



Graph Neural Network

Table of **contents**

01

Introduction

02

Gragh neural network

03

**Application of Gragh
neural network**

04

**Gragh convolutional
network**

05

Graphsage

06

Bài tập



Introduction

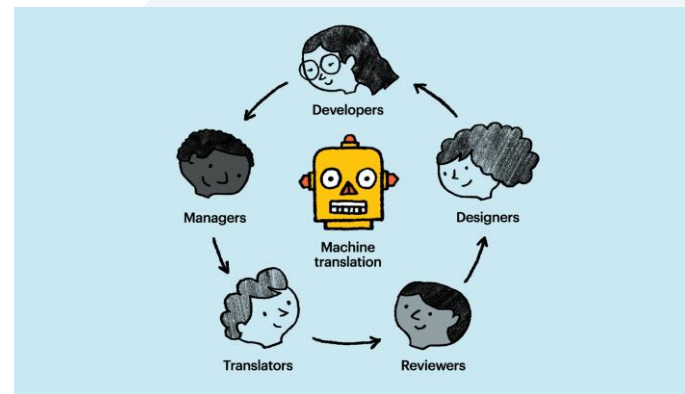


Thành công gần đây của mạng thần kinh đã thúc đẩy nghiên cứu về nhận dạng mẫu và khai thác dữ liệu.

Các nhiệm vụ học máy, như phát hiện đối tượng, dịch máy và nhận dạng giọng nói, đã được mang lại sức sống mới với các mô hình học sâu từ đầu đến cuối như CNN, RNN.

