Week 10 Reflections

Course	CS 598 - Deep Learning for Healthcare
Submitted By	Rohit Bhagwat (rohitb2@illinois.edu)

Questions

What are the main messages you learned from this chapter?

This week was focused on Graph Neural Networks (GNNs).

Compared with other representations such as vectors or sequences, graphs are expressive models that can capture more complex interactions between heterogeneous biomedical concepts.

Deep learning methods are difficult to directly apply to graphs due to several challenges:

- *Irregular structure*
- *Heterogeneity of graphs*
- *Large graphs*

Tasks on Graph

- **Node classification** is to classify each node in the graph into a set of predefined labels
- **Link prediction** is to estimate the missing edges on a graph
- **Community detection** is to partition the graph into tight subgraphs (i.e., communities)
- **Graph property prediction** is about predicting certain properties for the entire graph
- **Graph generation** is to produce new graphs based on existing graphs

Graph Neural Networks (GNNs)

The main idea of a graph neural network is to pass messages across edges and aggregate those messages on nodes to compute node embedding vectors. The general design can have:

- a neighborhood aggregation function
- a loss function on the embeddings
- training on a set of nodes based on local neighborhood
- generating embeddings for any node based on node features

Graph Convolutional Networks (GCNs)

It is an approximation of the original GNN. GCN produces node representations based on the input graph structure (e.g., adjacency matrix A) and input node features X. GCN models encode node information based on its neighbors on the graph. Intuitively GCN represents a node as a function of its neighbors like convolution operations in CNN. Iteratively GCN aggregates information from a node's local neighborhood to produce better features (or embedding) for that node.

Message Passing Neural Networks (MPNNs)

To extend GCN to a more general framework, a message passing neural network (MPNN) has been introduced, where node and edge features are supported, and many existing methods are covered under this framework. MPNN framework has two stages: the message passing stage and the readout stage. The general MPNN can support edge features but require storing edge-based activations and can be much more expensive than GCN. MPNN is only applicable to small graphs such as molecule graphs.

Graph Attention Networks (GATs)

The graph attention networks utilize attention mechanism on graphs, which allows different weights to different nodes in the neighborhood. Computationally, GAT is more efficient than MPNN with edge features due to the ease of parallel computation. Compared to GCN, GAT allows different importance to the neighboring nodes, making the model more flexible.

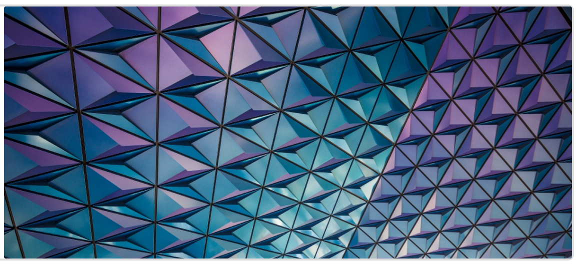
What related resources (book, paper, blog, link) do you recommend your classmates to checkout?

Some great references, that also proved useful -

Program a simple Graph Net in PyTorch

A quite new and fast-evolving field in machine learning is graph neural nets. As the name already suggests they are capable of learning relations between nodes in any kind of network. This can be useful in a lot of areas, e.g.

https://towardsdatascience.com/program-a-simple-graph-net-in-pytorch-e00b500a642d



https://towardsdatascience.com/beyond-graph-convolution-networks-8f22c403955a

Which part do you want to improve in this chapter?

In the lectures as well as in the chapter, we are introduced to the concepts very well. The author and the professor walked through the idea of Graph Neural Network models..

What is the main difference between Graph Convolutional Network and Graph Attention Network?

Graph Attention Networks employ attention mechanisms which assign larger weights to the more important nodes, walks, or models. This allows GAT model to be more adaptable.

GCNs

GCNs aggregate information from a node's local neighborhood to produce better features (or embedding) for that node

GCNs explicitly assign a non-parametric weight to the neighbor during the aggregation process

GATs

GATs seek an aggregation function to fuse the neighboring nodes' representations from multiple candidate models to learn a new representation

GATs implicitly capture the weight via an end-to- end neural network architecture, so that more important nodes receive larger weights