Introduction to Machine Learning Final Report
Mechanisms of Action (MoA) Prediction

Cheng Hsun Chang
Tsing Hua University
Interdisciplinary Program of
Electrical Engineering and Computer Science
Student ID: 107060008
Email: didwdidw0309@gmail.com

Shao Ting Chung
Tsing Hua University
Computer Science
Student ID: 107062231
Email: timclock9324@gmail.com

An Yan Chang
Tsing Hua University
Life Science
Student ID: 107081028
Email: stanleychang0914110366@gmail.com

Hung Chih Chen
Tsing Hua University
Interdisciplinary Program of
Electrical Engineering and Computer Science
Student ID: 107060015
Email: h881109@gmail.com

***Abstract*—With the advent of more powerful technologies nowadays, drug discovery has changed from the serendipitous approaches of the past to a more targeted model based on an understanding of the underlying biological mechanism of a disease.**

Scientists determine the MoAs of a new drug by treating a sample of human cells with the drug and then analyze the cellular responses with algorithms that search for similarity to known patterns in large genomic databases, such as libraries of gene expression or cell viability patterns of drugs with known MoAs. And we are eager to help the progress of drug developing by finding the certain algorithm.

1. Introduction

Scientists derived drugs from natural products or were inspired by traditional remedies in the past. For example, paracetamol, a very common drug known in the US as acetaminophen, were put into clinical use decades before the biological mechanisms driving their pharmacological activities were understood.

Scientists seek to identify a protein target associated with a disease and develop a molecule that can modulate that protein target. As a shorthand to describe the biological activity of a given molecule, scientists assign a label referred to as mechanism-of-action, or MoA for short.

2. Methods

2.1. Preprocessing

Before we can start training our model, the first step is doubtlessly to preprocess the data.

2.1.1. Drop unnecessary columns. There are 771 genes in total, and after experiments, we consider 74 of them are not helpful, or may even be harmful, to our task. Thus, we decide to drop these columns by making a drop list to indicate which columns as long as their labels to drop. The operation is done inplace and return none.

2.1.2. Principal component analysis. The dimension of the data is still so tremendous after some columns have been dropped, and therefore it is quite essential to reduce it. We employ PCA to reduce the loss of data as much as possible while decreasing the dimension, where genes and cells are used separately to train PCA. In this part, we decide to decrease the dimension of genes to 80 and the dimension of cells to 10, according to our experiments.

2.2. Overfitting

Due to the large amount of parameters in practice, the problems of overfitting emerge during the process of traning. Below, we use to primary ways that we combat overfitting.

2.2.1. Data Augmentation. We employ a theorem called “Perturbation Theorem” to augment our data. Perturbation theory comprises mathematical methods for finding an approximate solution to a problem, by starting from the exact solution of a related, simpler problem. In the case of MoA prediction, we use the theorem to transform the input of the origional data to sets of approxamately problem, while the target datas remain unchanged. This process can be expressed as a ”effective range” of change to a known MoA, which is scientifically been proved. After the transformation, we obtain the data of twice the size of the origional. During the process of training, we augment the data in each epoch so that the augmented data are different during each iteration of training. Using this method, we significantly enhace the robustness of our model.

2.2.2. Dropout. Combining the predictions of many different models is a very successful way to reduce test errors, but it is truelly difficult for us to design several sophisicated model in just a few days. Therefore, we adopt the method "Dropout", which is a very efficient version of model combination that only costs about a factor of two during training. Dropout consists of setting to zero the output of each hidden neuron with probability number. Tuning this probability number is tricky, we need to consistently try different value to deternmine which parameter peforms well. We can neither use a too large parameter such that the overfitting issue emerges nor a too small parameter such that the result can not converges. After the trying, we found out that the drop out rate be 0.26 is optimal solution in this case.

2.3. Model

We aimed at defining the correlative responses of the drug samples, so how to design a model that clarifies the relationship bothered us. With the paper we found[1] questioning the regulation between chemical interactions of transcription factors and genes, grouping up the gene expressions and cell viability that concentrates on each mechanisms was our determined training goal, thus, Attention-based[2] model was required. Furthermore, Tabnet, the model we implemented, enables local interpretability that visualizes the importance of features and how they are combined, and global interpretability which quantifies the contribution of each feature to the trained model.

2.3.1. Tabnet. [3] Tabnet uses a sequential attention mechanism to choose a subset of meaningful features to process at each decision step. Selection based on samples enables efficient learning as the model capacity is fully used for the most significant features, and it also produces more interpretable decisions by visually selecting masks. In our condition, which gene or cell activation.

