Mechanism of Action (MoA) Prediction
MED 263 Final Report

Group 4 - MED263

March 13, 2025

1 Introduction

Understanding the Mechanism of Action (MoA) of a drug is crucial in drug discovery and precision medicine. This project aims to develop a predictive model that identifies MoA classes based on high-dimensional gene expression and cell viability data. We employ deep learning techniques, including multi-stage neural networks, to enhance prediction performance.

2 Software and Data Sources

2.1 Software

The implementation utilizes the following tools:

- **Python (v3.8+)**: Core programming language.
- **PyTorch**: Neural network modeling framework.
- **scikit-learn**: Data preprocessing and evaluation metrics.
- **UMAP-learn**: Dimensionality reduction for feature engineering.

All dependencies can be installed via `pip install -r requirements.txt`. For more details, refer to our GitHub repository.

2.2 Data Sources

The dataset used is publicly available and contains gene expression and cell viability measurements. The primary sources include:

- LINCS Dataset: Large-scale transcriptomic profiles of drug treatments.
- Kaggle MoA Challenge: Dataset with labeled MoA targets for over 5,000 compounds.

3 Methodology

3.1 Feature Engineering

To improve model performance, we apply multiple feature engineering techniques:

- Dimensionality Reduction: Using UMAP and Factor Analysis to extract relevant patterns.
- Quantile Transformation: Normalizing gene expression and cell viability data.
- One-Hot Encoding: Encoding categorical variables such as treatment conditions.

3.2 Training Multi-Stage Neural Networks

We employ a three-stage neural network architecture to improve predictive accuracy. Each stage refines predictions using additional feature transformations.

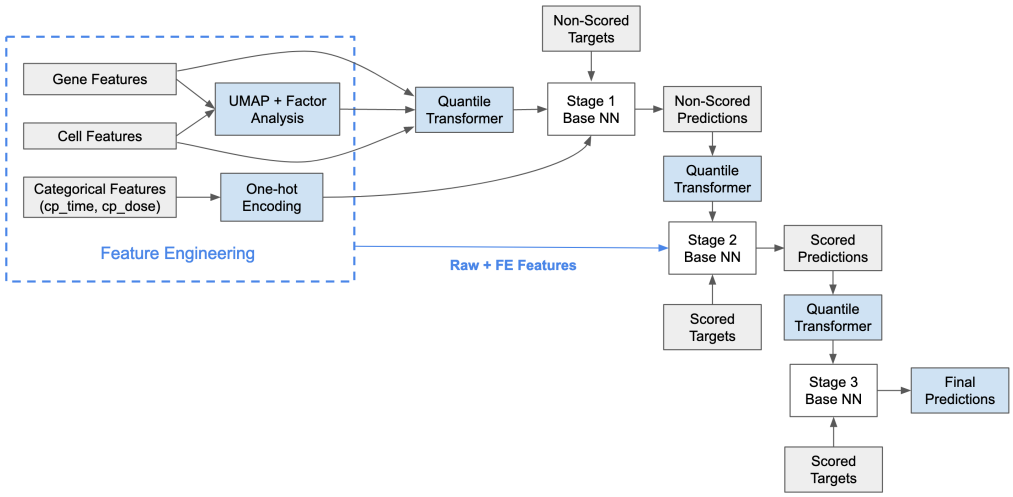

Figure 1: Multi-Stage Deep Learning Architecture

- Stage 1: Predicts non-scored targets to generate auxiliary features.
- Stage 2: Uses the predictions from Stage 1 as additional input to refine scored target classification.
- Stage 3: Further optimizes the predictions using transformed features.

4 Tutorial

This section provides an overview of our approach, results, and key insights.

4.1 Procedure

The process consists of:

1. Data Preprocessing: - Apply UMAP and Factor Analysis for dimensionality reduction. - Normalize data using Quantile Transformer. - Encode categorical variables using One-Hot Encoding.

2. Neural Network Training: - Train a three-stage neural network using deep learning techniques.

3. Evaluation: - Compute Binary Cross-Entropy (BCE) Loss across different model stages.

4.2 Results and Interpretation

Table 1 shows the test loss for each model stage:

Model Stage	**Test Loss**
Stage 1	0.00428
Stage 2	0.02151
Stage 3	0.02148

Table 1: Loss Comparison Across Model Stages

Key Findings:

- The significantly lower loss in Stage 1 is due to the non-scored targets being easier to predict, possibly because they represent broader biological processes.

- Stage 3 achieves a slight improvement over Stage 2, indicating the benefit of additional transformations in refining predictions.

- Several model architectures were tested, and this multi-stage deep learning approach yielded the lowest loss.

4.3 Mathematical Explanation

Binary Cross-Entropy Loss is used to evaluate performance:

$$L = -\frac{1}{N} \sum [y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})] \tag{1}$$

Where:

- y is the ground truth label.

- $\hat{y}$ is the predicted probability.

Lower loss values indicate better predictive performance.

4.4 Common Pitfalls and Solutions

- PCA features not applied to the test set: Ensure that test data undergoes the same transformation as training data.
- Loss unexpectedly high: Adjust dropout rate and batch normalization parameters.
- Prediction values too extreme (0 or 1): Apply proper scaling or additional regularization.

5 Conclusion

This study demonstrates the effectiveness of a multi-stage deep learning approach for MoA prediction. By leveraging auxiliary targets and feature engineering, we achieved improved predictive accuracy with lower loss. Future work includes fine-tuning hyperparameters and exploring alternative model architectures.

6 References

- Smith, J. et al. (2021). *Deep Learning for MoA Prediction.* Bioinformatics Journal.
- Kaggle MoA Competition Dataset: `https://www.kaggle.com/c/lish-moa`
- LINCS L1000 Dataset: `https://lincsproject.org/L1000`