right) [25]$$

Discriminator:

Real SMILES, label: (x, y)

Fake sample: (x*|y, y)

Output: 1/0 (1 = real, 0 = fake)

Update Discriminators gradient by ascending:

$$\nabla_{\theta d} \frac{1}{m} \sum_{i=0}^{m} [\log D\left(x^{(i)}\right) + \log(1 - D(G(z^{(i)})))] [25]$$

end for

GAN19 implemented the same conditional GANS architecture. The Generator randomly selects SMILS from the training dataset, generates fake labels (the property great than baseline), and the Discriminator receives real samples and labels, as well as fake ones, Discriminator trains by the real samples and fake samples to learn how to recognize samples and give the feedback to Generator as real or false.

3.6 Dataset

We established datasets as the following list for our project:

- Qm8
  included 20000 synthetically feasible small organic molecules. [4] [26]

- Qm9
  included 134000 small organic molecules and is the subset of GDB-17. [4] [5]

- Tox21 [27]
  Large-scale screening data: cells, informatics of genes, medicinal chemistry, and biochemical pathways.

- Hiv [28]
  Inhibit HIV replication

- GDB-17 [4]
  More than 160 Billion organic small molecules, this is too big for us.

- GDB-13 [29]
  13 atoms of C,N.O.S and CI up to 13 heavy atoms, this is too big for us too.

We pick qm9 dataset to train our algorithm, because of the size and the labels - the QM9 dataset [4] [5] has 134K small organic

molecules which correspond to the subset of GDB-17 [4] which called “chemical universe” [30] and it has more than 160 billion organic molecules [31]. We are interested in the following properties (Table 1), those properties are not in the original QM9, but we can get those properties from QM9 by cap394 [32] solution.

Table 1: Dataset properties

I.	Property	Description
1.	smiles	SMILES
2.	mu	Dipole moment

Table 2: Dataset Statistics

	SMILES	Labels
qm9	133,885	133,885

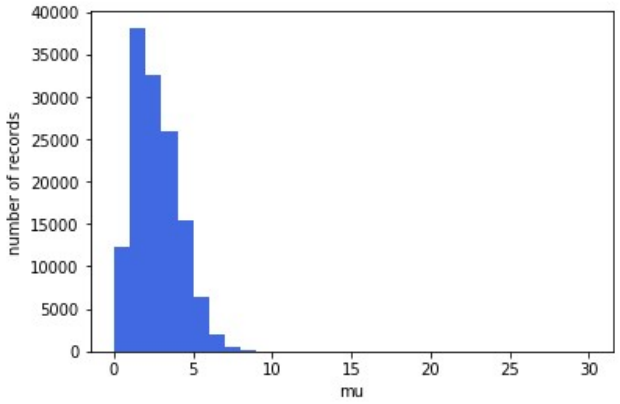

Figure 3: mu distribution

QM9 dataset’s feature ‘mu’: the minimum value is 0 and the maximum value is 29.5564, and the mean is close 2.5 for the whole dataset, 2.3 for the first 6000 rows, we have created a classifier for our project:

- **Greater than the mean, the value is 1**
- **Less than or equal to the mean, the value is 0**

The model is not limited to use mean to create the group; it could be any other number. We use the mean here to make the model simpler.

4. EXPERIMENT

We design experiments to evaluate GAN19 on the accurate model prediction.

4.1 Setup

Datasets: We evaluated GANs by qm9.

We split qm9 into two parts, one is for discriminator, and another half is as generator’s output by a dictionary query, which guarantees the format is SMILES. The generator might generate noise by random data, the generator does not receive any SMILE true property value. The noise included the random index to select SMILES and the random label for the selected SMILES. the generator doesn’t know any SMILES property’s true value, but we know the true value which provides an easy way for us to evaluate the result.