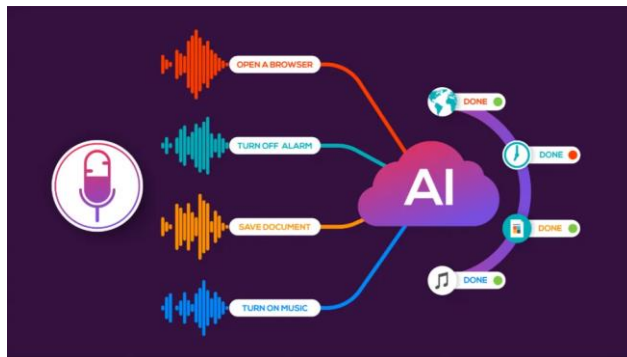


Object detection



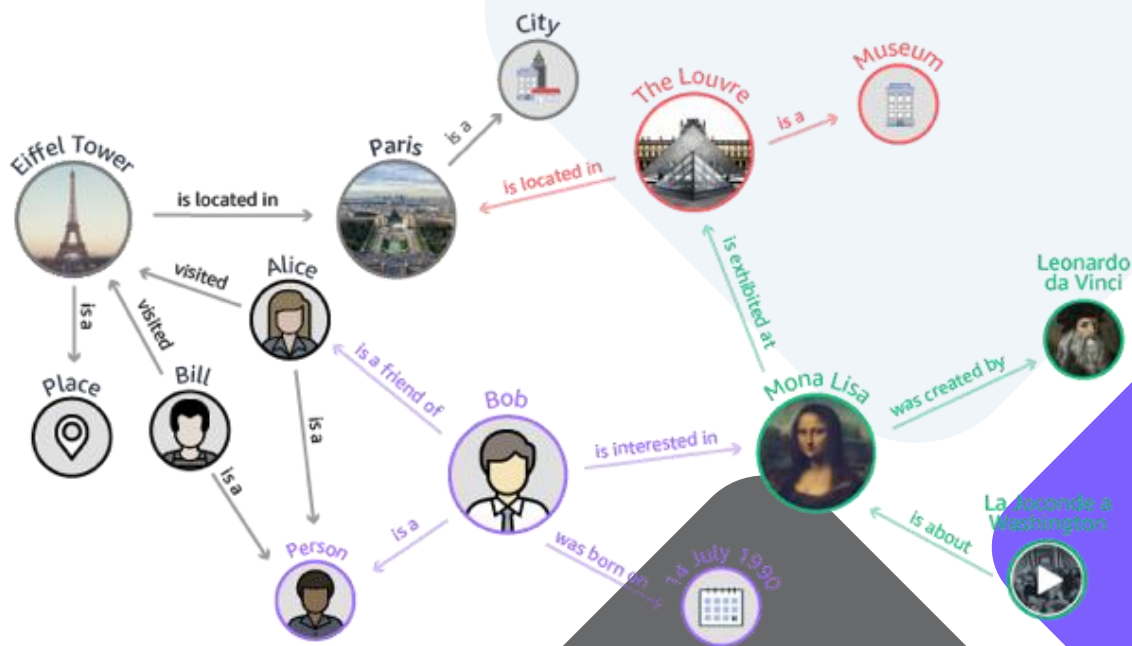
Machine translation

Speech
recognition



Deep Learning rất giỏi trong việc nắm bắt các mẫu dữ liệu Euclide ản (hình ảnh, văn bản, video).

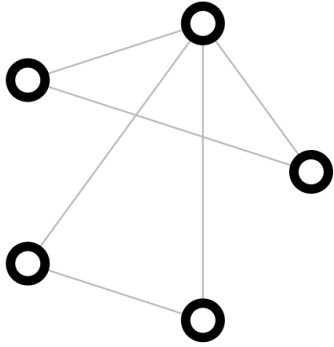
Nhưng còn những dữ liệu được tạo ra từ các miền không thuộc Euclide, được biểu diễn dưới dạng đồ thị với các mối quan hệ phức tạp và có sự phụ thuộc lẫn nhau giữa các đối tượng thì sao?



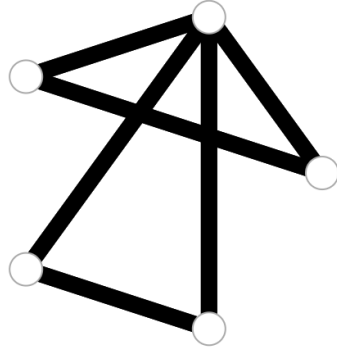
⇒ Graph neural network



Graph theory

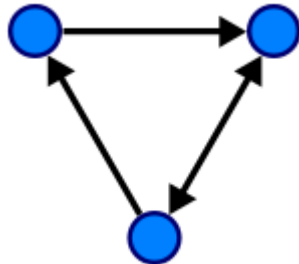


V là tập hợp các **nút**
của đồ thị



E là tập hợp các **cạnh** kết
nối các nút của đồ thị

- Đồ thị được biểu diễn dưới dạng $G = (V, E)$
- G là đồ thị được cấu thành
- Kí hiệu $N(v) = \{u \in V \mid (v, u) \in E\}$ là các nút kề u với v



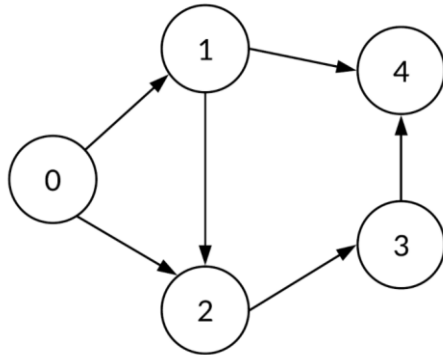
Nếu có sự phụ thuộc hướng giữa các nút thì các cạnh được định hướng. Nếu không, các cạnh là vô hướng

Graph theory

Đồ thị có thể biểu thị những thứ như mạng truyền thông xã hội hoặc phân tử. Với các nút là người dùng và các cạnh là kết nối

