Stacked Denoising Autoencoder (SDA) for Drug-Target Interaction Identification

Chen Guo
cxg451@case.edu

Yunzhou Cao
yxc1426@case.edu

1. Introduction

Although scientists have improved the knowledge of disease, transformation of those advances into therapeutic gains has been far slower than original though **[1]**. High attrition rates, mounting time to bring a new drug into the market and ascending cost demonstrated that money spent on research and development is way higher than the return in the market **[2].**

Drug repositioning aiming to identify new uses of existing drugs outside it's original indentation. This process provides multiple benefits than developing a new drug which includes low failure rates, reduced time and less investment needed. For whole drug repositioning process, there will be three steps: firstly, identify the candidate drug; secondly, evaluate the effectiveness of drug; thirdly, clinical trials. **[2]** Among them, the first step, identifying a candidate drug, is pivotal.

Drug-target interactions (DTIs) is crucial to find new uses of existing drugs in the field of drug repositioning. Traditional approaches including ligand-based, structure-based and text mining approach, ligand-based approach assume that similar ligands tend to have similar biological features. Quantitative Structure-Activity Relationship (QSAR) based on every ligand of targets to scan drugs, hence to identify new DTIs. But most targets do not have enough ligand information. Structure based method (molecular docking) is the second type in drug repositioning. These methods based on the crystallographic structure, search algorithm will search all positions of target and ligand such as different angle or rotation and try to match them. But the 3D shape is not always available, for example, most drugs are matching with membrane protein which do not have 3D shape. Third approach is based on text mining. applications including matching side-effects of drugs and matching keywords in the name of drugs or compounds. But redundancy of those names and hard to use those side-effects in the drug instruction build a barrier to these approach.

Chemical structure based approach brings a light to challenges of those traditional methods.These approach excavate chemical structure of both drugs and targets as well as disease features. Drug and features including gene expression or transcriptomic data, chemical structures, side effects and pathways. Disease features including phenotype, retrospective clinical data such as Electronic health records(EHRs) of patients and symptomatic states **[3]**.

Chemical structure based approach contains two categories: network-based and learning-based approach. Network-based approach construct a drug-target network base on the known drug-target pairs to predict unknown interactions. And Learning method firstly build a classification model in which positive sample consists of interacting drug-target pairs. Negative samples consists of non-interacting drug-target pairs. Using these samples to make classification. But all experimentally validated negative samples are not reported. Using all unknown samples as negative sample will indeed influence the accuracy.

In this project, we use a relatively new learning method to balance that limitations. We combined stacked and denoising into auto-encoders --Stacked Denoising Auto-encoder--to predict unknown DTIs. We also use little unknown samples as negative samples, in this way, the rate of wrong labeled data can be maintained at an extremely low level.

2.Related work

Wen et.al.**[4]** implemented a method that based on deep belief networks to predict unknown drug-target interactions, their results exceeded other state-of art learning based methods. Wang et.al. **[5]** developed a method based on stacked-autoencoder with random forest classifier to find the relationships between drugs and targets. Meriem et.al. **[6]** made prediction by stacked-autoencoder with SVM classifier and find 10 new drug-target pairs. They found their model outperforms other traditional learning model such as SVM, Nearest neighbor and Random forest.

3. Methods

This project has mainly three steps: (i)preprocess the unsupervised, training and testing data; (ii)build the stacked-denoising-autoencoder model; (iii)training the model.

3.1 Data preprocessing

(i) Data source and Drug target interaction space

Previously preprocessed data used in this project were download from a recent publication **[7]** which are extracted from DrugBank database. DrugBank is a bioinformatics and cheminformatics database that combines detailed drug data with comprehensive drug target information. The detail information of the data we use is:The interactions of drugs and targets matrix, in which each column denote a target and each row is a drug. If any drug and target have interaction, then corresponding position in matrix is marked as 1, otherwise 0. This matrix also represent the Drug target space (DTS) which is defined as all possible drug-target pairs (DTPs). In all, there are 5877 drugs and 3348 targets. The dimension of drug-target interaction matrix is 5877 ＊ 3348, thus the DTS has 5877 ＊ 3348(i.e.,19676196) DTPs. Among them, 12674 pairs are positive DTIs(as mentioned before, Drug-Target interactions marked as +1) which have known interaction and the others are unlabeled data (marked as 0).

