An algorithmic reasoning approach to GNNs

A project for the *Deep Learning* course

Angela Carraro, Matteo Scorcia

DSSC + IN20 - UniTS

Aim of the project

Graph Neural Networks can have a lot of meanings, there isn't just one architecture that can be recognized as "GNN". We will try to understand the general, abstract structure of a GNN that is presented in the book [3] (which also includes [2]) and to shed light about the relational inductive bias and combinatorial generalization of a GNN.

Our motivation is to better understand the extent to which graph neural networks are capable of **precise and logical reasoning**.

Graph Theory

Graphs are a widespread data structure and a universal language for describing and modelling complex systems. In the most general view, a graph is simply a collection of objects (i.e., nodes), along with a set of interactions (i.e., edges) between pairs of these objects.

Graphs are an important building block since they can naturally encode an entity-relationship structure, as well as an invariance to permutations (of both nodes and edges) and awareness of input sparsity.

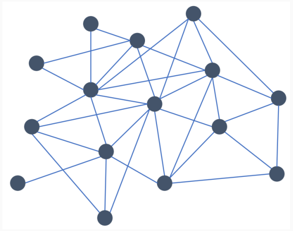

Figure 1: A graph.

Basics

Definition

A graph is a tuple $G = (V, E)$ where V is the set of nodes and E is the set of edges between these nodes. We denote an edge going from node $u \in V$ to node $v \in V$ as $(u, v) \in E$, so $E \subseteq V \times V$. The graph is **undirected** if $(u, v) \in E \iff (v, u) \in E$, otherwise it is **directed**.

A convenient way to represent graphs is through an adjacency matrix $A \in \mathbb{R}^{|V| \times |V|}$, with $A[u, v] = 1$ if $(u, v) \in E$ and $A[u, v] = 0$ otherwise. If the graph is undirected the matrix in *symmetric*. If the graph has weighted edges we have that $A[u, v] \in \mathbb{R}$.

We also have attribute or feature information associated with a graph. Most often these are *node-level attributes* that we represent using a real-valued matrix $\mathbf{X} \in \mathbb{R}^{d \times |V|}$, where we assume that the ordering of the nodes is consistent with the ordering in the adjacency matrix. In some cases we even associate real-valued *features with entire graphs*.

GNN real-world applications

GNNs are used for one of three tasks:

- *node classification*: predict the label of a given node
  $\longrightarrow$ E.g., predicting whether a user is a bot in a social network
- *edge prediction*: infer the edges between nodes in a graph
  $\longrightarrow$ E.g., content recommendation in online platforms, predicting drug side-effects, or inferring new facts in a relational databases
- *graph classification*: make independent predictions specific to each graph
  $\longrightarrow$ E.g., property prediction based on molecular graph structures

Thank you for your attention!

☺

References

Miltos Allamanis. *An Introduction to Graph Neural Networks: Models and Applications*. Youtube. 2020. URL: *https://www.youtube.com/watch?v=zCEYiCxrL_0*.

Peter W. Battaglia et al. "Relational inductive biases, deep learning, and graph networks". In: *CoRR* abs/1806.01261 (2018). arXiv: *1806.01261*. URL: *http://arxiv.org/abs/1806.01261*.

William L. Hamilton. *Graph Representation Learning*. Vol. 14. 3. Morgan and Claypool, 2020, pp. 1–159. URL: *https://www.cs.mcgill.ca/~wlh/grl_book/*.

William L. Hamilton. *Graph Representation Learning*. Youtube. 2021. URL: *https://www.youtube.com/watch?v=fbRDfhNrCwo*.

- William L. Hamilton, Rex Ying, and Jure Leskovec. “Representation Learning on Graphs: Methods and Applications”. In: *CoRR* abs/1709.05584 (2017). arXiv: *1709.05584*. URL: *http://arxiv.org/abs/1709.05584*.
- Petar Veličković. *Graph Representation Learning for Algorithmic Reasoning*. Youtube. 2020. URL: *https://www.youtube.com/watch?v=IPQ6CPoluok*.
- Petar Veličković. *Theoretical Foundations of Graph Neural Networks*. Youtube. 2021. URL: *https://www.youtube.com/watch?v=uF53xsT7mjc*.
- Yujun Yan et al. “Neural Execution Engines: Learning to Execute Subroutines”. In: *CoRR* abs/2006.08084 (2020). arXiv: *2006.08084*. URL: *https://arxiv.org/abs/2006.08084*.