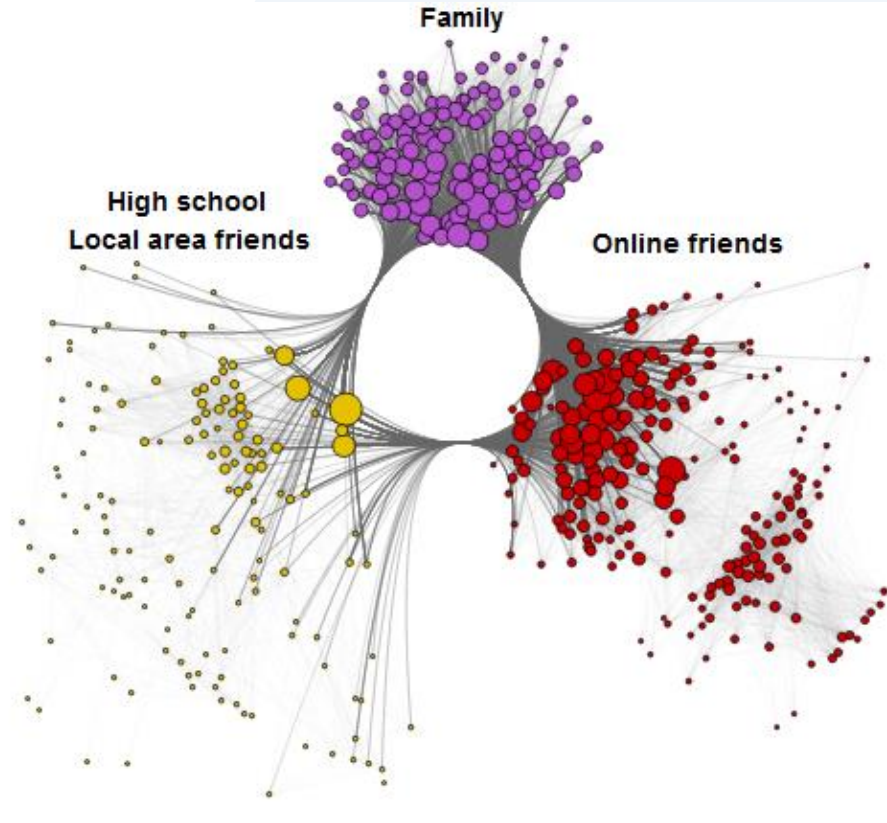


Một đồ thị thường được biểu diễn bằng A, một ma trận kề.
Nếu đồ thị có n nút thì A có số chiều là $(n \times n)$.