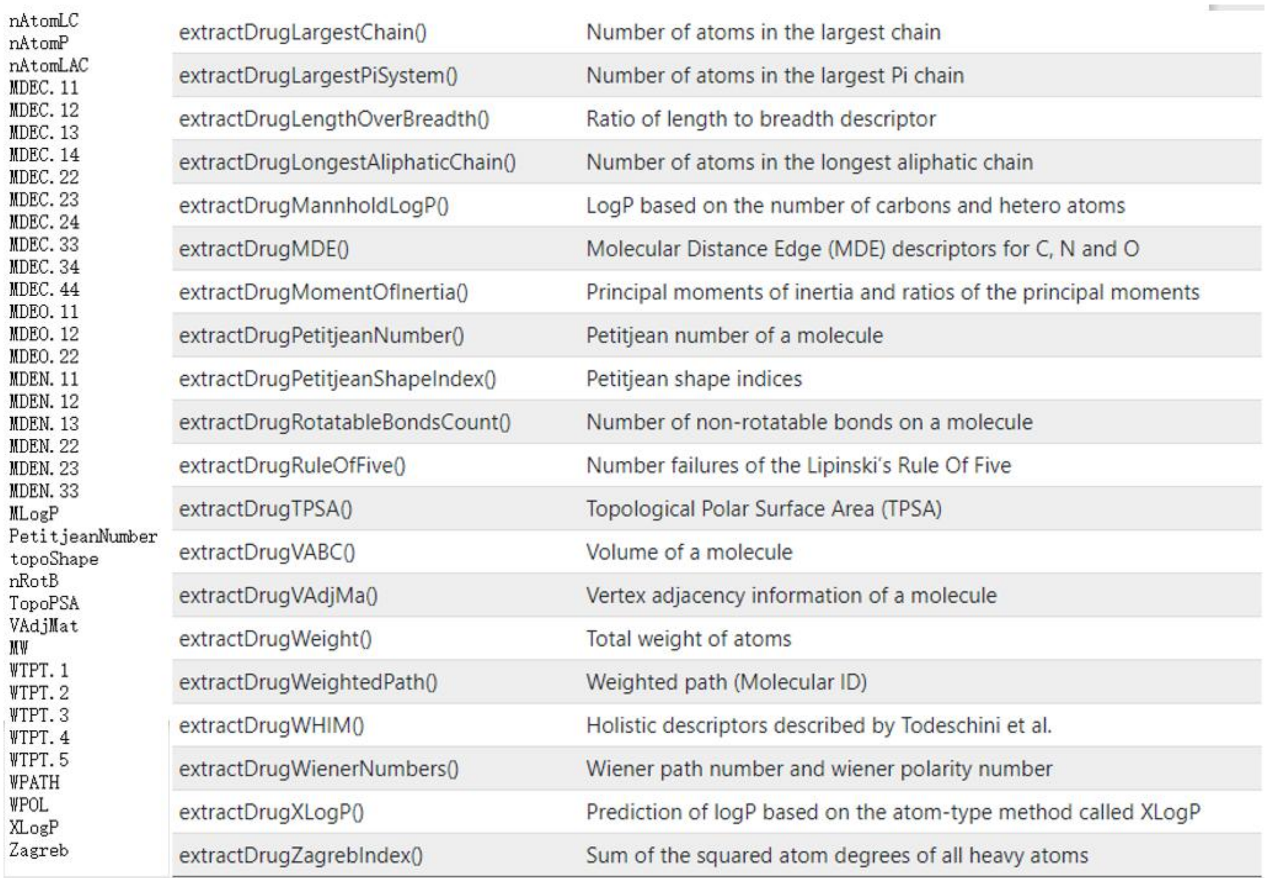
We extract all positive interactions from the drug-target interaction matrix and then form the feature vector, details of the feature vector will be discussed later. Then, as the number of no interaction pairs is much more than the number of interaction pairs, the negative dataset can be randomly selected from the DTS. In this project,we randomly select 12674 drug-target pairs from the DTS except positive data as a negative dataset (marked as 0). Hence, the whole labeled dataset for supervised learning contains 25348 samples. To form the labeled dataset, we build positive dataset and negative dataset separately, then combine them together. After that we generate the label vector by similar method. For label vector, first 12674 dimension is 1 and last 12674 dimension is 0. Then transform them to one-hot for softmax classifier. Finally, we shuffle the dataset and separate 75 percent of data into training data and 25 percent as testing data.

Because we do not use whole unlabeled data to make a prediction, the wrong labeled rate can be constrained at an extremely low level.

The data used for unsupervised learning are randomly selected from the Drug target space. As the whole Drug target space is too large for training (19676196 samples will overflow the runtime and slower the loading data process), we randomly selected 10,000 samples from the Drug target space as unlabeled data.

