

Lecture Note: Graph Neural Networks

Bài Giảng: Mạng Nơon Đồ Thị

Nguyễn Quân Bá Hồng¹

Ngày 6 tháng 8 năm 2025

¹A scientist- & creative artist wannabe, a mathematics & computer science lecturer of Department of Artificial Intelligence & Data Science (AIDS), School of Technology (SOT), UMT Trường Đại học Quản lý & Công nghệ TP.HCM, Hồ Chí Minh City, Việt Nam.
E-mail: nguyenquanbahong@gmail.com & hong.nguyenquanba@umt.edu.vn. Website: <https://nqbh.github.io/>. GitHub: <https://github.com/NQBH>.

Mục lục

1	Introduction to Graph Neural Networks – Nhập Môn Mạng Nơ-ron Đồ Thị	3
1.1	Some Basic Concepts – Vài Khái Niệm Cơ Bản	3
1.1.1	Understanding Graph Structures – Hiểu Về Cấu Trúc Đồ Thị	3
1.2	Architecture of GNNs – Kiến Trúc của GNNs	5
1.3	Message Passing Layers – Các Lớp Truyền Tin Nhấn	5
1.3.1	Graph convolutional network (GCN) – Mạng lưới tích chập đồ thị	6
1.3.2	Graph attention network – Mạng lưới chú ý đồ thị	7
1.3.3	Gated graph sequence neural network – Mạng nơ-ron chuỗi đồ thị có cổng	7
1.4	Local Pooling Layers – Lớp Gộp Cục Bộ	7
1.5	Heterophilic Graph Learning – Học Đồ Thị Dị Hướng	7
1.6	Applications of GNNs – Ứng Dụng của GNNs	7
2	Miscellaneous	8
2.1	Contributors	8

Preface

Abstract

This text is a part of the series *Some Topics in Advanced STEM & Beyond*:

URL: https://nqbh.github.io/advanced_STEM/.

Latest version:

- *Lecture Note: Graph Neural Networks*

Bài Giảng: Mạng Nơ-ron Đồ Thị.

PDF: URL: https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/lecture/NQBH_combinatorics_graph_theory_lecture.pdf.

TEX: URL: https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/lecture/NQBH_combinatorics_graph_theory_lecture.tex.

- *Slide: Combinatorics & Graph Theory – Slide Bài Giảng: Tổ Hợp & Lý Thuyết Đồ Thị.*

PDF: URL: https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/slide/NQBH_combinatorics_graph_theory_slide.pdf.

TEX: URL: https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/slide/NQBH_combinatorics_graph_theory_slide.tex.

- *Survey: Combinatorics & Graph Theory – Khảo Sát: Tổ Hợp & Lý Thuyết Đồ Thị.*

PDF: URL: https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/NQBH_combinatorics.pdf.

TEX: URL: https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/NQBH_combinatorics.tex.

- Codes:

- C/C++: https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/C++.

- Pascal: https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/Pascal.

- Python: https://github.com/NQBH/advanced_STEM_beyond/blob/main/combinatorics/Python.

Tài liệu này là bài giảng tôi dạy cho sinh viên Khoa Công Nghệ (undergraduate Computer Science students) chuyên ngành Kỹ Thuật Phần Mềm (Software Engineering, abbr., SE) & Trí Tuệ Nhân Tạo-Khoa Học Dữ Liệu (Artificial Intelligence-Data Science, abbr., AIDS) nên sẽ tập trung vào phương diện lập trình cho các khái niệm Tổ hợp & Lý thuyết đồ thị được nghiên cứu. Bài giảng này gồm 2 phần chính:

- **Part I: Combinatorics – Tổ Hợp.**

- **Part II: Graph Theory – Lý Thuyết Đồ Thị.** Tập trung vào các thuật toán trên cây (algorithms on trees) & thuật toán trên đồ thị (algorithms on graphs)