Adjacency Matrix

	0	1	2	3	4
0	0	1	1	0	0
1	0	0	1	0	1
2	0	0	0	1	0
3	0	0	0	0	1
4	0	0	0	0	0

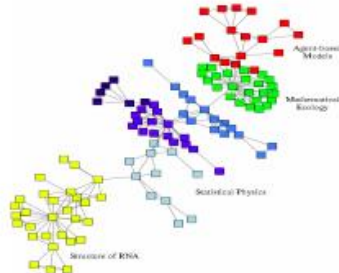




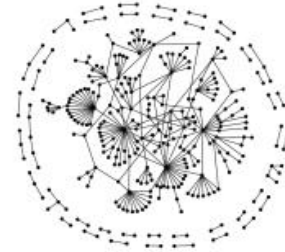
Graph theory



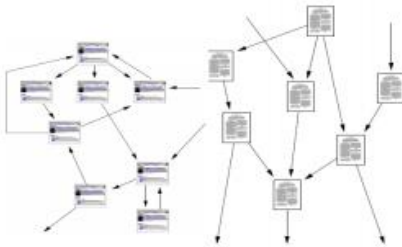
Social networks



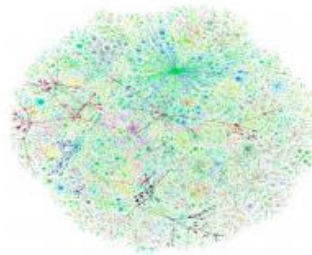
Economic networks



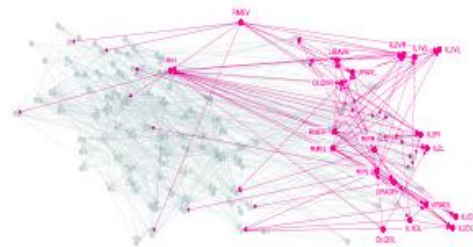
Communication networks



Information networks:
Web & citations



Internet



Networks of neurons



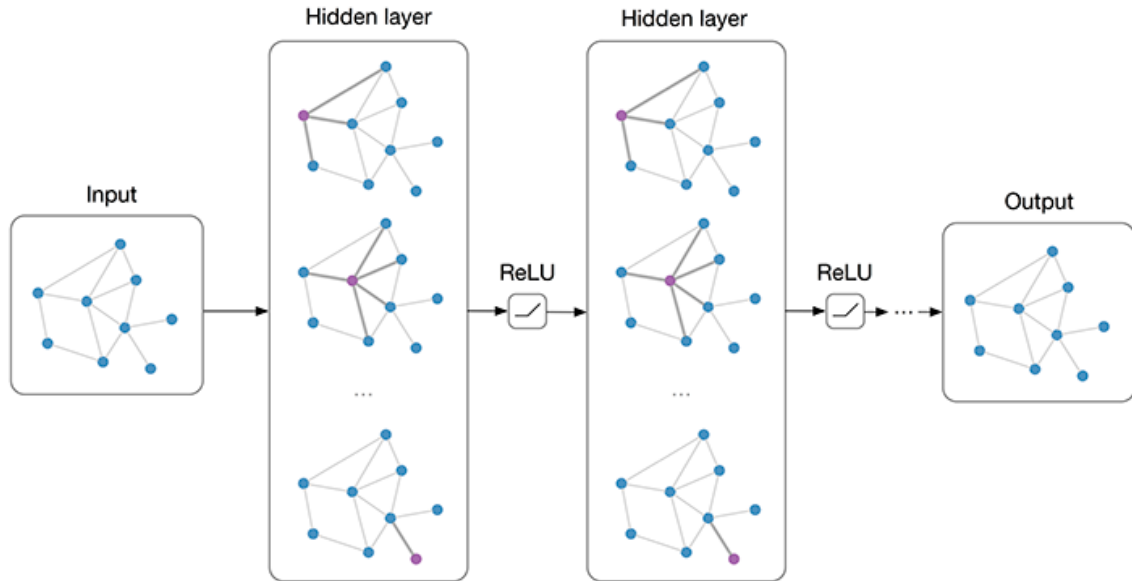
02

Graph Neural Network

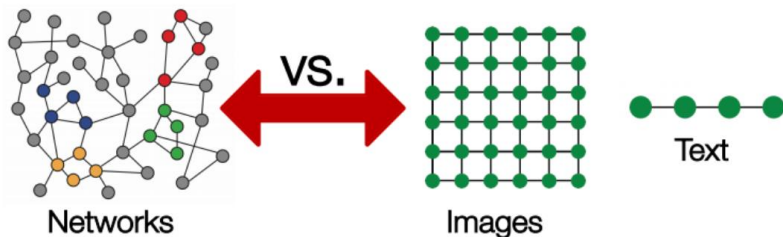
Graph Neural Network

Graph neural network (GNN) là một lớp của machine learning được thiết kế để thực hiện suy luận trên dữ liệu được mô tả bằng biểu đồ.

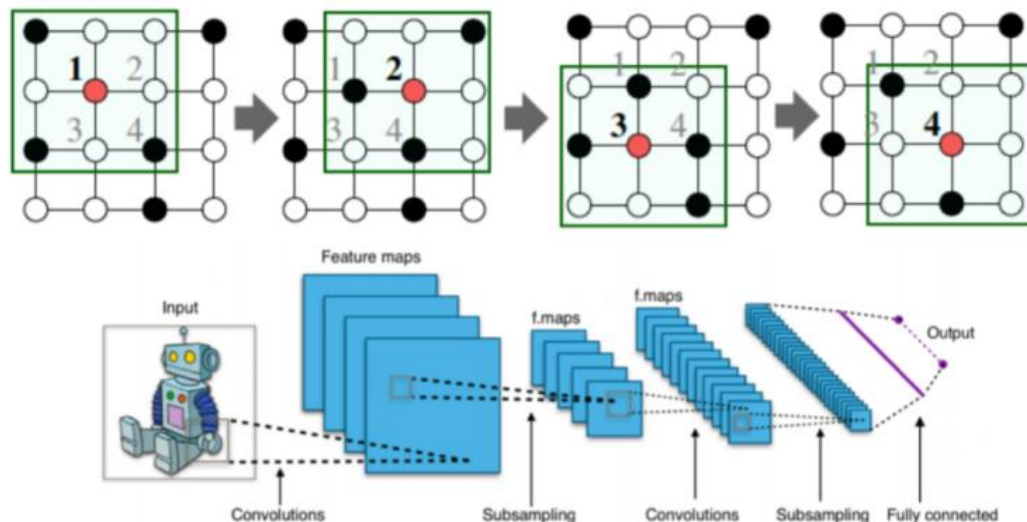
GNN là mạng thần kinh có thể được áp dụng trực tiếp cho biểu đồ và cung cấp một cách dễ dàng để thực hiện các tác vụ dự đoán ở cấp độ nút, cấp độ cạnh và cấp độ biểu đồ.



