COMP 642 — Machine Learning Feb 24, 2022

Lecture 13: Learning with Data Types 4: Graphs and relations

Lecturer: Anshumali Shrivastava Scribe By: Chieh Ju Chueh, Cynthia Zhu, Guan-Yu Lin, Yuying Liu

Disclaimer: *These lecture notes are intended to develop the thought process and intuition in machine learning. The materials are not thoroughly reviewed and can contain errors.*

1 Graph Representation

Mathematically, a graph G is defined as a tuple of a set of vertices V and a set of edges $E.G = (V, E)$. Each edge is a pair of two vertices. A graph is shown below in Figure 1.

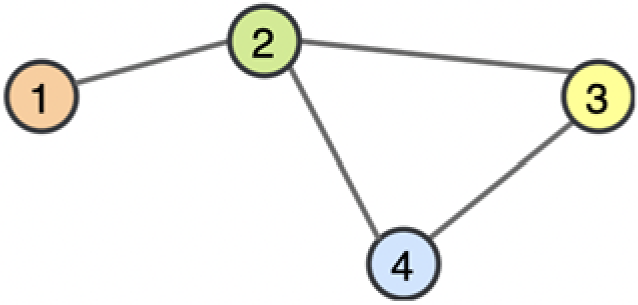
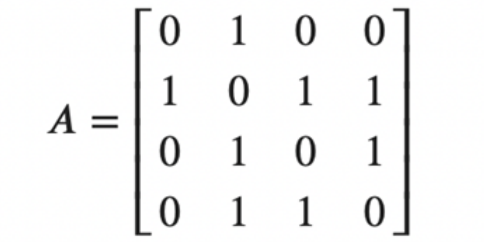


Figure 1: Simple graph

The vertices of this graph are $V = 1, 2, 3, 4$ and edges $E = (1, 2), (2, 3), (2, 4), (3, 4)$. We do not add pairs like $(4, 2)$ since we assume this graph is undirected. The graph can also be a directed graph if we draw directed edges.

We can use an adjacency matrix as Figure 2 below to represent the graph. The adjacency matrix is a square matrix whose elements indicate whether pairs of vertices are connected. A_{ij} is 1 if there is a connection from vertex i to j



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Figure 2: Graph representation using matrix

2 Origin of graph neural networks

2.1 Deficiencies of CNN

The core features of CNN are local connection, shared weights, and multi-layer. These are also very trivial in graph problems because graph structure is the most typical local connection structure. Secondly, shared weights can reduce the computational effort. Lastly, the multi-layer structure is the key to dealing with hierarchical patterns. Traditional deep learning methods have been applied to extract features from Euclidean space data with great success. However, many practical application scenarios are generated from non-Euclidean space, and the performance of traditional deep learning methods in dealing with non-Euclidean space data is still hardly satisfactory. CNN can only process Euclidean data, such as two-dimensional pictures and one-dimensional text data, which are the only special cases of the graph structure. In conclusion, it is difficult to use CNN for general graph structure.

2.2 Defects of Graph Embedding

Graph embeddings can be broadly classified into three categories: matrix decomposition, random wandering, and deep learning methods. However, these methods have two serious drawbacks. Firstly, the weights are not shared in the node encoding, which leads to a linear increase in the number of weights with more nodes. The second is that it leads to the lack of generalization capability of direct embedding methods, which means that they cannot handle dynamic graphs and generalize to new graphs.

3 Basic of Graph Neural Networks

3.1 Intuition of Graph Convolutions

Graph Convolutional Network is similar to convolutions in images in the sense that the filter parameters are typically shared over all locations in the graph. In Graph Convolutional Network, vertices will exchange information with neighbors. Each vertex in Graph Convolutional Network will first create a feature vector that represents the information it wants to send to all its neighbors. Then, the information will be sent and each vertex will receive information from all its neighbors. Each vertex can further sum or take the mean of all the information it receives.