Chương 1

Introduction to Graph Neural Networks – Nhập Môn Mạng Nơ-ron Đồ Thị

Contents

1.1	Some Basic Concepts – Vài Khái Niệm Cơ Bản	3
1.1.1	Understanding Graph Structures – Hiểu Về Cấu Trúc Đồ Thị	3
1.2	Architecture of GNNs – Kiến Trúc của GNNs	5
1.3	Message Passing Layers – Các Lớp Truyền Tin Nhắn	5
1.3.1	Graph convolutional network (GCN) – Mạng lưới tích chập đồ thị	6
1.3.2	Graph attention network – Mạng lưới chú ý đồ thị	7
1.3.3	Gated graph sequence neural network – Mạng nơ-ron chuỗi đồ thị có cổng	7
1.4	Local Pooling Layers – Lớp Gộp Cục Bộ	7
1.5	Heterophilic Graph Learning – Học Đồ Thị Dị Hướng	7
1.6	Applications of GNNs – Ứng Dụng của GNNs	7

1.1 Some Basic Concepts – Vài Khái Niệm Cơ Bản

Resources – Tài nguyên.

1. [Geeks4Geeks/what are graph neural networks?](#).
2. [Wikipediagraph neural network](#).

We explore what GNNs are, how they work, their types, & their wide range of applications.

Graph Neural Networks (GNNs) are a neural network specifically designed to work with data represented as graphs. Unlike traditional neural networks, which operate on grid-like data structures like images (2D grids) or text (sequential), GNNs can model complex, non-Euclidean relationships in data, e.g. social networks, molecular structures, & knowledge graphs.

– Mạng nơ-ron đồ thị (GNN) là một mạng nơ-ron được thiết kế đặc biệt để hoạt động với dữ liệu được biểu diễn dưới dạng đồ thị. Không giống như các mạng nơ-ron truyền thống, hoạt động trên các cấu trúc dữ liệu dạng lưới như hình ảnh (lưới 2D) hoặc văn bản (tuyến tính), GNN có thể mô hình hóa các mối quan hệ phức tạp, phi Euclid trong dữ liệu, ví dụ như mạng xã hội, cấu trúc phân tử, & đồ thị tri thức.

1.1.1 Understanding Graph Structures – Hiểu Về Cấu Trúc Đồ Thị

A graph is a data structure that represents relationships between pairs of objects. Each object is known as a *node* (or *vertex*), & each connection between nodes is an *edge*. Graphs can vary based on the type of edges:

1. **Directed graphs:** In these graphs, edges have a direction, indicated by an arrow, showing a 1-way relationship from 1 node to another.
2. **Undirected graphs:** In these graphs, edges are bidirectional, representing 2-way relationships without any direction.

– Đồ thị là một cấu trúc dữ liệu biểu diễn mối quan hệ giữa các cặp đối tượng. Mỗi đối tượng được gọi là một *node* (hoặc *vertex*), và mỗi kết nối giữa các nút là một *edge*. Đồ thị có thể thay đổi tùy theo loại cạnh:

1. **Đồ thị có hướng:** Trong các đồ thị này, các cạnh có hướng, được biểu thị bằng mũi tên, thể hiện mối quan hệ một chiều từ nút này đến nút khác.
2. **Đồ thị vô hướng:** Trong các đồ thị này, các cạnh là song hướng, biểu diễn mối quan hệ hai chiều không có hướng.

Graphs can be classified based on their structural properties: **Type of graph** | **Description** | **Example/Use case**

1. *Homogeneous graph* | contains only 1 type of node & edge, simplifying representation & analysis. | Social network with people as nodes & friendships as edges.
2. *Heterogeneous graph* | consists of multiple types of nodes & edges, suitable for complex relationships | Knowledge graph with nodes for entities (people, places) & varied edges.
3. *Static graph* | remains fixed once started, with no changes to nodes or edges | Scenarios where relationships are constant over time.
4. *Dynamic graph* | allows the addition or deletion of nodes & edges, ideal for situations with frequently changing relationships. | Real-time communication tracking or evolving social networks.

Graph neural networks (GNN) are specialized artificial neural networks (ANNs) that are designed for tasks whose inputs are graphs.

– Mạng nơ-ron đồ thị (GNN) là mạng nơ-ron nhân tạo chuyên biệt (ANN) được thiết kế cho các tác vụ có đầu vào là đồ thị.