(ii) Drug-Target repersentation

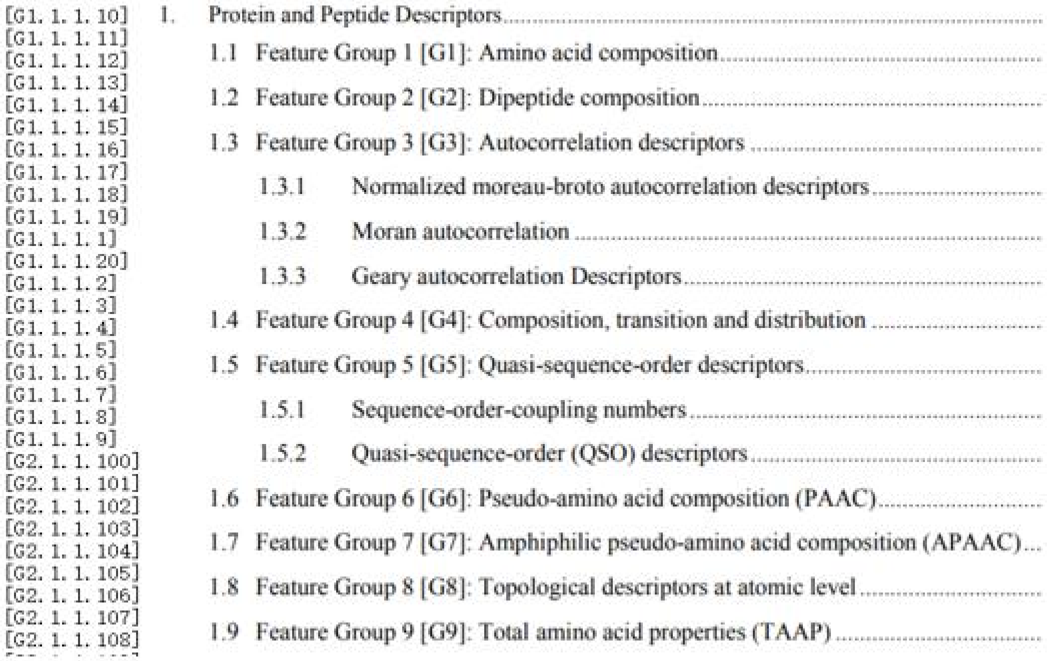
Drugs and targets are represented by sets of feature vectors.These features are classified into two categories: (i) the chemical structure of drugs (molecular fingerprints) and (ii) protein sequence (molecular descriptors). The data for the first part used the Rcpi package **[8]** to calculate drug features. Examples of drug features include constitutional, topological and geometrical descriptors among other molecular properties. The target features were represented by PROFEAT (Protein Feature Server) descriptors **[9][10][11][12]**. The features that have been used to represent targets are descriptors related to amino acid composition, dipeptide composition, autocorrelation, composition, transition, and distribution, quasi-sequence-order, amphiphilic pseudo amino acid composition and total amino acid properties. Thus, it obtained 193 and 1290 features for drugs and targets, respectively. **Fig.1** shows examples of feature name of drugs and corresponding function to calculate them in Rcpi package. Example feature name of target and their description are shown in **Fig. 2**.



nAtomLC
nAtomP
nAtomLAC
MDEC.11
MDEC.12
MDEC.13
MDEC.14
MDEC.22
MDEC.23
MDEC.24
MDEC.33
MDEC.34
MDEC.44
MDEO.11
MDEO.12
MDEO.22
MDEN.11
MDEN.12
MDEN.13
MDEN.22
MDEN.23
MDEN.33
MLogP
PetitjeanNumber
topoShape
nRotB
TopoPSA
VAdjMat
MW
WTPT.1
WTPT.2
WTPT.3
WTPT.4
WTPT.5
WPATH
WPOL
XLogP
Zagreb

extractDrugLargestChain()	Number of atoms in the largest chain
extractDrugLargestPiSystem()	Number of atoms in the largest Pi chain
extractDrugLengthOverBreadth()	Ratio of length to breadth descriptor
extractDrugLongestAliphaticChain()	Number of atoms in the longest aliphatic chain
extractDrugMannholdLogP()	LogP based on the number of carbons and hetero atoms
extractDrugMDE()	Molecular Distance Edge (MDE) descriptors for C, N and O
extractDrugMomentOfInertia()	Principal moments of inertia and ratios of the principal moments
extractDrugPetitjeanNumber()	Petitjean number of a molecule
extractDrugPetitjeanShapeIndex()	Petitjean shape indices
extractDrugRotatableBondsCount()	Number of non-rotatable bonds on a molecule
extractDrugRuleOfFive()	Number failures of the Lipinski's Rule Of Five
extractDrugTPSA()	Topological Polar Surface Area (TPSA)
extractDrugVABC()	Volume of a molecule
extractDrugVAdjMa()	Vertex adjacency information of a molecule
extractDrugWeight()	Total weight of atoms
extractDrugWeightedPath()	Weighted path (Molecular ID)
extractDrugWHIM()	Holistic descriptors described by Todeschini et al.
extractDrugWienerNumbers()	Wiener path number and wiener polarity number
extractDrugXLogP()	Prediction of logP based on the atom-type method called XLogP
extractDrugZagrebIndex()	Sum of the squared atom degrees of all heavy atoms

Fig.1. Last 38 features of drug and the description of them (For example, nRotB means Rotatable Bonds Count) **[7][8]**. Note the features and functions are not in the same order and as Rcpi package update recently, several features are no longer available.



