Prediction of Drug-Drug interaction using GCN

SRUTHIMOL RAJESH

February 17, 2023

Problem statement

To find a Transductive method to predict links in a drug-drug interaction knowledge graph using Graph Convolutional Network

Introduction
Drug interactions may make drug less effective, cause unexpected side effects, or increase the action of a particular drug. Some drug interactions can even be harmful. We can reduce the risk of potentially harmful drug interactions.Drug-drug interactions occur when two or more drugs react with each other. This drug-drug interaction may cause you to experience an unexpected side effect. For example, mixing a drug you take to help you sleep (a sedative) and a drug you take for allergies (an antihistamine) can slow your reactions and make driving a car or operating machinery dangerous. The framework uses graph embedding to overcome data incompleteness and sparsity issues to make multiple DDI label predictions. First, a large-scale drug knowledge graph is generated from different sources. The knowledge graph is then embedded with comprehensive biomedical text into a common low-dimensional space. Finally, the learned embedding are used to efficiently compute rich DDI information through a link prediction process.
1. What is Knowledge Graphs
A knowledge graph is a data structure that represents a collection of interconnected information about various concepts and entities. It is a graphical representation of relationships between different pieces of information, such as people, places, events, and objects. Knowledge graphs are used to model complex real-world data and to enable computers to understand and process the meaning of the data. They are often used in natural language processing and artificial intelligence applications to enable machines to understand and interpret human language. Knowledge graphs are used for a variety of purposes, including:

- Information retrieval: Knowledge graphs can be used to help people find information more easily by providing a structured way to represent and organize data.
- Natural language processing: Knowledge graphs can be used to help computers understand and interpret human language.

- Artificial intelligence: Knowledge graphs can be used to support the development of artificial intelligence systems by providing a way to represent and organize knowledge in a machine-readable format.
- Data integration: Knowledge graphs can be used to integrate data from multiple sources and provide a single, unified view of the data.
- Data visualization: Knowledge graphs can be used to visualize complex data sets and relationships in a way that is easy for people to understand.
- Overall, the main purpose of a knowledge graph is to provide a structured and organized way to represent and store knowledge, which can then be used for a variety of purposes such as information retrieval, natural language processing, and artificial intelligence.
- Search engines: Knowledge graphs are often used by search engines to provide more relevant and accurate search results. For example, when you search for a person's name, a knowledge graph might be used to display information about that person, such as their occupation, location, and any relevant links or images.

2.Dataset

To investigate drug-drug interactions, we will apply graph ML techniques to the ogbl-ddi dataset, a homogeneous, unweighted, and undirected graph representing a drug-drug interaction network. This graph contains 4,267 nodes and 1,334,889 edges. Nodes represent FDA-approved or experimental drugs, and edges represent potential interactions between drugs.

3.Methods

- To train our prediction model, we first embed the nodes in the graph using various embedding approaches.
- We want to use graph structure information to predict whether two nodes (drugs) share an edge (interaction),and generate meaningful representations.
- Feed them into a neural network to predict the existence of an edge between any 2 nodes.
- Implement by using GraphSAGE model
- GraphSAGE message, update, and aggregate functions are as follows:

$$m_u^{(l)} = h_u^{(l-1)}$$

- Aggregate
- For aggregate, GraphSAGE uses an element-wise max-pooling function, where MLP can be an arbitrarily deep multi-layer perceptron (a simple single-layer architecture also suffices).

$$h_{N(v)}^{(l)} = \mathbf{Max}(\{MLP(m_u^{(l)}), \forall u \in N(v)\})$$

element-wise max-pooling function; multi-layer perceptron

- Update

$$h_v^{(l)} = \sigma(W^{(l)} \cdot CONCAT(h_v^{(l-1)}, h_{N(v)}^{(l)}))$$

4. Architecture