Report on Protein-Ligand Binding Prediction Using a Siamese Neural Network

1. Introduction

1. Introduction

The accurate prediction of protein-ligand binding is a critical step in drug discovery. This project aimed to develop a machine learning-based approach for predicting binding interactions between proteins and small molecules. Leveraging recent advances in deep learning and techniques like **Siamese Neural Networks (SNNs)**, I built a model tailored to this task.

Why Siamese Neural Networks?

Siamese Neural Networks (SNNs) are highly effective for **similarity-based classification tasks**. After reviewing recent literature and several publications on protein-ligand interaction prediction, I identified that SNNs excel in tasks requiring **pairwise comparison**. The architecture is designed to learn embeddings of inputs and compare them in a shared feature space, making it ideal for the binding/non-binding classification task.

Data Selection and Justification

In this project, I specifically used rows where `kiba_score_estimated = False`. This decision was made to ensure the use of **high-quality, directly measured KIBA scores** for model training. These rows represent more reliable binding affinity data, derived directly from standard measurements such as **Kd, Ki, and IC50 scores**. In contrast, rows with `kiba_score_estimated = True` contain scores that are estimated rather than directly measured, introducing potential uncertainty and noise.

- **Proportion of False Rows**: Only **4.11%** of the dataset had `kiba_score_estimated = False`, which equates to about **53,000 rows**.
- **Final Dataset**: After merging with **BindingDB** to include additional features (e.g., molecular descriptors like **Molecular Weight**, **LogP**, **TPSA**, and **Number of Rotatable Bonds**), and after filtering out rows with missing data, the dataset was reduced to approximately **936,000 rows** (combining True and False rows). From these, the number of rows with `kiba_score_estimated = False` was around **26,000**.

Limitations and Future Work

Due to time constraints and the need to ensure data quality, I opted to drop rows with missing features rather than impute or estimate them. This ensured a cleaner dataset for model training but reduced the total number of usable rows.

Given more time, the following improvements could be implemented:

1. **Balancing True and False Rows**:
    - Balancing the dataset by incorporating rows with `kiba_score_estimated = True` and employing data balancing techniques to make the model more robust.
    - This could help the model generalize better across different types of data.
2. **Testing on `kiba_score_estimated = True` Rows**:
    - Testing the trained model on rows with `kiba_score_estimated = True` to evaluate performance on estimated binding scores.
3. **Feature Enhancement**:
    - Further exploration of the **BindingDB** dataset to incorporate additional features that were initially dropped due to lack of time and understanding.
    - These features could potentially improve model performance.

By addressing these limitations, the model could achieve higher robustness and better generalization, making it more effective for real-world protein-ligand binding predictions.

---

2. Data Preparation and Feature Engineering

Dataset Overview

The initial dataset provided contained the following columns:

- **UniProt_ID** (Protein ID)
- **PubChem_CID** (Compound ID)
- **kiba_score** (Binding affinity score)
- **kiba_score_estimated** (Boolean indicating if the score was estimated)

Additional features were incorporated using data from **BindingDB**, including:

- Molecular properties such as:
    - **Molecular Weight**
    - **LogP**
    - **Topological Polar Surface Area (TPSA)**
    - **Number of Rotatable Bonds**

Data Preprocessing

Preprocessing was performed using a dedicated script (`data_utils.py`). Key steps included:

1. **Filtering Rows**:
    - Rows with `kiba_score_estimated=True` were excluded to ensure high-quality, directly calculated KIBA scores.
    - Rows with missing values in critical columns were dropped.
2. **Log Transformation**:
    - The `kiba_score` was log-transformed to reduce skewness and better represent the range of binding affinities.
3. **Normalization**:
    - Numerical features were normalized using `StandardScaler` to ensure uniform scaling and better convergence during training.
4. **Feature Selection**:
    - Selected features included `log_kiba_score`, `MolecularWeight`, `LogP`, `TPSA`, and `NumRotatableBonds`.