[G1.1.1.10]
[G1.1.1.11]
[G1.1.1.12]
[G1.1.1.13]
[G1.1.1.14]
[G1.1.1.15]
[G1.1.1.16]
[G1.1.1.17]
[G1.1.1.18]
[G1.1.1.19]
[G1.1.1.1]
[G1.1.1.20]
[G1.1.1.2]
[G1.1.1.3]
[G1.1.1.4]
[G1.1.1.5]
[G1.1.1.6]
[G1.1.1.7]
[G1.1.1.8]
[G1.1.1.9]
[G2.1.1.100]
[G2.1.1.101]
[G2.1.1.102]
[G2.1.1.103]
[G2.1.1.104]
[G2.1.1.105]
[G2.1.1.106]
[G2.1.1.107]
[G2.1.1.108]

Fig.2. Example feature name of target and the description all features (For example, from [G1.1.1.10] to [G1.1.1.10] are Feature Group1 [G1], the amino acid composition descriptors) **[7][9][10][11][12]**.

After we collected the features, we need to generate the drug-target pairs as our input data, each drug-target pair is represented by feature vectors that are formed by concatenating the feature vectors of the corresponding drug and target involved. For instance, a drug-target pair is represented by the feature vector:

$$[d1,\ d2,\ d3,,d193,\ t1,\ t2,\ t3,,t1290]$$

where [*d*1, *d*2,..., *d*193] is the feature vector corresponding to drug *d*, and [*t*1, *t*2,..., *t*1290] is the feature vector corresponding to target *t* **[6]**. In detail, for labeled data, we scan all the position at drug-target matrix to obtain the index of row and column. Then use the indexes to get the corresponding feature in drug feature vector and target feature vector respectively. The dimension 0 of input data is 1483. For unlabeled data, we randomly generate 10,000 indexes, and get the drug feature vector and target feature vector, then concatenate them together. The dimension 0 of unlabeled data for unsupervised learning is also 1483. Labeled data preprocessing concept is illustrated in **Fig.3**.

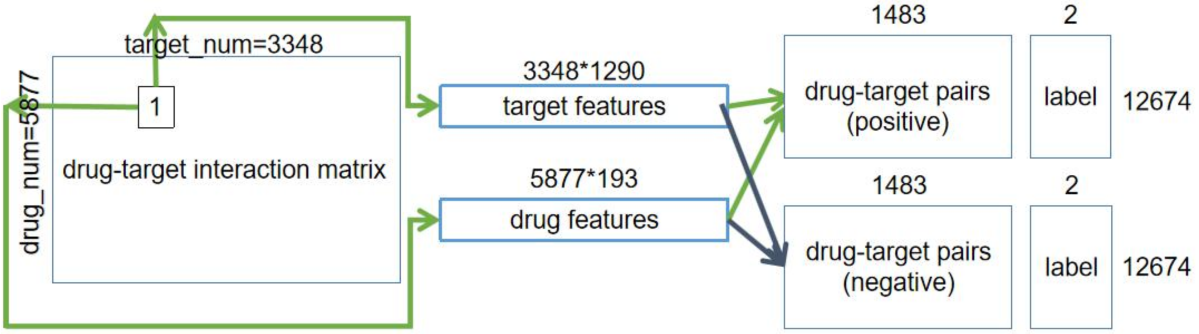


Fig.3. Labeled data preprocessing concept, negative drug-target pairs are randomly selected from drug-target space except positive pairs.

3.2 Model

The model we built contains two parts: **(i)**Stacked Denoising autoencoder(SDA) and **(ii)** a fined-tuning deep neural network.

To understand SDA, we need to know Stacked Autoencoder(SAE) and Denoising Autoencoder(DAE) first since SDA is the combination of these two types of autoencoders yet before that we need to know the basic autoencoder.

(i) Basic Autoencoder

An autoencoder is a neural network with the same input and learning goals and it consists of an encoder and a decoder. The encoder takes the input vector $x \in [0,1]^d$ and maps it to a hidden representation$y \in [0,1]^{d''}$ **[13]**. Then do a deterministic mapping so that $y = f(x) = a(Wx + b)$.W is the dxd' weight matrix and b is a bias vector. In this way,the resulting latent representation y is reconstructed by the decoder to a vector $z \in [0,1]^d$,which can be written as:$z = h(y) = s(W'y + b')$. At each training step x is mapped to a corresponding y. We adjust the two parameters to minimize the reconstruction error. The loss function L can be written with a classical squared error:$L(x,z) = ||x - z||^2$. However, there is another loss caused by the interpretation of x and z, when they are both bit vectors or vectors of Bernoullis probabilities and this is known as reconstruction cross-entropy:

$$\begin{aligned} L_H(x,z) &= H(B_x||B_z) \\ &= -\sum_{k=1}^{d} \quad [x_k log z_k + (1 - x_k) log(1 - z_k)] \end{aligned} \qquad (1)$$