GAN19 Setup

We set up the following parameters to fit the model:

The Discriminator:

- The input hidden size: vector [0] * (vector [1] + 1)
  The plus one is the label one dimension
- Full connection to 200
- We find the LeakyReLU do a better job, we set the parameter to 0.02
- Applies layer normalization
- Full connection
- The activation function is Sigmoid

The Generator:

- Full connection to 200
- Again, we use LeakyReLU, parameter is 0.02
- Full connection
- Sigmoid as activation function

We use a pc with the following configuration:

The hardware environment:

- Intel CPU i9-10920x
- 64G RAM
- nvidia geforce rtx 2080ti

The software environment:

- Ubuntu 20.10 Groovy Gorilla
- Python > 3 + Pytorch

4.2 Rdkit

Rdkit is open-source cheminformatics and machine learning tools, the core data structures and algorithms are written by C++, Rdkit supports Python, there are some major functions of rdkit: such as the operations of 2D and 3D molecular [33], it is widely used in the application of machine learning [34]. To transfer to vectors, the input is molecule format, we need to convert the SMILES to molecular by rdkit.chem’s PandasTools in this project.

4.3 EBjerrum / molvecgen

After convert SMILES to molecular, we need another tool to generate molecular vectorization. EBjerrum/molvecgen meets what we need: the input is molecular, and the output is a vector [35].

4.4 Evaluation Strategies

We measure the accuracy by the following metrics:

- **ROC-AUC**: calculate operating characteristic curve (ROC-AUC) [36].
- **PR-AUC**: calculate "different thresholds" [37] via "precision-recall" [37].
- **F1 Score**: the balanced F-score/F-measure [38].
- **Number of Parameters:** the model's parameters

Table 3: GAN19 Training Parameters

Model	epochs	Batch Size	Learning Rate(D)	Learning Rate(G)
Baseline	16	1	0.00075	0.00075
Model 1	15	1	0.00075	0.00075
Model 2	15	1	0.00075	0.00075
Model 3	16	1	0.00070	0.00070
Model 4	16	1	0.00065	0.00065
Model 5	16	1	0.00500	0.00500
Model 6	16	1	0.05000	0.05000 *
Model 7	16	1	0.10000	0.10000 *
Model 8	16	1	0.50000	0.50000 *

* the model is not convergence

Table 4: Evaluation Metrics on GAN19

Model	QM9 (Molecules)	ROC-AUC	PR-AUC	F1
Baseline	6000	0.502	0.746	0.8542
Model 1	6000	0.526	0.820	0.9011
Model 2	133,885	0.502	0.721	0.8378
Model 3	6000	0.523	0.790	0.8827
Model 4	6000	0.516	0.700	0.8235
Model 5	6000	0.517	0.720	0.8372
Model 6	6000	0.516	0.690	0.8166
Model 7	6000	0.513	0.620	0.7654
Model 8	6000	0.510	0.530	0.6928

we have trained each model multiple times, and GAN can't get the same result each time, so we get the best ones as the model result.

Table 5: GAN19 Parameters

Model	#Parameters Generator	#Parameter Discriminator
Baseline	643,200	120,801
Model 1	643,200	120,801
Model 2	13,495,743	175,801
Model 3	643,200	120,801
Model 4	643,200	120,801
Model 5	643,200	120,801
Model 6	643,200	120,801
Model 7	643,200	120,801
Model 8	643,200	120,801

5. DISCUSSION

In this section, we will discuss the performance of different model parameter settings of the GAN19 models.

The Baseline, Model 1,2,3,4,5 is convergence, the Figures 4, 5 are typical diagrams of convergence, the values are changed between 0 to 1.