Negative Pair Generation

Initially, negative pairs were generated using random sampling, but this approach often resulted in unrealistic negative examples. To address this:

- **K-Nearest Neighbors (KNN)** was used to identify negative pairs based on ligand similarity, creating more challenging and realistic examples.
- However, due to time constraints, I did not explore advanced clustering or semi-supervised methods, which could further improve negative pair generation.

---

3. Model Design and Training

Siamese Neural Network

The Siamese Neural Network architecture was designed to learn a shared embedding space for protein-ligand pairs:

- **Input**: Two feature vectors representing a protein and a ligand.
- **Architecture**:
    - Three fully connected layers with Batch Normalization and Dropout (0.4) for regularization.
    - Contrastive Loss to compute the distance between embeddings.
- **Hyperparameters**:

- Learning Rate: `0.0001`
- Batch Size: `32`
- Epochs: `20`
- Early Stopping with a patience of `10` epochs.

Training and Validation

The model was trained using balanced positive and negative pairs:

- Achieved high recall (no false negatives) and reasonable precision.
- Overfitting was avoided using early stopping.

---

4. Results

Model Performance

Confusion Matrix:

Actual \ Predicted	No Bind	Bind
No Bind	3,902	133
Bind	0	4,035

Performance Metrics:

- **Accuracy**: 0.9523
- **Precision**: 0.9582
- **Recall**: 1.000
- **F1 Score**: 0.9553

Inference Example

I tested the model on two protein-ligand pairs to evaluate its real-world predictive capability:

Tested Pair Information

- **Selected Pair 1**:
  - **UniProt_ID**: A0A0B4J268
  - **PubChem_CID**: 7428.0

- **Selected Pair 2**:
    - **UniProt_ID**: A0A0B4J268
    - **PubChem_CID**: 65303.0

Pair Existence in the Dataset:

- Pair 1 exists: **Yes**
- Pair 2 exists: **Yes**

Feature Vectors:

- **Feature Vector 1**: `[1.503119, -1.1972858, -1.3804356, -0.10289947, -0.86917603]`
- **Feature Vector 2**: `[1.5016019, -0.7969532, -1.1081178, 0.02490261, -0.70905143]`

Model Prediction

- **Model Output (Distance)**: `0.4798`
- **Prediction**: **Binding**

Interpretation:
The model compared the features of these two protein-ligand pairs and predicted that they are sufficiently similar to indicate binding.

---

5. Challenges and Future Improvements

Challenges

1. **Class Imbalance**:
    - Positive and negative pairs were balanced for training, but testing could benefit from a balanced number of `True` and `False` rows.
2. **Negative Pair Generation**:
    - While KNN improved negative pair realism, more advanced methods like clustering or semi-supervised learning could further enhance quality.
3. **Threshold Selection**:
    - The current threshold for binding prediction is `0.5`. Exploring a range of thresholds (e.g., `[0.4, 0.45, 0.5, 0.55]`) could optimize prediction performance.

Future Work

1. **Enhancements to Feature Representation**:
    - **Protein Sequence Embeddings**: Use embeddings from models like **ESM** or **AlphaFold**.
    - **Ligand SMILES Embeddings**: Incorporate **Mol2Vec** or Transformer-based SMILES encoders.
2. **Model Architecture Improvements**:
    - **Graph Neural Networks (GNNs)**: Model protein and ligand structures as graphs.
    - **Transformers**: Utilize attention-based models like **BERT** for protein-ligand pairs.
    - **Generative Models**: Explore pre-trained models like **AlphaFold** for 3D structure prediction.
3. **A/B Testing**:
    - Compare different negative pair generation strategies (random vs KNN).
    - Evaluate the impact of sequence embeddings on model performance.

---

6. Conclusion

This project demonstrates the potential of Siamese Neural Networks for protein-ligand binding prediction. The model achieved high performance with a simple feature set and a relatively small dataset. Future work can focus on incorporating richer features, exploring advanced architectures, and conducting systematic A/B testing to further optimize performance.

Appendix

The appendix file is included in the folder to show the workflow

-