Why is Graph Neural Network?



Analyze a graph



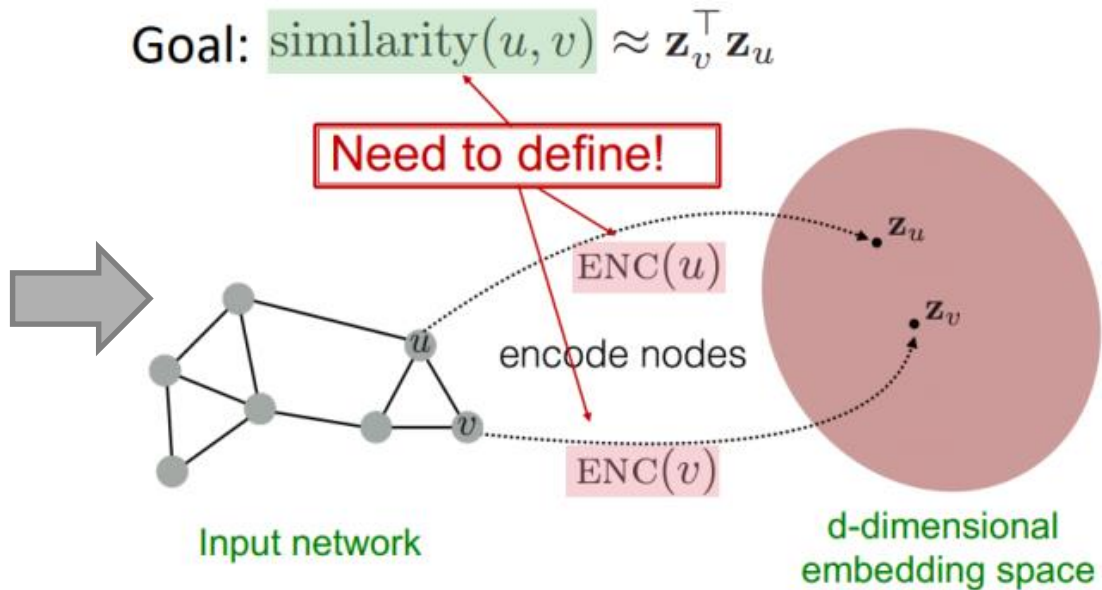
Convolutional Neural Networks (CNNs)

Node embedding

Xác định u và v là hai nút trong biểu đồ.

x_u và x_v là hai vector đặc trưng.

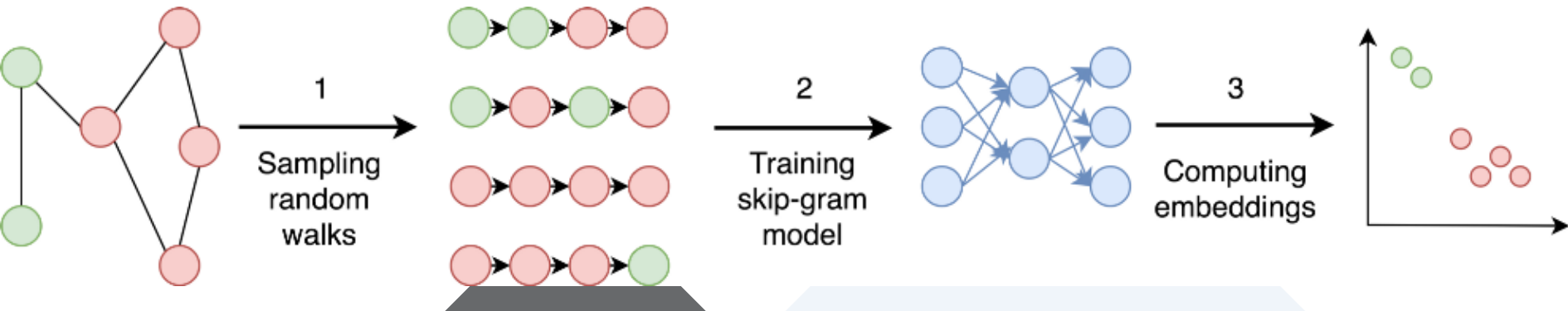
Định nghĩa hàm mã hóa $\text{Enc}(u)$ và $\text{Enc}(v)$, hàm này chuyển đổi các vector đặc trưng thành z_u và z_v .