With the loss function,the average reconstruction error is:

$$\theta,\theta' = argmin\ \frac{1}{n}\sum_{i=1}^{n} \quad L(x^i,z^i)$$

$$= argmin\ \frac{1}{n}\sum_{i=1}^{n} \quad L(x^i,h(f(x^i)))$$ (2)$\theta =$ $\{W,b\},\theta' = \{W',b'\}$

(ii) Denoising Autoencoder(DAE)

DAE is one type of improved autoencoder also produced by Bengio et al in 2008 **[13]**. DAE adds random noise to the input data to avoid over-fitting. The idea behind denoising autoencoders is not hard to understand. In order to make the hidden layer to extract more robust features and prevent it from simply learning the identity at the same time, we train the autoencoder to reconstruct the input from a corrupted version. The denoising autoencoder is a stochastic version of the autoencoder. Intuitively, a denoising auto-encoder does two things: try to encode the input (i.e.,preserve the information about the input), and try to undo the effect of a corruption process applied to the input of the auto-encoder. The latter can only be done by capturing the statistically dependencies between the inputs. **Fig.4** shows the flowchart of DAE.

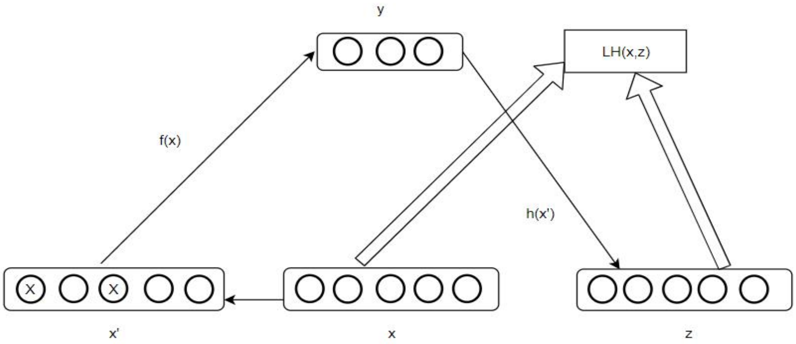


Fig.4 Flowchart of a Denoising Autoencoder

So, for DAE,the hidden representation y and reconstructed input z can be mathematically written as the same as that for a traditional autoencoder.But the input x should be corrupted to $\hat{x}$. We can assume it is stochastic mapped:
$\hat{x}{\sim}q_D(\hat{x}|x)$.That to say,for input x, a certain number of it is corrupted by forced to be 0 while the rest remain unchanged.Assume the joint distribution:

$$q^0(x,\hat{x},y) = q^0(x)q_D(\hat{x}|x)\alpha_{f(\hat{x})}(y) \qquad (3)$$

$\alpha_i(j)$is function which performs like XOR(i.e.,put to 0 when $i \neq j$).
Then according to Eq.1,Eq.2 and Eq.3,the objective function is now minimized stochastic descent:

$$argmin\ E_{q^0}(x,\hat{x})[L_H(x,h(f(\hat{x})))] \qquad (4)$$

(iii) Stacked Autoencoder(SAE)

SAE is a widely used deep learning model,first produced by Bengio et al in 2007 **[14]**. SAE plays a critical role in unsupervised learning.Generally,it is optimized by greedy layer-wise training.

SAE has multiple layers of autoencoders in which the output of each layer is connected to the input of the next layer **[15]**. For a normal autoencoder,though features are extracted many times, the objective function is computed only once. With SAE, such drawback can be eliminated.Because we know that typical neural network is a supervised process. Apart from input x,we also need label y. Then we feed the input x to the neural network and minimize the difference value between the output and label Y. However for a traditional autoencoder, the whole process is not supervised,i.e,we do not need label y. The optimization goal in training is to minimize the difference between input and output.The basic idea is that input data represents some information and these information in represented in certain dimensions, then the hidden layer decreases the dimensions and outputs results close to initial data. During the process, the dimension of data changes, but the information represented is not lost. Following this idea, we can realize dimensionality reduction in an unsupervised way with a simple one-hidden-layer neural network.

(iv) Stacked Denoising autoencoder(SDAE)

Combining these two different types of autoencoders together, we can have SDA. Its mathematical theory is not much different from the combination of SAE and DAE. However,note that the training part is different, which will be discussed in the next section.

3.3 Training

The training part has two parts:the layer-wise unsupervised pre-training process with stacked Denoising Autoencoders and the supervised fine-tuning of the deep neural network.

At the present work, we implemented two stacked denoising auto-encoders for the layer-wise unsupervised pre-training process in order to obtain the lower and hidden representation of drug-target pairs data. The optimization algorithm we used for this model is Adam optimizer which can adjust learning rate automatically, the initial learning rate is 0.0001. The input and output layer of first autoencoder has 1483 neural units and fully connect to the bottleneck with 400 neural units. The data corruption rate of the first autoencoder in 30 percent, which means 30 percent of input data were replaced by gaussian noise. The input and output of second autoencoder is 400 and fully connect to the bottleneck of 100 units. The data corruption rate of the second autoencoder is 25 percent.

