Machine Learning Bioinformatics

Final Report

Ruchik D, Sumanth G, Harsha R, Siddhant M

INTRODUCTION:

Drug response prediction is a well-studied problem in which the molecular profile of a given sample is used to predict the effect of a given drug on that sample. Effective solutions to this problem hold the key for precision medicine. In cancer research, genomic data from cell lines are often utilized as features to develop machine learning models predictive of drug response.

Recent advances in artificial intelligence, including machine learning and deep learning, have generated considerable interest in solving classic biomedical problems. Whereas popular applications include disease diagnosis from biomedical images, interpretation of electronic medical records, etc. Owing to the significant molecular heterogeneity observed across tumors, there are often many different molecular features and feature combinations that can lead a model to predict drug response. In this study, we aim to use Deep learning models improve Predicting the response of cancer cell to drugs.

METHODOLOGY:

Machine Learning and Deep Learning models in recent times have a promising impact on drug response prediction, but due to lack of interpretation of model response those models' proposed drugs enters the clinical trials are failing. To address these issues DrugCell, an interpretable deep learning model is proposed, The cell-subsystem is relating with the drug structure to predict the response.

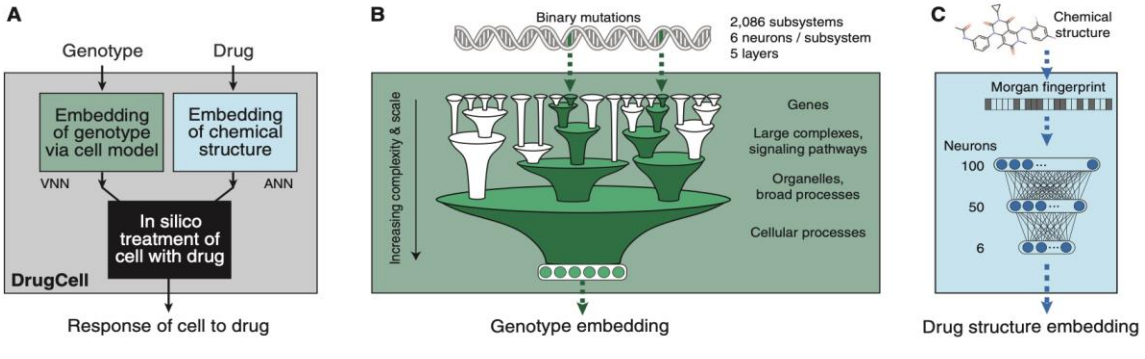

Fig 1: DrugCell structure [1]

DrugCell architecture as is shown in Fig 1, which takes the genotype and drug embedding which is encoded using the Visible neural Network (VNN) and artificial neural network (ANN) respectively. The visible neural network called Dcell which is developed by imitating the structure of simple eukaryotic cell, *Saccharo- myces cerevisiae.* A large hierarchy of known putative molecular components and pathways are modeled using deep neural network architecture. By using interpretable network architecture it's easy to give reasoning for drug interaction response. For representing chemical structure of drug to embedding the artificial neural network is used on the Morgan fingerprint chemical representation.

Drugcell is able to achieve interpretability and better accuracy to predict drug response. More meaningful representations of cell mutation and drug chemical structure in terms of embedding can help to increase the accuracy. Auto encoders are one of those representations which can able to encode the features of similar components in one place. Using those representations can help in increasing predictive model performance. So, we have chosen to represent the genotype and drug embeddings by training different autoencoders. The Auto-encoder architecture is discussed below.

AUTOENCODERS:

We are using Autoencoders which is a type of feedforward neural network where the input is same as the output. It learns the data encodings by compressing the input into a lower-dimensional space and then reconstructs the output from this representation. The goal of this Autoencoder is to learn the lower dimensional representation (encoding) of the genomic and the drug data to capture the most important parts (good feature extraction).

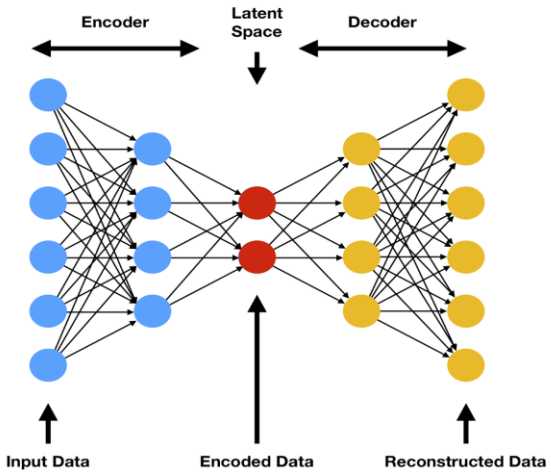

Fig 2: AutoEncoder Architecture

The network architecture for the autoencoders was a deep neural network with 4 hidden layers. We train the autoencoder model on the drug fingerprint and the cell to mutation dataset to encode our dataset.