1 prominent example is molecular drug design. Each input sample is a graph representation of a molecule, where atoms form the nodes & chemical bonds between atoms form the edges. In addition to the graph representation, the input also includes known chemical properties for each of the atoms. Dataset samples may thus differ in length, reflecting the varying numbers of atoms in molecules, & the varying number of bonds between them. The task is to predict the efficacy of a given molecule for a specific medical application, like eliminating *E. coli* bacteria.

– 1 ví dụ nổi bật là thiết kế thuốc phân tử. Mỗi mẫu đầu vào là một biểu diễn đồ thị của một phân tử, trong đó các nguyên tử tạo thành các nút & các liên kết hóa học giữa các nguyên tử tạo thành các cạnh. Ngoài biểu diễn đồ thị, dữ liệu đầu vào còn bao gồm các đặc tính hóa học đã biết của từng nguyên tử. Do đó, các mẫu tập dữ liệu có thể khác nhau về độ dài, phản ánh số lượng nguyên tử khác nhau trong phân tử, & số lượng liên kết khác nhau giữa chúng. Nhiệm vụ là dự đoán hiệu quả của một phân tử nhất định cho một ứng dụng y tế cụ thể, chẳng hạn như loại bỏ vi khuẩn *E. coli*.

The key design element of GNNs is the use of *pairwise message passing*, s.t. graph nodes iteratively update their representations by exchanging information with their neighbors. Several GNN architectures have been proposed, which implement different flavors of message passing, started by recursive or convolutional constructive approaches. As of 2022, it is an open question whether it is possible to define GNN architectures “going beyond” message passing, or instead every GNN can be built on message passing over suitably defined graphs.

– Yếu tố thiết kế chính của GNN là việc sử dụng *pairwise message passing*, tức là các nút đồ thị cập nhật biểu diễn của chúng một cách lặp đi lặp lại bằng cách trao đổi thông tin với các nút lân cận. Một số kiến trúc GNN đã được đề xuất, triển khai các phương pháp truyền thông điệp khác nhau, bắt đầu bằng các phương pháp xây dựng đệ quy hoặc tích chập. Tính đến năm 2022, vẫn còn là một câu hỏi bỏ ngỏ liệu có thể định nghĩa kiến trúc GNN “vượt ra ngoài” việc truyền thông điệp hay không, hay thay vào đó, mọi GNN đều có thể được xây dựng dựa trên việc truyền thông điệp trên các đồ thị được xác định phù hợp.

In the more general subject of “geometric DL”, certain existing neural network architectures can be interpreted as GNNs operating on suitably defined graphs. A **convolutional neural network** layer, in the context of computer vision, can be considered a GNN applied to graphs whose nodes are **pixels** & only adjacent pixels are connected by edges in the graph. A **transformer** layer, in **natural language processing**, can be considered a GNN applied to **complete graphs** whose nodes are words or tokens in a passing of natural language text.

– Trong lĩnh vực tổng quát hơn của “học máy hình học”, một số kiến trúc mạng nơ-ron hiện có có thể được hiểu là mạng nơ-ron tích chập (GNN) hoạt động trên các đồ thị được định nghĩa phù hợp. Trong bối cảnh thị giác máy tính, một lớp mạng nơ-ron tích chập có thể được coi là một GNN áp dụng cho các đồ thị có các nút là **pixels** & chỉ các điểm ảnh liền kề được kết nối bằng các cạnh trong đồ thị. Trong xử lý ngôn ngữ tự nhiên, một lớp biến đổi có thể được coi là một GNN áp dụng cho các đồ thị hoàn chỉnh có các nút là các từ hoặc token trong quá trình truyền văn bản ngôn ngữ tự nhiên.

Relevant application domains for GNNs include natural language processing, social networks, citation networks, molecular biology, chemistry, physics, & NP-hard combinatorial optimization problem.

– Các lĩnh vực ứng dụng có liên quan cho GNN bao gồm xử lý ngôn ngữ tự nhiên, mạng xã hội, mạng trích dẫn, sinh học phân tử, hóa học, vật lý và bài toán tối ưu hóa tổ hợp NP-hard.

Open source libraries implementing GNNs include PyTorch Geometric (PyTorch), TensorFlow GNN (TensorFlow), Deep Graph Library (framework agnostic), jraph (Google JAX), & **GraphNeuralNetworks.jl/GeometricFlux.jl** (Julia, Flux).