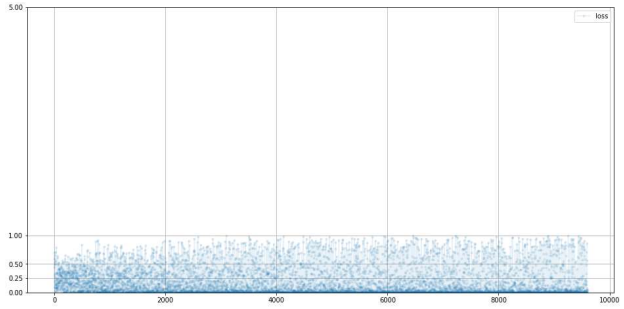

Figure 4: Baseline discriminator progress plot

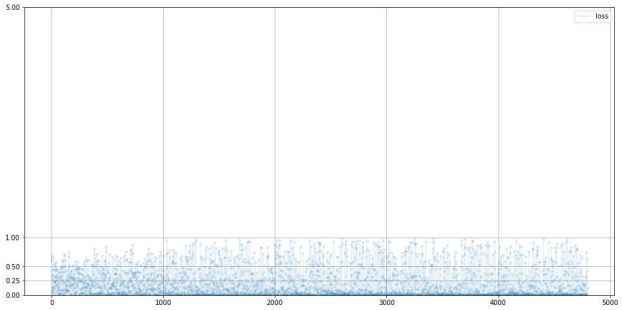

Figure 5: Baseline generator progress plot

Training model 2 used the whole dataset, and the same pattern is found (Figure 9,10) of the baseline model. The values are changing between 0 to 1.

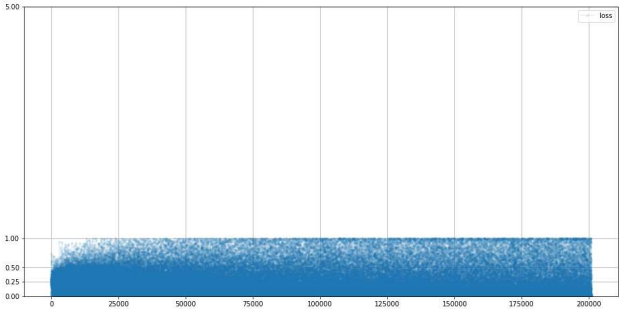

Figure 6: Model 2 discriminator progress plot

When checking the model 2 generator (Figure 7), the values are not only changing between 0 to 1, we could find the change is close 0.25 too [23].

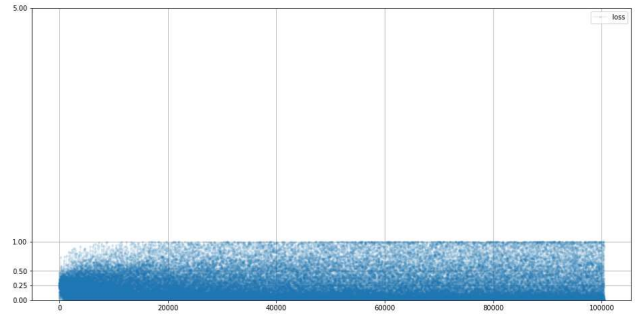

Figure 7: Model 2 generator progress plot

Model 6's diagram (Figure 8, 9) is an example of GAN program called diminished gradient, the discriminator gets too successful (the loss is zero), and the generator learns nothing.