Here is the snapshot of the network architecture of the Autoencoder:

```
AeDrug(
  (encoder): Sequential(
    (0): Linear(in_features=2048, out_features=512, bias=True)
    (1): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Dropout(p=0.2, inplace=False)
    (3): ReLU()
    (4): Linear(in_features=512, out_features=256, bias=True)
    (5): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (6): Dropout(p=0.2, inplace=False)
    (7): ReLU()
    (8): Linear(in_features=256, out_features=128, bias=True)
  )
  (decoder): Sequential(
    (0): Linear(in_features=128, out_features=256, bias=True)
    (1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): Dropout(p=0.2, inplace=False)
    (3): ReLU()
    (4): Linear(in_features=256, out_features=512, bias=True)
    (5): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (6): Dropout(p=0.2, inplace=False)
    (7): ReLU()
    (8): Linear(in_features=512, out_features=2048, bias=True)
  )
)
```

Fig 3: Defined Autoencoder Architecture

RESTRICTED BOLTZMANN MACHINES:

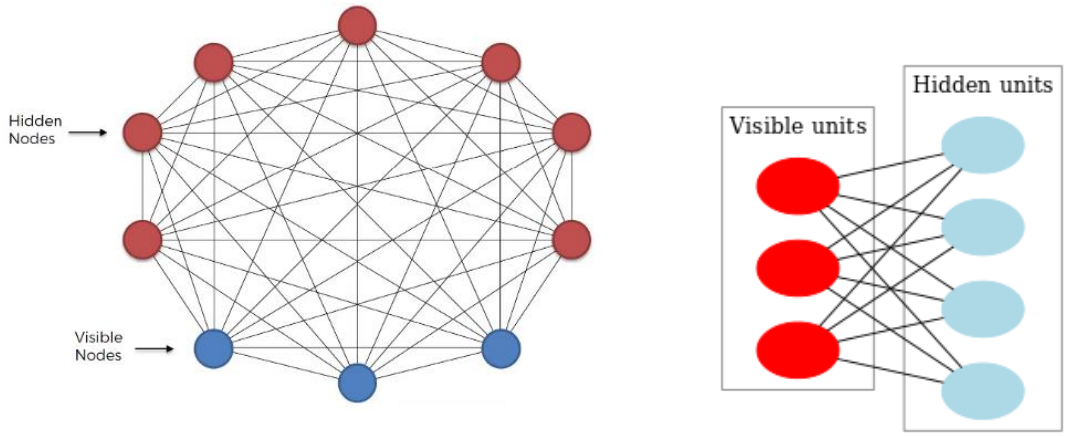

Fig 4: left) Boltzman machine Right) Restricted Boltzmann machine

Boltzmann Machines is an unsupervised DL model in which every node is connected to every other node, even the visible neurons connected to each other and hidden neurons also connected to each other. The training data is fed into the Boltzmann Machine and the weights of the system are adjusted accordingly. Boltzmann machines help us understand abnormalities by learning about the working of the system in normal conditions. What makes RBMs different from Boltzmann machines is that visible nodes aren't connected to each other, and hidden nodes aren't connected with each other.

The vanilla Autoencoder lower dimensional representation mainly depends upon the initial weight initialization, most of the complex model encoding leads to stuck in local optimum, which in turn will not give best representation. To overcome this problem RBM initialized weights are used for Autoencoder.

In RBM, the probability distribution defined by energy of network is matched with the marginal probability of visible units. For example, in input we start with 3008 nodes in visible layer, and we encode it 3008 → 1024 → 512 → 256 → 128 (output). Each pair of network layers we will train a RBM and store the weights. For example, 3008 ---> 1024 layers first we will train the RBM to represent the probability distribution of visible 3008 vectors in the 1024 hidden latent space and store the weights). Initialize the stored weights to the encoder of the Auto-Encoder network to represent the lower dimensional embedding. Since the lower dimensional embedding is continuous but the RBM only provides the binary hidden representation using Gibbs sampling. So, to provide the continuous representation of latent vector the Gibbs sampling in bottleneck layer RBM weight training is replaced by gaussian sampling.