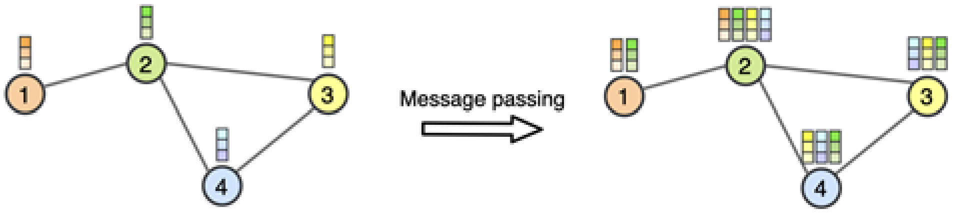


Figure 3: Information exchanging example of graph neural network

Figure 4 below is a basic graph neural network.

The first node's output values are the average of itself and the second node. In GNN, we also want to allow nodes to exchange information beyond their neighbors. This can be achieved by applying multiple GNN layers, which gives us the final layout of a GNN.

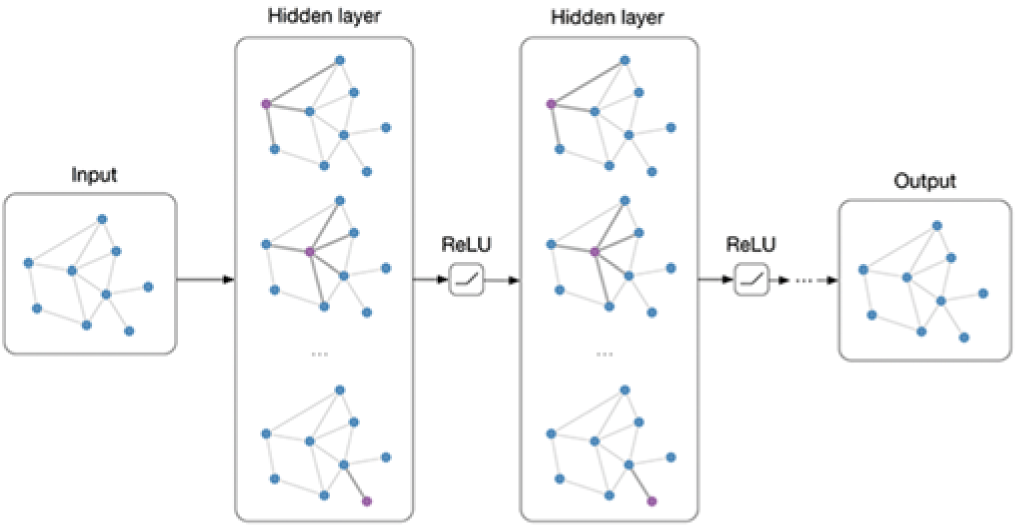


Figure 4: Visualization of simple graph neural network

3.2 Definition of GNN

A graph is a kind of data structure, often used graph structure containing nodes and edges, GNN is a branch of deep learning on graph structure. A graph neural network (GNN) is a class of neural networks for processing data best represented by graph data structures. They were popularized by their use in supervised learning on the properties of various molecules. Since their inception, variants of the message passing neural network (MPNN) framework have been proposed. These models optimize GNNs for use on larger graphs and apply them to domains such as social networks, citation networks, and online communities.