1.2 Architecture of GNNs – Kiến Trúc của GNNs

The architecture of a generic GNN implements the following fundamental layers:

1. *Permutation equivariant*: a permutation equivariant layer maps a representation of a graph into an updated representation of the same graph. In the literature, permutation equivariant layers are implemented via pairwise message passing between graph nodes. Intuitively, in a message passing layer, nodes *update* their representations by *aggregating* the *messages* received from their immediate neighbors. As such, each message passing layer increases the receptive field of the GNN by 1 hop.
2. *Local pooling*: a local pooling layer coarsens the graph via downsampling. Local pooling is used to increase the receptive field of a GNN, in a similar fashion to pooling layers in CNNs. Examples include k-nearest neighbors pooling, top-k pooling, & self-attention pooling.
3. *Global pooling*: a global pooling layer, also known as *readout* layer, provides fixed-size representation of the whole graph. The global pooling layer must be permutation invariant, s.t. permutations in the ordering of graph nodes & edges do not alter the final output. Examples include element-wise sum, mean or maximum.

– Kiến trúc của một GNN chung triển khai các lớp cơ bản sau:

1. *Permutation equivariant*: một lớp permutation equivariant ánh xạ một biểu diễn của một đồ thị thành một biểu diễn được cập nhật của cùng một đồ thị. Trong các tài liệu, các lớp permutation equivariant được triển khai thông qua việc truyền thông điệp từng cặp giữa các nút đồ thị. Theo trực giác, trong một lớp truyền thông điệp, các nút *cập nhật* biểu diễn của chúng bằng cách *tổng hợp* các *thông điệp* nhận được từ các nút lân cận trực tiếp của chúng. Như vậy, mỗi lớp truyền thông điệp sẽ tăng trường tiếp nhận của GNN thêm 1 bước nhảy.
2. *Local pooling*: một lớp local pooling làm đồ thị thô hơn thông qua việc giảm mẫu. Local pooling được sử dụng để tăng trường tiếp nhận của GNN, tương tự như các lớp pooling trong CNN. Các ví dụ bao gồm pooling k-nearest neighbors, pooling top-k, và pooling & self-attention.
3. *Tổng hợp toàn cục*: một lớp tổng hợp toàn cục, còn được gọi là lớp *readout*, cung cấp biểu diễn kích thước cố định cho toàn bộ đồ thị. Lớp tổng hợp toàn cục phải bất biến hoán vị, tức là các hoán vị trong thứ tự các nút & cạnh của đồ thị không làm thay đổi kết quả đầu ra cuối cùng. Ví dụ bao gồm tổng từng phần tử, trung bình hoặc cực đại.

It has been demonstrated that GNNs cannot be more expensive than the **Weisfeiler–Leman Graph Isomorphism Test**. In practice, this means that there exist different graph structures (e.g., molecules with the same atoms but different bonds) that cannot be distinguished by GNNs. More powerful GNNs operating on higher-dimension geometries e.g. simplicial complexes can be designed. As of 2022, whether or not future architectures will overcome the message passing primitive is an open research question.

– Người ta đã chứng minh rằng GNN không thể đắt hơn Phép thử Đồng cấu Đồ thị Weisfeiler-Leman. Trên thực tế, điều này có nghĩa là tồn tại các cấu trúc đồ thị khác nhau (ví dụ: các phân tử có cùng nguyên tử nhưng liên kết khác nhau) mà GNN không thể phân biệt được. Có thể thiết kế các GNN mạnh hơn hoạt động trên các hình học chiều cao hơn, ví dụ như các phức hợp đơn giản. Tính đến năm 2022, liệu các kiến trúc trong tương lai có vượt qua được nguyên thủy truyền tin hay không vẫn là một câu hỏi nghiên cứu chưa có lời giải.

1.3 Message Passing Layers – Các Lớp Truyền Tin Nhắn

Message passing layers are permutation-equivariant layers mapping a graph into an updated representation of the same graph. Formally, they can be expressed as message passing neural networks (MPNNs).