EXPERIMENTAL SETUP:

- The datasets used in the project undertaken are obtained from following databases:
    - Cancer Therapeutics Response Portal (CTRP) v2: It was developed by the Center for the Science of Therapeutics and contains several hundreds of thousands of drug dose-response curves
    - Genomics of Drug Sensitivity in Cancer (GDSC): It is the largest public resource for information on drug sensitivity in cancer cells and molecular markers of drug response.
- The combined dataset consisted of 509,294 cell line drug pairs, covering 684 drugs and 1,235 cell lines.
- We train the Autoencoder on this combined dataset.
- The train dataset consists of three columns which are the drug fingerprints, gene-ids and the drug response value.
- Using this Autoencoder we get the encoded embeddings of the train data.
- Using these encodings and the drug response column we run various machine learning models.
- The Machine Learning Models we used are:
    - Support Vector Machine Regressor
    - Linear Regression
    - Random Forest Regressor
    - Multi-Layer Perceptron

- The different models are evaluated using the Pearson correlation, which is calculated between predicted and observed values. The higher the correlation, the higher is the prediction accuracies for various drugs. This is further elaborated in the results section.

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

r = correlation coefficient

EXPERIMENTAL RESULTS:

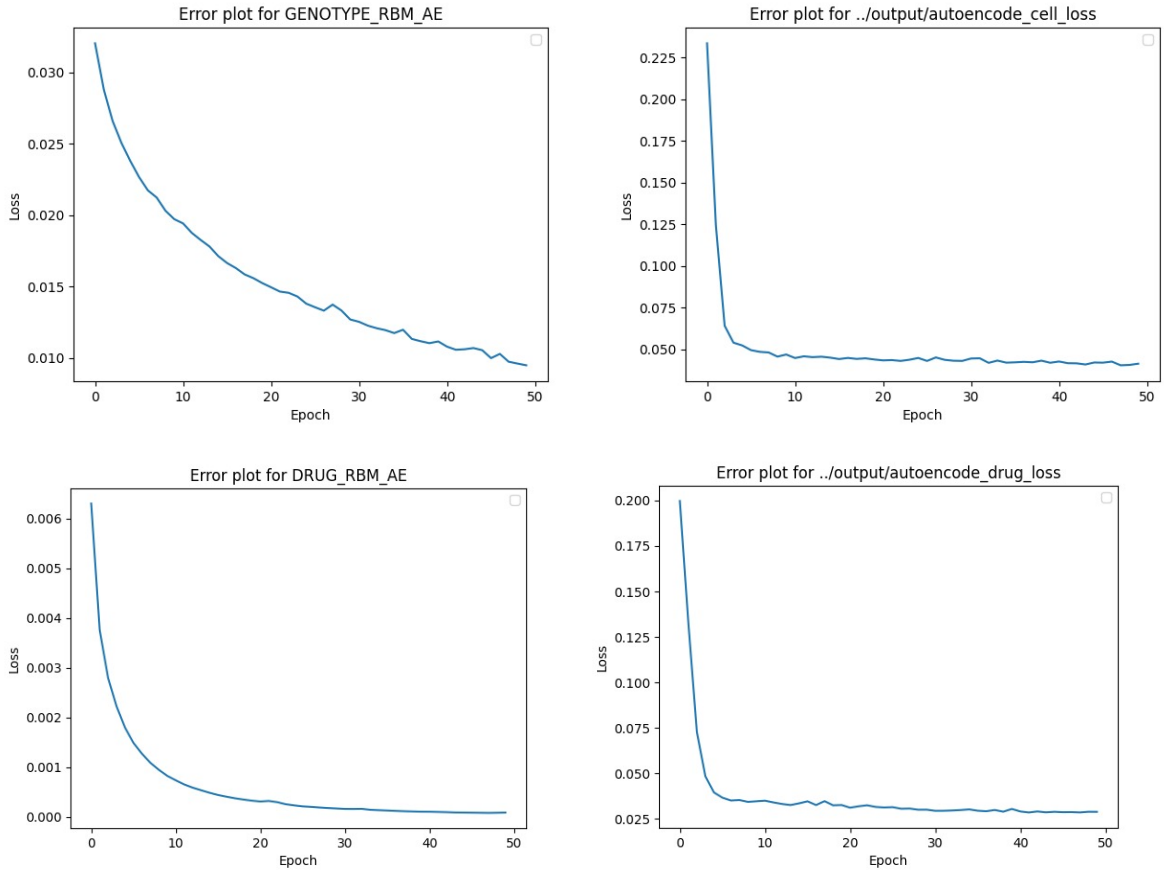

Fig 5: Training Error curves left) RBM based AE right) Simple AutoEncoder.