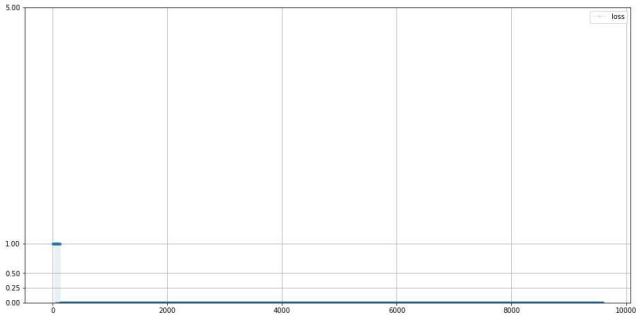

Figure 8: Model 6 discriminator progress plot

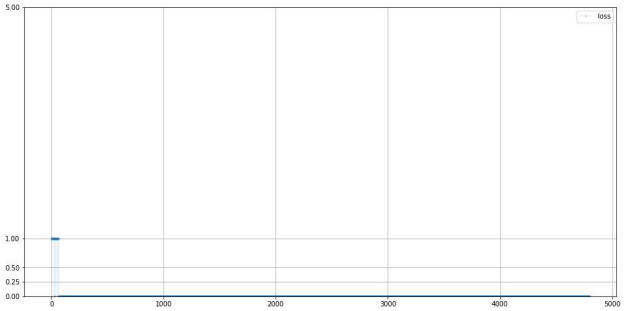

Figure 9: Model 6 generator progress plot

Model 8 is the example of mode collapse, and the generator produces limited samples, for which the PR-AUC is 0.530 only.

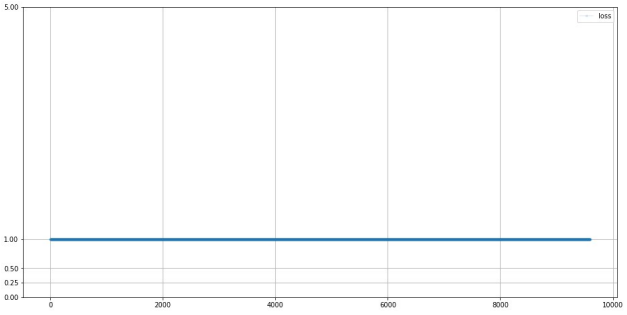

Figure 10: Model 8 discriminator progress plot

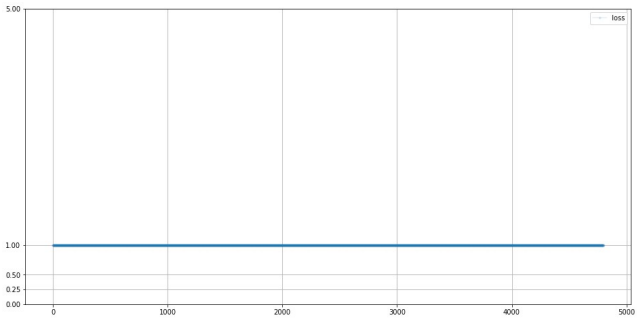

Figure 11: Model 8 generator progress plot

After the model was trained, the generator should be able to output the SMILE by the label, the following list is one example, we query the model to output SMILES by label's value is 1, we got 8 one in 10 outputs.

Table 6: GAN19 output

baseline: 2.3
SMILES: COC true value: 1.1502 y_true: 0.0
SMILES: c1cncnc1N true value: 3.6179 y_true: 1.0
SMILES: CC(N)(CC#C)C#N true value: 2.5013 y_true: 1.0
SMILES: c1conc1C=O true value: 4.8782 y_true: 1.0
SMILES: C(CO)C(=O)C=O true value: 3.2106 y_true: 1.0
SMILES: O=CCC1CCO1 true value: 3.6035 y_true: 1.0
SMILES: CN=c1n(cco1)C true value: 2.9829 y_true: 1.0
SMILES: NC1=NOC(=N)N1 true value: 4.3517 y_true: 1.0
SMILES: OC1C2CC(=O)C12 true value: 2.9846 y_true: 1.0
SMILES: c1(=O)onno1 true value: 1.0079 y_true: 0.0

6. CONCLUSION

We have developed a GANs for Molecular Property Prediction. Our models were training by a small amount of data first, around 6000 rows of total 130,000 rows' dataset to do a quick train, and the discriminator has guided the generator to update its output to be closer to the preferable result. The GAN19 has performed well and provided the desirable result on the proof-of-concept stage.

We also find the baseline model (generator and discriminator) both converged by the plot of progress as following (Figure 4 and 5). When we used mean squared loss (MSELoss), the loss is 0.25 (0.5 squared), the plot displays the same results.

We trained the GAN19 on the whole dataset, and the result of model 3 (Figure 6 and 7) is closed to baseline in that the Generator's parameters are 30 times of baseline model, but the data is 22 times of baseline model too.