– Các lớp truyền thông điệp là các lớp hoán vị tương đương biến thể, ánh xạ một đồ thị thành một biểu diễn được cập nhật của cùng đồ thị đó. Về mặt hình thức, chúng có thể được biểu diễn dưới dạng mạng nơ-ron truyền thông điệp (MPNN).

Definition 1 (MPNN layer). *Let $G = (V, E)$ be a graph, where V is the node set & E is the edge set. Let N_u be the neighborhood of some node $u \in V$. Additionally, let \mathbf{x}_u be the features of node $u \in V$, & \mathbf{e}_{uv} be the features of edge $(u, v) \in E$. An MPNN layer can be expressed as follows:*

$$\mathbf{h}_u = \phi \left(\mathbf{x}_u, \bigoplus_{v \in N_u} \psi(\mathbf{x}_u, \mathbf{x}_v, \mathbf{e}_{uv}) \right),$$

where ϕ, ψ are differentiable functions (e.g., artificial neural networks), & \bigoplus is a permutation invariant **aggregation operator** that can accept an arbitrary number of inputs (e.g., element-wise sum, mean, or max). In particular, ϕ, ψ are referred to as update & message functions, respectively.

Định nghĩa 1 (Lớp mạng nơ-ron truyền tin). Cho $G = (V, E)$ là một đồ thị, trong đó V là tập nút & E là tập cạnh. Cho N_u là lân cận của một nút $u \in V$. Ngoài ra, cho \mathbf{x}_u là các đặc trưng của nút $u \in V$, & \mathbf{e}_{uv} là các đặc trưng của cạnh $(u, v) \in E$. Một lớp MPNN có thể được biểu diễn như sau:

$$\mathbf{h}_u = \phi \left(\mathbf{x}_u, \bigoplus_{v \in N_u} \psi(\mathbf{x}_u, \mathbf{x}_v, \mathbf{e}_{uv}) \right),$$

trong đó ϕ, ψ là các hàm khả vi (ví dụ: mạng nơ-ron nhân tạo), & \bigoplus là một toán tử tổng hợp bất biến hoán vị có thể chấp nhận một số lượng đầu vào tùy ý (ví dụ: tổng từng phần tử, trung bình hoặc cực đại). Cụ thể, ϕ, ψ được gọi tương ứng là các hàm cập nhật & tin nhắn.

Intuitively, in an MPNN computational block, graph nodes *update* their representations by *aggregating* the *messages* received from their neighbors.

– Theo trực giác, trong khối tính toán MPNN, các nút đồ thị *cập nhật* biểu diễn của chúng bằng cách *tổng hợp* các *thông điệp* nhận được từ các nút lân cận của chúng.

The outputs of 1 or more MPNN layers are node representations \mathbf{h}_u for each node $u \in V$ in the graph. Node representations can be employed for any downstream task, e.g. node/graph classification or edge prediction.

– Đầu ra của 1 hoặc nhiều lớp MPNN là biểu diễn nút \mathbf{h}_u cho mỗi nút $u \in V$ trong đồ thị. Biểu diễn nút có thể được sử dụng cho bất kỳ tác vụ hạ nguồn nào, ví dụ: phân loại nút/đồ thị hoặc dự đoán cạnh.

Graph nodes in an MPNN update their representation aggregation information from their immediate neighbors. As such, stacking n MPNN layers means that 1 node will be able to communicate with nodes that are at most n “hops” away. In principle, to ensure that every node receives information from every other node, one would need to stack a number of MPNN layers equal to the graph diameter. However, stacking many MPNN layers may cause issues e.g. oversmoothing & oversquashing. Oversmoothing refers to issue of node representations becoming indistinguishable. Oversquashing refers to the bottleneck that is created by squeezing long-range dependencies into fixed-size representations. Countermeasures e.g. skip connections (as in residual neural networks), gated updated rules & jumping knowledge can mitigate oversmoothing. Modifying the final layer to be a fully-adjacent layer, i.e., by considering the graph as a complete graph, can mitigate oversquashing in problems where long-range dependencies are required.