2.3.2. Ghost batch normalization. [4] GBN allows us to train large batches of data and promotes better simultaneously. Simply, we split the input batch into equal-sized sub-batches and apply the same Batch Normalization layer on them. All the batch normalization layers used in the model except the first batch normalization layer applied to the input features are GBN layers.

1. M. Mattiazzi, T. Curk, I. Krizaj, B. Zupan and U. Petrovic, *Inference of Molecular Mechanism of Action from Genetic Interaction and Gene Expression Data*
2. André F. T. Martins, Ramón Fernandez Astudillo, *From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification*
3. Sercan O. Arık, Tomas Pfister, *TabNet: Attentive Interpretable Tabular Learning*
4. *https://towardsdatascience.com/implementing-tabnet-in-pytorch-fc977c383279*

2.3.3. Gated linear unit. GLU decides information that is allowed to flow through the network: we doubled the dimension of the input features to the GLU using a fully connected layer, then normalize the resultant matrix using a GBN Layer, applying a sigmoid to the second half of the resultant features and multiply the results to the first half. The result is multiplied with a scaling factor and added to the input. This summed result is the input for the next GLU Block in the sequence. The Leaky ReLU activation is applied on the dimensioned vector.

2.3.4. Binary cross entropy with logits. The evaluation method, binary cross entropy with logits, was determined by Kaggle and is showed below in the Results.

2.3.5. Pre-train. A non-scored training set was provided, which meant composed of outputs that were not going to appear on the testing data. They were our pre-trained training sets.

2.3.6. Blend. Finally, we shuffled the training set in random combination to create different model, and concluded the predictions into our latest submission.

3. Results

3.1. evaluation formula

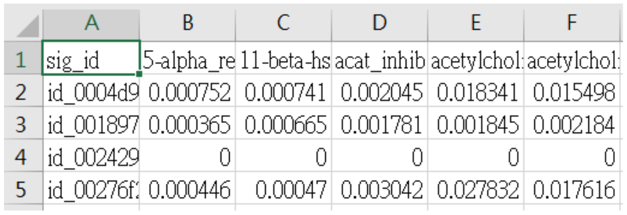
$$\text{score} = -\frac{1}{M}\sum_{m=1}^{M}\frac{1}{N}\sum_{i=1}^{N}\left[y_{i,m}\log(\hat{y}_{i,m}) + (1 - y_{i,m})\log(1 - \hat{y}_{i,m})\right]$$

Figure 1. score

Self-test average log loss of each MoA label possibility prediction score approximately 0.017 (about 500th at public leader board)

3.2. submission sample

We evaluated training data to get models and generated this possibility prediction form of MoA labels.



	A	B	C	D	E	F
1	sig_id	5-alpha_re	11-beta-hs	acat_inhib	acetylchol:	acetylchol:
2	id_0004d9	0.000752	0.000741	0.002045	0.018341	0.015498
3	id_001897	0.000365	0.000665	0.001781	0.001845	0.002184
4	id_002429	0	0	0	0	0
5	id_00276f:	0.000446	0.00047	0.003042	0.027832	0.017616

Figure 2. submission sample

4. Discussion and Conclusion

Our results show that augmentation of the dataset is capable of achieving remarkable results on a highly challenging dataset using interpretable learning methods. It is notable that our network's performance degrades if augmentation is removed. So the augmenting methods really are important for achieving our results.

Due to limit of time, we did not use much combination of models even though we expect that it will help, especially if we obtain enough computational power to significantly increase the size of the network without obtaining a corresponding increase in the amount of labeled data.

Thus far, our results have improved as we have made our dataset stronger and used a more powerful model to train it, but we stillhave a long way to go if we want to get higher rank in this Kaggle competition.

5. Author Contribution Statements

C.H.C (0.25): data process design, model searching, final presentation and report(methods)
S.T.C (0.25): model structure, result evaluation, final presentation and report(results)
A.Y.C (0.25): MoA algorithm, model structure, final presentation and report(model)
H.C.C (0.25): Perturbation algorithm, augmentation design, final presentation and report(overfitting)

References

[1] M. Mattiazzi, T. Curk, I. Krizaj, B. Zupan and U. Petrovic, *Inference of Molecular Mechanism of Action from Genetic Interaction and Gene Expression Data*

[2] André F. T. Martins, Ramón Fernandez Astudillo, *From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification*

[3] Sercan O. Arık, Tomas Pfister, *TabNet: Attentive Interpretable Tabular Learning*