We have changed the model parameters (learning rate) and train some new models (model 4 to 8), the final result is: learning rate value 0.00075(both generator and discriminator) is the best one up to present. The model can predict SMILE by label, it provides a tool to select molecules by properties in the drug discovery.

7. CHALLENGES

GANS are so hard to train, we can feel the pain during the project. We are still looking for a better way to improve the rate of success. Training GANS like a two-player non-cooperative game [39].

8. CONTRIBUTION

Zhouning Ma conceived the study, experiments, analyzed the experimental data, wrote the models, and this paper.

9. REFERENCE

[1] J. Li, R. Topaloglu and S. Ghosh, "Quantum Generative Models for Small Molecule," p. 1, 2021.

[2] V. Hu, "Project Topic B: Deep learning in Drug Discovery," UIUC, 4 2021. [Online]. Available: https://piazza.com/class/kjyow0m1rkd7mf?cid=522.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. WardeFarley, S. Ozair, A. courville and Y. Bengio, "Generative Adversarial Nets," *NIPS,* p. 1, 2014.

[4] L. Ruddigkeit, R. v. Deursen, L. C. Blum and J.-L. Reymond, " Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17," *J. Chem. Inf. Model,* vol. 52, pp. 2864-2875, 2012.

[5] R. Ramakrishnan, P. O. Dral, M. Rupp and O. v. Lilienfeld, "Quantum chemistry structures and properties of 134 kilo molecules," *Scientific Data 1,* vol. 140022, 2014.

[6] Y. Bian and X. Xie, "Generative chemistry: drug discovery with deep learning generative models," *arXiv preprint,* vol. 09000, pp. 20,21, 2008.

[7] C. Grow, G. D, D. Nguyen and G.-W. Wei, "Generative network complex (gnc) for drug discovery," vol. 1910.14650, p. 1, 2019.

[8] A. Kadurin, S. Nikolendo, K. Khrabrov, A. Aliper and A. Zhavoronkov, "druGAN: An Advanced Generative Adversarial Autoencoder Model for de Novo Generation of New Molecules with Desired Molecular Properties in Silico," *molecular pharmaceutics,* p. 1, 2021.

[9] D. C. Nicola and K. Thomas, "MolGAN: An implicit generative model for small molecular graphs," *arXiv,* vol. 1805.11973v1, p. 1, 2018.

[10] A. Tropsha, "Best practices for QSAR model development,validation, and exploitation," *Mol. Inf,* pp. 476-488, 2010.

[11] X. Yao, A. Panaye, J.-P. Doucet, R. Zhang, H. Chen, M. Liu, Z. Hu and B. Fan, " Comparative Study of QSAR/QSPR Correlations Using Support Vector Machines, Radial Basis Function Neural Networks, and Multiple Linear Regression.," *J. Chem. Inf. Comput. Sci.,* vol. 44, pp. 1257-1266, 2004.

[12] V. Svetnik, A. Liaw, C. Tong, J. Culberson, R. Sheridan and B. Feuston, "Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling.," *J. Chem. Inf.Comput. Sci.,* vol. 43, p. 1947, 2003.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory.," *Neural computation,* vol. 9, pp. 1735-1780, 1997.

[14] D. P. Kingma and M. Welling, "An Introduction to Variational Autoencoders," *arXiv,* vol. 1906.02691, p. 1, 2019.

[15] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling.," *arXiv preprint,* vol. 1412.3555, 2014.

[16] A. AlirezaMakhzani, J. Shlens, N. Jaitly, I. Goodfellow and B. Frey, "Adversarial Autoencoders," *arXiv,* vol. 1511.05644, p. 1, 2015.

[17] H. Lee, "GAN Lecture 1 (2018):Introduction," Youtube, 11 May 2018. [Online]. Available: https://www.youtube.com/watch?v=DQNNMiAP5lw&list=PLJV_el3uVTsMq6JEFPW35BCiOQTsoqwNw.