– Các nút đồ thị trong MPNN cập nhật thông tin tổng hợp biểu diễn của chúng từ các nút lân cận trực tiếp. Do đó, việc xếp chồng n lớp MPNN có nghĩa là 1 nút sẽ có thể giao tiếp với các nút cách xa tối đa n “bước nhảy”. Về nguyên tắc, để đảm bảo rằng mọi nút đều nhận được thông tin từ mọi nút khác, người ta cần xếp chồng một số lớp MPNN bằng với đường kính đồ thị. Tuy nhiên, việc xếp chồng nhiều lớp MPNN có thể gây ra các vấn đề, ví dụ như làm mịn quá mức & đè bẹp quá mức. Làm mịn quá mức đề cập đến vấn đề biểu diễn nút trở nên không thể phân biệt được. Đè bẹp quá mức đề cập đến nút thất cổ chai được tạo ra bằng cách nén các phụ thuộc tầm xa vào các biểu diễn có kích thước cố định. Các biện pháp đối phó, ví dụ như kết nối bỏ qua (như trong mạng nơ-ron dư), các quy tắc cập nhật có cổng & kiến thức nhảy có thể giảm thiểu việc làm mịn quá mức. Việc sửa đổi lớp cuối cùng thành một lớp hoàn toàn liền kề, tức là bằng cách coi đồ thị là một đồ thị hoàn chỉnh, có thể giảm thiểu việc đè bẹp quá mức trong các vấn đề yêu cầu các phụ thuộc tầm xa.

Other “flavors” of MPNN have been developed in the literature, e.g. graph convolutional networks & graph attention networks, whose definitions can be expressed in terms of the MPNN formalism.

– Các “hương vị” khác của MPNN đã được phát triển trong tài liệu, ví dụ như mạng tích chập đồ thị & mạng chú ý đồ thị, định nghĩa của chúng có thể được thể hiện theo hình thức MPNN.

1.3.1 Graph convolutional network (GCN) – Mạng lưới tích chập đồ thị

The graph convolutional network (GCN) was 1st introduced by THOMAS KIPF & MAX WELLING in 2017. A GCN layer defines a 1st-order approximation of a localized spectral filter on graphs. GCNs can be understood as a generalization of convolutional neural networks to graph-structured data.

– Mạng tích chập đồ thị (GCN) được giới thiệu lần đầu tiên bởi THOMAS KIPF & MAX WELLING vào năm 2017. Một lớp GCN định nghĩa phép xấp xỉ bậc 1 của bộ lọc phổ cục bộ trên đồ thị. GCN có thể được hiểu là sự tổng quát hóa của mạng nơ-ron tích chập lên dữ liệu có cấu trúc đồ thị.

The formal expression of a GCN layer reads as follows:

$$\mathbf{H} = \sigma \left(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{X} \Theta \right),$$

where \mathbf{H} is the matrix of node representations \mathbf{h}_u , \mathbf{X} is the matrix of node features \mathbf{x}_u , $\sigma(\cdot)$ is an activation function (e.g., ReLU), $\tilde{\mathbf{A}}$ is the graph adjacency matrix with the addition of self-loops, $\tilde{\mathbf{D}}$ is the graph degree matrix with the addition of self-loops, & Θ is a matrix of trainable parameters.

– Biểu thức chính thức của một lớp GCN được đọc như sau:

$$\mathbf{H} = \sigma \left(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{X} \Theta \right),$$

trong đó \mathbf{H} là ma trận biểu diễn nút \mathbf{h}_u , \mathbf{X} là ma trận đặc trưng nút \mathbf{x}_u , $\sigma(\cdot)$ là một hàm kích hoạt (ví dụ: ReLU), $\tilde{\mathbf{A}}$ là ma trận kề đồ thị với việc bổ sung các vòng lặp tự thân, $\tilde{\mathbf{D}}$ là ma trận bậc đồ thị với việc bổ sung các vòng lặp tự thân, & Θ là ma trận các tham số có thể đào tạo được.

In particular, let \mathbf{A} be the graph adjacency matrix: then, one can define $\tilde{\mathbf{A}} := \mathbf{A} + \mathbf{I}$ & $\tilde{\mathbf{D}}_{ii} = \sum_{j \in V} \tilde{A}_{ij}$, where \mathbf{I} denotes identity matrix. This normalization ensures: eigenvalues of $\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$ are bounded in range $[0, 1]$, avoiding numerical instabilities & exploding/vanishing gradients.