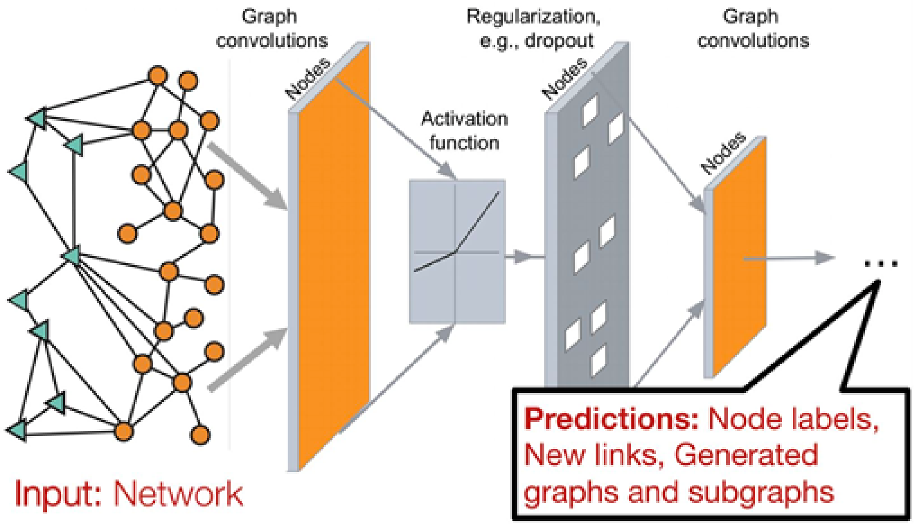

Graph convolutions
Regularization, e.g., dropout
Graph convolutions
Nodes
Activation function
Nodes
Nodes
...
Predictions: Node labels, New links, Generated graphs and subgraphs
Input: Network

Figure 5: Application model of graph neural network

3.3 Advantage of GNN

3.3.1 Nodes

Since there isn't a natural order of nodes in the graph, if we use CNN and RNN to completely present a graph, we should traverse all possible orders, which is time-consuming. GNN can solve this problem by propagating on each node respectively, ignoring the input order of nodes

3.3.2 Edges (the edges in the graph represent the dependencies between nodes)

Instead of expressing this dependency relationship explicitly, traditional neural networks express the relationship between nodes indirectly through different node features, and this dependency information is only used as features of the nodes. CNN can be propagated through graphical structures instead of using them as part of node features, and update the hidden states of nodes by a weighted summation of neighboring nodes.

3.3.3 Reasoning

Reasoning is a very important research topic in advanced artificial intelligence, and the reasoning process in the human brain is almost always based on graphs extracted from everyday experience. Standard neural networks have shown the ability to generate synthetic images and documents by learning data distributions, while they still cannot learn reasoning graphs from large experimental data. However, GNNs explore the generation of graphs from unstructured data such as scene images and story documents, which can be a powerful neural model for further advanced AI.

3.4 Limitations of GNN Model

a. Updating the hidden states of nodes iteratively for fixed points is inefficient.

b. The original GNN uses the same parameters in the iteration, while other more well-known models use different parameters in different network layers for hierarchical feature extraction, which allows the model to learn deeper feature representations. Moreover, the update of the hidden layer of nodes is a sequential process that can be further optimized by using RNN kernels, such as GRU and LSTM.

c. There may be some informative information features on the edges that cannot be effectively taken into account in the GNN. For example, edges in a knowledge graph have relation types and message propagation through different edges should be different according to their types.

d. If we need to learn vector representations of nodes instead of graph representations, it is not suitable to use fixed points because the distribution of representations in the fixed points will be much more smooth in value and less informative for distinguishing each node.

4 What GNN can do

a. Node Classification: The goal here is to predict the labels of nodes by considering their labels.

b. Link Prediction: In this case, the goal is to predict the relationships between the various entities in the graph. This can be applied, for example, to predicting connections in social networks.

c. Graph Clustering: This involves dividing the nodes of the graph into clusters. Partitioning can be done based on edge weights or edge distances or by treating the graph as objects and grouping similar objects together.

d. Graph Classification: This involves classifying the graph into a class. This can be applied to social network analysis and document classification in natural language processing. Other applications in NLP include text classification, extraction of semantic relations between texts, and sequence tagging.
The most common task for graph classification is molecular property prediction, in which molecules are represented as graphs. Each atom is linked to a node.