[18] U. E. P. Agency, "SMILES Turorial," U.S. Environmental Protection Agency, 21 2 2016. [Online]. Available: https://archive.epa.gov/med/med_archive_03/web/html/smiles.html#:~:text=What%20is%20SMILES%3F,learn%20a%20handful%20of%20rules..

[19] J. Li, R. Topaloglu and S. Ghosh, "Quantum Generative Models for Small Molecule Drug Discovery," *arXiv,* vol. 0348v1, p. 1, 2021.

[20] Google, "The Generator," Google, 18 4 2021. [Online]. Available: https://developers.google.com/machine-learning/gan/generator.

[21] Google, "The Discriminator," Google, 18 4 2021. [Online]. Available: https://developers.google.com/machine-learning/gan/discriminator.

[22] J. Langr and V. Bok, "Chapter 8. Conditional GAN," in *GANs in Action: Deep learning with Generative Adversarial Networks*, NY, MANNING PUBLICATIONS, 2019, p. 274.

[23] R. Tariq, Make your First GAN with pytorch, 3: 14, 2020, pp. 1-204.

[24] O. Simon and Flickr, "Conditional Generative Adversarial Nets," *arXiv,* vol. 1411.1484v1, p. 1, 2014.

[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative Adversarial Nets," *arXiv,* vol. 1406.2661v1, p. 4, 2014.

[26] R. Ramakrishnan, M. Hartmann, E. Tapavicza and A. v. Lilienfeld, "Electronic Spectra from TDDFT and Machine Learning in Chemical Space," *J.Chem.Phys,* vol. 084111, p. 143, 2015.

[27] U. E. 2. T. &. T. S. Files, "invitrodb_v3. Retrieved," U.S. EPA., 9 2020. [Online]. Available: https://www.epa.gov/chemical-research/exploring-toxcast-data-citing-toxcast-data.

[28] Glambard, "Glambard/Molecules_Dataset_Collection," 17 6 2018. [Online]. Available:

https://github.com/GLambard/Molecules_Dataset_Collection.

[29] L. C. Blum and J.-L. Reymond, "970 Million Druglike Small Molecules for Virtual Screening in the Chemical Universe Database GDB-13," *J.Am.Chem,* vol. 131, pp. 8732-8733, 2009.

[30] C. Solutions, "CUI Solutions," CUI Solutions, 5 2021. [Online]. Available: https://www.chemicaluniverse.com/.

[31] http://quantum-machine.org/, "QM9 Dataset," quantum-machine.org, 05 2021. [Online]. Available: http://quantum-machine.org/datasets/.

[32] cap394, "QM9 - Quantum chemistry structures and properties of 134 kilo molecules," kaggle, 01 01 2021. [Online]. Available: https://www.kaggle.com/rmonge/predicting-molecule-properties-based-on-its-smiles.

[33] rdkit/rdkit, "rdkit/rdkit," rdkit, 05 2021. [Online]. Available: https://github.com/rdkit/rdkit.

[34] rdKit, "RDKit: Open-Source Cheminformatics Software," rdkit, 18 4 2021. [Online]. Available: rdkit.org.

[35] J. B. Esben, "SMILES Enumeration as Data Augmentation for Neural Network Modeling," *CoRR,* vol. abs/1703.07076, 2017.

[36] s.-l. 0.24.2, "sklearn.metrics.roc_auc_score," sklearn.metrics.roc_auc_score.html, 05 2021. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html.

[37] sklearn.metrics.precision_recall_curve, "sklearn.metrics.precision_recall_curve," sklearn.metrics, 05 2021. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_curve.html.

[38] wiki, "F-score," wiki, 05 2021. [Online]. Available: https://en.wikipedia.org/wiki/F-score.

[39] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford and X. Chen, "Improved Techniques for Training GANs," *arXiv,* vol. 1606.03498, p. 6, 2016.