– Cụ thể, giả sử \mathbf{A} là ma trận kề đồ thị: khi đó, ta có thể định nghĩa $\tilde{\mathbf{A}} := \mathbf{A} + \mathbf{I}$ & $\tilde{\mathbf{D}}_{ii} = \sum_{j \in V} \tilde{A}_{ij}$, trong đó \mathbf{I} biểu thị ma trận đơn vị. Chuẩn hóa này đảm bảo: các trị riêng của $\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$ bị chặn trong khoảng $[0, 1]$, tránh được hiện tượng bất ổn định số & bùng nổ/gradient biến mất.

A limitation of GCNs: they do not allow multidimensional edge features \mathbf{e}_{uv} . It is however possible to associate scalar weight w_{uv} to each edge by imposing $A_{uv} = w_{uv}$, i.e., by setting each nonzero entry in adjacency matrix equal to weight of corresponding edge.

– 1 hạn chế của GCN: chúng không cho phép các đặc trưng cạnh đa chiều \mathbf{e}_{uv} . Tuy nhiên, có thể liên kết trọng số vô hướng w_{uv} với mỗi cạnh bằng cách áp đặt $A_{uv} = w_{uv}$, tức là bằng cách đặt mỗi phần tử khác không trong ma trận kề bằng với trọng số của cạnh tương ứng.

1.3.2 Graph attention network – Mạng lưới chú ý đồ thị

Graph attention network (GAT) was introduced by PETAR VELIČKOVIĆ et al. in 2018. Graph attention network is a combination of a GNN & an attention layer. Implementation of attention layer in graphical neural networks helps provide attention or focus to important information from data instead of focusing on whole data.

– Mạng lưới chú ý đồ thị (GAT) được giới thiệu bởi PETAR VELIČKOVIĆ & cộng sự vào năm 2018. Mạng lưới chú ý đồ thị là sự kết hợp giữa mạng nơ-ron nhân tạo (GNN) và một lớp chú ý. Việc triển khai lớp chú ý trong mạng nơ-ron đồ họa giúp tập trung sự chú ý hoặc tập trung vào thông tin quan trọng từ dữ liệu thay vì tập trung vào toàn bộ dữ liệu.

A multi-head GAT layer can be expressed as follows:

$$\mathbf{h}_u = \left\| \sigma \left(\sum_{v \in N_u} \alpha_{uv} \mathbf{W}^k \mathbf{x}_v \right) \right\|_{k=1}^K,$$

where K is the number of attention heads, $\|$ denotes vector concatenation, $\sigma(\cdot)$ is an activation function (e.g., ReLU), α_{ij} are attention coefficients, & \mathbf{W}^k is a matrix of trainable parameters for the k th attention head.

– 1 lớp GAT nhiều đầu có thể được biểu diễn như sau:

$$\mathbf{h}_u = \left\| \sigma \left(\sum_{v \in N_u} \alpha_{uv} \mathbf{W}^k \mathbf{x}_v \right) \right\|_{k=1}^K,$$

trong đó K là số đầu chú ý, $\|$ biểu thị phép nối vectơ, $\sigma(\cdot)$ là hàm kích hoạt (ví dụ: ReLU), α_{ij} là các hệ số chú ý, & \mathbf{W}^k là ma trận các tham số có thể huấn luyện được cho đầu chú ý thứ k .

+++

1.3.3 Gated graph sequence neural network – Mạng nơ-ron chuỗi đồ thị có cổng

1.4 Local Pooling Layers – Lớp Gộp Cục Bộ

1.5 Heterophilic Graph Learning – Học Đồ Thị Dị Hướng

1.6 Applications of GNNs – Ứng Dụng của GNNs

Chương 2

Miscellaneous

Contents

2.1 Contributors	8
----------------------------	---

2.1 Contributors

1. VÕ NGỌC TRÂM ANH [VNTA].
2. ĐẶNG PHÚC AN KHANG [DPAK]
3. NGUYỄN NGỌC THẠCH [NNT].
4. PHAN VĨNH TIẾN [PVT].