By observing from fig 5, error loss on the Autoencoder on the right starts off with a higher value as we initialize the weights using the Xavier weight initialization. At the same time when we compare it with the Restricted Boltzmann Machine (RBM) based Autoencoders we see that we start off with a lower loss as we have a good set of pre-trained weights. So, we start off with near optimal weights for every layer and in turn begin with a lower loss which decreases as the epochs increase.

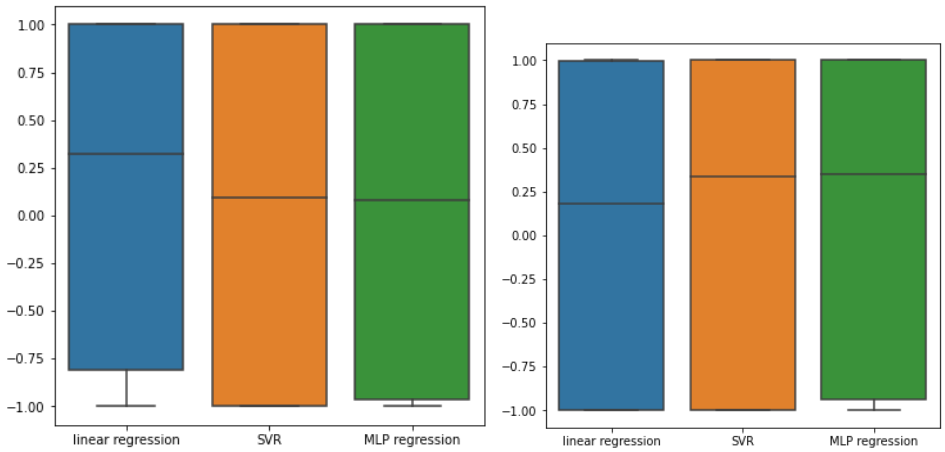

Fig 6: Box plots for Drug response pearson coefficient values between predicted and Actual drug responses using different Machine learning models.

The above Box plots from fig 6 interprets the median drug response of pearson coefficient rho for all drugs in test dataset. The DrugCell achieved the rho value of 0.35 as the median, in par with drug cell the proposed RBM based Autoencoder is also able to provide 0.36 as the median pearson coefficient whereas, simple auto encoder is not able to achieve that score. The percentage of drug having the more than 0.5 is 36-38% for RBM based AE than the DrugCell which is 30% which is an improvement from the drugcell.

MODELS	**AUTOENCODER** **PEARSON CORRELATION**	**RBM-AE** **PEARSON CORRELATION**
SVR	0.60	0.79
LINEAR REGRESSION	0.45	0.45
RANDOM FOREST	0.72	0.69
MLP	0.45	0.62

Table 1: Pearson correlation of different machine learning models for AE and RBM based AE as base embeddings.

Observations:-

- We got the highest pearson correlation in the SVR model using the RBM AE.
- Random Forest model gives a good pearson correlation between both.
- After we get the encodings on the train data, we run the ML models to get the predicted drug response value.
- We are using the Pearson Correlation as our evaluation metric. The Pearson correlation measures the strength of the linear relationship between two variables. It has a value between -1 to 1, with a value of -1 meaning a total negative linear correlation, 0 being no correlation, and + 1 meaning a total positive correlation.
- From the table 1, we can see that the Pearson correlation for the RBM AE is comparatively higher. One main reason could be because we are using the RBM pre-trained weights.

DISCUSSION:

As anticipated the meaningful representation of data in lower dimensional increases the drug response prediction as per observation of results. Even though the better performance is achieved by the RBM based Autoencoder due to lack of interpretability the model may not succeed in the drug clinical trials. In future direction incorporating encoding the features using the VNN network in the RBM based Autoencoder can help in better interpretability and better representation.

REFERENCES:

- B. M. Kuenzi et al., "Predicting Drug Response and Synergy Using a Deep Learning Model of Human Cancer Cells", Vol. 38, Issue 5, pp. 672-684, November 2020, doi: https://doi.org/10.1016/j.ccell.2020.09.014
- G. Adam et al., "Machine learning approaches to drug response prediction: challenges and recent progress", Vol 4, Issue 19, June 2020, doi: https://doi.org/10.1038/s41698-020-0122-1
- V. Dumoulin et al., "Adversarially Learned Inference", ICLR, February 2017
- Cancer Therapeutics Response Portal (CTRP) v2, https://pharmacodb.pmgenomics.ca/datasets/2
- Genomics of Drug Sensitivity in Cancer, https://www.cancerrxgene.org/downloads/bulk_download
- Image: https://www.compthree.com/blog/autoencoder/
- Seashore-Ludlow et al., 2015; Yang et al., 2013