The training strategy of unsupervised pre-train stage of autoencoder is layer-wised performed. We assign the pre-trained autoencoder weights to multilayer perceptron (MLP) in order to use the compressed data representation of drug-target pairs. This deep neural network has the following structure: 1483-400-100-2. This deep neural network take input features with 1483 dimension and the output space is 2 dimensional. The first hidden layer has 1000 units and the second hidden layer has 600 units. Then connect the second hidden layer to softmax classifier. This concept of pre-train strategy is demonstrated in **Fig.5.** and **Fig.3.** The orange and blue lines refer to the weights of encoder of the first and the second autoencoder.

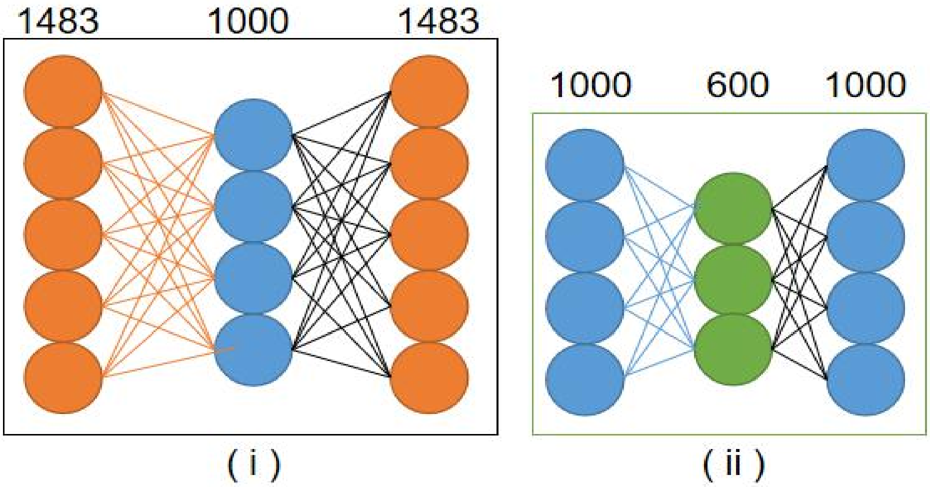


Fig.5. The structure of the first autoencoder and the second autoencoder. **(i)** 1483-1000-1483. **(ii)** 1000-600-1000.

For layer-wise training, the hidden layer is trained single layer at each time. We trained the first autoencoder(AE1) with structure 1483-1000-1483 at the beginning, for 600 epochs. After finishing the training of AE1, We took the hidden representation of AE1 which are weights illustrated in orange as the input of the second autoencoder(AE2). During the training stage for AE2, the weights of AE1 are fixed. The epoch for training AE2 is 600. Then we used the representation of stacked-autoencoder to train the softmax classifier. Similarly, the weights of AE1 and AE2 are fixed. The training epoch for softmax is 200.

Fig.6 shows the supervised-trained deep neural network which utilize the previous unsupervised-trained information. This neural network contains the trained weights of previous unsupervised learning. Thus, this supervised learning stage is called finetune. The aim of finetune is to optimize the weights of neural network to classify the drug-target pairs and minimize the prediction cost by supervised learning. The output layer of this model contains two output neurals, each of them represent the probability of an input drug-target pairs to be interactive. The output of the model is compared with the ground truth label which is represented in one-hot formation.The output is obtained applying a softmax function. Softmax functions normalize the output distribution, Thus, the outputs of the model represent complementary probabilities (i.e. sum of probabilities of being positive drug-target interaction or negative drug-target interaction is 1; for example, an output of probability of being positive drug-target interaction is 80%, and of being drug-target interaction is 20%).

