Data Mining Exam

This repository contains the implementation and evaluation of KANG for various graph learning tasks

Core Implementation Files

Graph Regression Framework

- **`graph_regression.py`** : Main script for performing graph-level regression tasks using the KANG model. Supports QM8 and QM9 molecular datasets with configurable target properties. Implements training, validation, and testing loops with early stopping and model checkpointing.

Molecular Graph Conversion

- **`smiles_to_graph.py`** : Utility module that converts SMILES (Simplified Molecular Input Line Entry System) strings into PyTorch Geometric graph representations. Implements atom and bond feature extraction, creating node features (atomic number, degree, formal charge, aromaticity, hydrogen count) and edge features (bond type, conjugation, ring membership, stereochemistry).

Dataset Implementations

The following four files implement PyTorch Geometric dataset classes that convert raw molecular data into graph representations suitable for training graph neural networks:

- **`qm8_dataset.py`**
- **`qm9_dataset.py`**
- **`toxcast_dataset.py`**
- **`hiv_dataset.py`**

These implementations handle the conversion of SMILES strings to molecular graphs, feature extraction, data validation, and provide standardized interfaces for training and evaluation.

Data Resources

`data/` Folder

Contains the raw CSV files for all molecular datasets:

- **`qm8.csv`** : Quantum mechanical properties for ~22k small organic molecules
- **`qm9.csv`** : Quantum mechanical properties for ~134k stable organic molecules
- **`HIV.csv`** : HIV inhibition data for molecular compounds
- **`toxcast_data.csv`** : Multi-endpoint toxicological screening data

State-of-the-Art References

`SoTA/` Folder

Contains reference papers documenting state-of-the-art performance benchmarks:

- **`CIN++_HIV.pdf`** : Current best results for HIV inhibition prediction
- **`D-MPNN_QM8-9.pdf`** : State-of-the-art performance on QM8 and QM9 regression tasks
- **`Dumpling-GNN_ToxCast.pdf`** : Leading approach for ToxCast multi-task toxicity prediction

Model Architecture

`src/KANG_regression.py`

Specialized adaptation of the KANG model for graph-level regression tasks. This implementation modifies the original KANG model to:

- Support continuous target prediction instead of classification
- Integrate regression-specific loss functions and evaluation metrics
- Maintain the core KAN (Kolmogorov-Arnold Network) components

Hyperparameter Optimization

`optuna_search_main.py`

Automated hyperparameter tuning framework using Optuna for both classification and regression tasks. Optimizes key parameters including:

- Learning rate and weight decay
- Hidden dimensions and number of layers
- Dropout rates and batch sizes
- KAN-specific parameters

- Saves best configurations for reproducible experiments

Documentation

`DM_Final.pdf`

Comprehensive report documenting:

- Theoretical background and motivation for KANGnn
- Experimental setup and evaluation methodology
- Results comparison with state-of-the-art methods
- Conclusions and future work directions

Adaptation Notice

The original `graph_classification.py` file has been adapted to work seamlessly with our newly implemented datasets (ToxCast, HIV).

Authors

Sahar Khanlari - 2107563
Marco Natale - 1929854