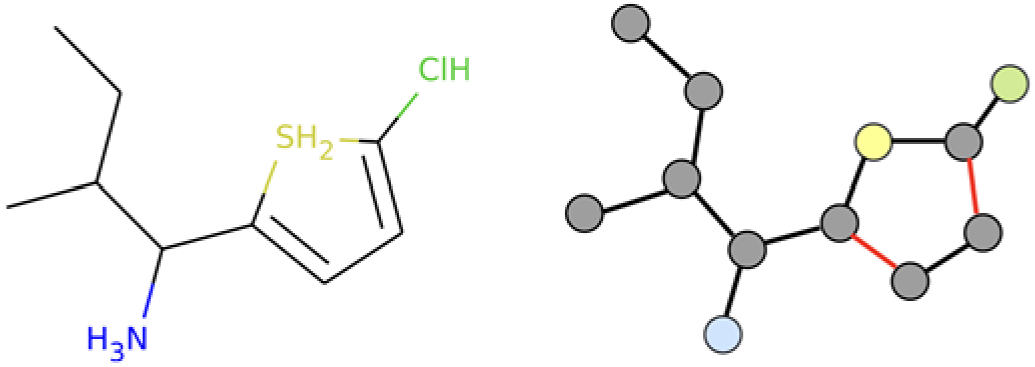


Figure 6: Molecular property prediction in GNN classification

On the left, we have a molecule with different atoms. On the right, we have the graph representation. Atoms(nodes) have their own features as well as edges.

e. Computer Vision: In the field of computer vision, GNNs can be used to generate target detection regions. They can also be used to generate scene maps for image classification. Scene generation models then identify objects in the image and the semantic relationships between them. Other applications in this field include interaction detection and region classification.

5 Future Directions of GNN

a. The success of deep learning lies in deep neural architectures. For example, in image classification, the model ResNet has 152 layers. However, in graph networks, empirical studies have shown that the model performance decreases dramatically as the number of network layers increases. This is due to the effect of graph convolution, which essentially drives the representations of neighboring nodes closer to each other, so that theoretically, through infinite convolution, the representations of all nodes will converge to a single point. This leads to the question: Is deepening the network still a good strategy for learning graph-structured data?

b. The receptive field of a node is a set of nodes, including the central node and its nearest neighbors. The number of nodes' nearest neighbors (nodes) follows a power-law distribution. Some nodes may have only one nearest neighbor, while others have thousands of nearest neighbors. Despite the sampling strategy, how to select a representative receptive field of nodes remains to be explored.

c. Most graph neural networks do not scale well to large graphs. The main reason is that when stacking multiple layers of a graph convolution, the final state of a node involves the hidden state of a large number of its nearest neighbors, leading to a very complicated backpropagation. Although some methods have been tried to improve model efficiency by fast sampling and subgraph training, they still do not scale to large graph depth architectures.

d. Most current graph neural networks deal with static homogeneous graphs. On the one hand, graph architectures are assumed to be fixed. It is assumed that the nodes and edges of the graph come from the same source. However, these two assumptions are unrealistic in many cases. In a social network, a new person may join at any time, and a pre-existing person may drop out of that social network. In a recommender system, products may have different types, and their output may be in different forms, perhaps text or images. Therefore, new methods should be developed to deal with dynamic and heterogeneous graph structures.

References

[1] Dickson, Ben. “What Are Graph Neural Networks (GNN)?” *VentureBeat*, 12 Oct. 2021, venturebeat.com/2021/10/13/what-are-graph-neural-networks-gnn.

[2] Phlippe, and Phlippe. “Uvadlc_notebooks/GNN_overview.Ipynb at Master · Phlippe/Uvadlc _notebooks.” *GitHub*, github.com/phlippe/uvadlc_notebooks/blob/master/docs /tutorial_notebooks/tutorial7/GNN_overview.ipynb. Accessed 2 Mar. 2022.

[3] White, Andrew D. *Deep Learning for Molecules and Materials*. 2021.

[4] Wu, Zonghan et al. “A Comprehensive Survey on Graph Neural Networks.” *IEEE Transactions on Neural Networks and Learning Systems* 32 (2019): 4-24.

[5] Xu, Keyulu, et al. ”How powerful are graph neural networks?.” *arXiv preprint arXiv:1810.00826* (2018).

[6] Zhou, Jie, et al. ”Graph neural networks: A review of methods and applications.” *AI Open* 1 (2020): 57-81.