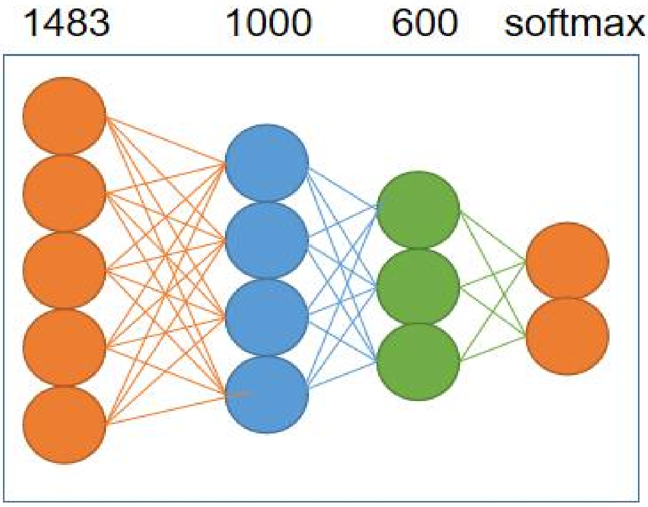


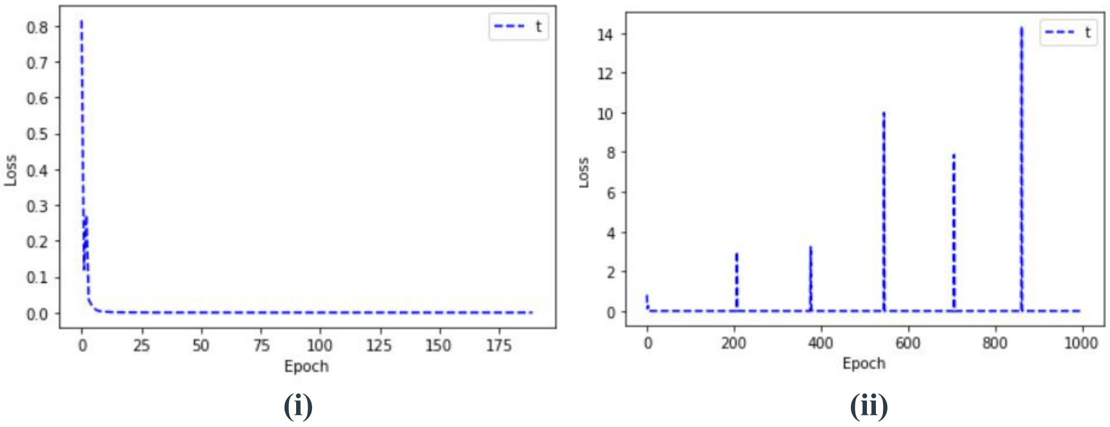
Fig.6. Migrate the value of weights from previous unsupervised learning stage. The orange and blue weights are transferred from encoders of AE1 and AE2.

4. Result

The result contains two parts. First part demonstrates an interesting finding on loss value. Second part illustrates the comparison among different models.

4.1 Peaks in loss value.

We plotted the loss value along the Epoch, as we can observe from **Fig.7**, the other part of loss value is decreasing (For example, **(i)** shows the first 200 epoch to verify the loss value is truly decreasing) , but some parts are peaking. Thus it is very likely that there are wrong labeled data in the training set. Thus it is very likely there are still interactions we don't know.



(i) **(ii)**

Fig.7. Peaks in loss value.

4.2 Model Performance

We compared our model with other models. AUC, accuracy and sensitivity score are used in these experiments. For DNN, we use the same structure as our finetune model which has structure: 1483-1000-600-2, with 2400 epoches for training.

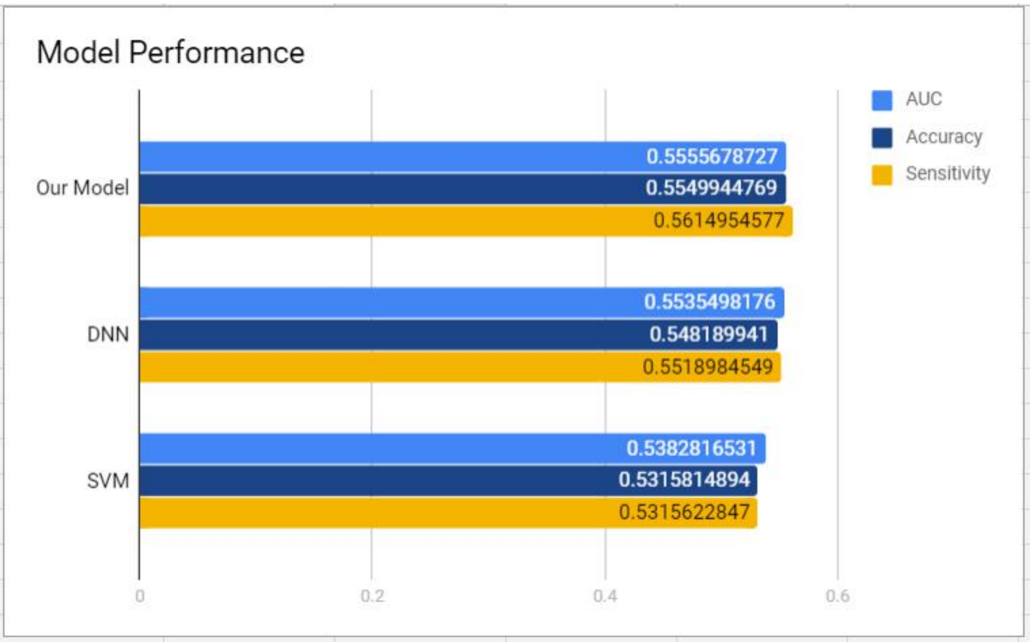


Fig.8. Comparison of performance with DNN and SVM

5. Conclusion and Future Work

In this project, we utilized a relatively new method to predict Drug-Target Interactions by combining stacked and denoising autoencoders to build Stacked Denoising Autoencoders. Experiments show that our method has a slightly better performance than some traditional methods like DNN and SVM. However,the total accuracy is not very satisfying, so definitely there are improvements to be made. What's more, we think we can scale up this project by using powerful tool like H2O Sparkling-water to handle big data to improve performance. From the loss value, we believe there must be other unknown drug-target pairs, thus the future for drug repositioning is still promising.

6. References

1. Scannell, J., Blanckley, A., Boldon, H. *et al.* Diagnosing the decline in pharmaceutical R&D efficiency. *Nat Rev Drug Discov* 11, 191–200 (2012) doi:10.1038/nrd3681
2. Pushpakom, S., Iorio, F., Eyers, P. *et al.* Drug repurposing: progress, challenges and recommendations. *Nat Rev Drug Discov* 18, 41–58 (2019) doi:10.1038/nrd.2018.168
3. Yamanishi Y. (2013) Chemogenomic Approaches to Infer Drug–Target Interaction Networks. In: Mamitsuka H., DeLisi C., Kanehisa M. (eds) Data Mining for Systems Biology. Methods in Molecular Biology (Methods and Protocols), vol 939. Humana Press, Totowa, NJ
4. Ming Wen, Zhimin Zhang, Shaoyu Niu, Haozhi Sha, Ruihan Yang, Yonghuan Yun, and Hongmei Lu Journal of Proteome Research 2017 16 (4), 1401-1409
5. Wang, L., You, Z., Chen, X., Xia, S., Liu, F., Yan, X., Zhou, Y., & Song, K. (2017). A Computational-Based Method for Predicting Drug-Target Interactions by Using Stacked Autoencoder Deep Neural Network. Journal of computational biology : a journal of computational molecular cell biology, 25 3, 361-373 .
6. Bahi M., Batouche M. (2018) Drug-Target Interaction Prediction in Drug Repositioning Based on Deep Semi-Supervised Learning. In: Amine A., Mouhoub M., Ait Mohamed O., Djebbar B. (eds) Computational Intelligence and Its Applications. CIIA 2018. IFIP Advances in Information and Communication Technology, vol 522. Springer, Cham
7. Ezzat, A., Wu, M., Li, X.L., Kwoh, C.K.: Drug-target interaction prediction via class imbalance-aware ensemble learning. BMC Bioinformatics 17(19), 509 (2016)
8. Dong-Sheng Cao, Nan Xiao, Qing-Song Xu, Alex F. Chen, Rcpi: R/Bioconductor package to generate various descriptors of proteins, compounds and their interactions, Bioinformatics, Volume 31, Issue 2, 15 January 2015, Pages 279–281, https://doi.org/10.1093/bioinformatics/btu624

9. P. Zhang, L. Tao, X. Zeng, C. Qin, S.Y. Chen, F. Zhu, Z.R. Li, Y.Y. Jiang, W.P. Chen, Y.Z. Chen. A protein network descriptor server and its use in studying protein, disease, metabolic and drug targeted networks. Brief Bioinform. pii: bbw071 (2016).

10. P. Zhang, L. Tao, X. Zeng, C. Qin, S.Y. Chen, F. Zhu, S.Y. Yang, Z.R. Li, W.P. Chen, Y.Z. Chen. PROFEAT Update: A Protein Features Web Server with Added Facility to Compute Network Descriptors for Studying Omics-Derived Networks. J Mol Biol. pii:S0022-2836(16)30428-4. (2016).

11. H.B. Rao, F. Zhu, G.B. Yang, Z.R. Li, and Y.Z. Chen. Update of PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. Nucleic Acids Res. Jul 1, 2011; 39(Web Server issue):W385-90.

12. Z.R. Li, H.H. Lin, L.Y. Han, L. Jiang, X. Chen, and Y.Z. Chen. PROFEAT: A Web Server for Computing Structural and Physicochemical Features of Proteins and Peptides from Amino Acid Sequence. Nucleic Acids Res. Jul 1, 2006; 34(Web Server issue):W32-7.

13. Bengio.Y.,Lamblin.P.,Popovici.D.,Larochelle.H.:Extracting and Composing Robust Features(2008)ICML '08 Proceedings of the 25th international conference on Machine learning Pages 1096-1103

14. Bengio.Y.,Lamblin.P.,Popovici.D.,Larochelle.H.:Greedy Layer-Wise Training of Deep Networks (2007). NIPS'06 Proceedings of the 19th International Conference on Neural Information Processing Systems Pages 153-160

15. Zhou, Y., Arpit, D., Nwogu, I., Govindaraju, V.: Is joint training better for deep auto-encoders? (2014). arXiv prepri nt arXiv:1405.1380