Node embedding

DeepWalk là 1 mô hình **Node Embedding**, dựa trên ý tưởng chủ đạo từ **Word2Vec**, mà cụ thể là từ mô hình Skip-gram.

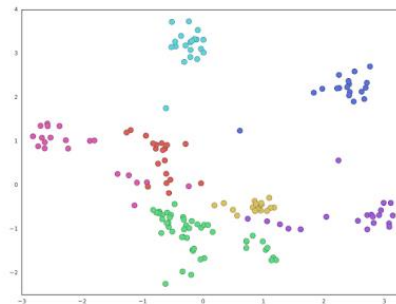
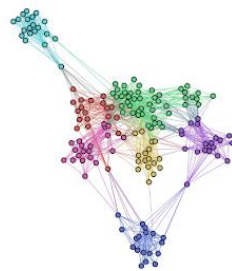
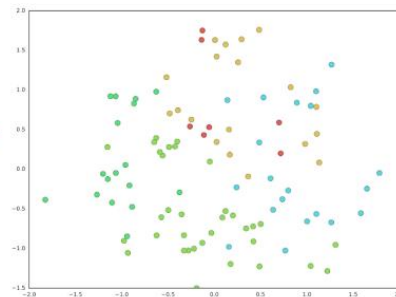
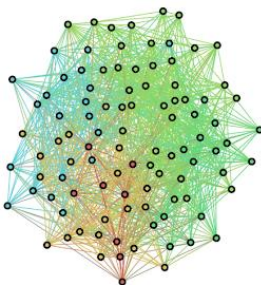
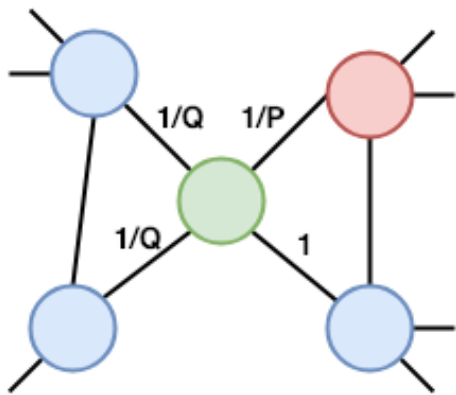
Bằng việc sampling sử dụng **random walk**, ta đã chuyển dữ liệu từ 1 dạng cấu trúc phức tạp là đồ thị sang dạng biểu diễn sequence 1D



Node2Vec

Node2Vec cũng là 1 mô hình Node Embedding dựa trên ý tưởng của **DeepWalk** và **Word2Vec**

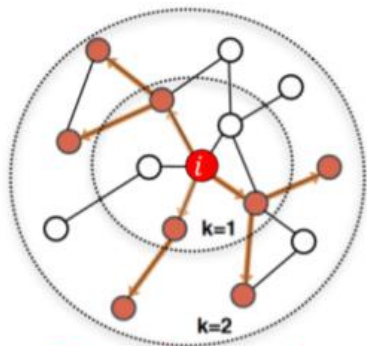
Điểm khác biệt của Node2Vec là ngoài việc sử dụng **random walk** như thông thường, mô hình giới thiệu thêm 2 thông số **P** và **Q** để điều chỉnh lại bước đi ngẫu nhiên trên đồ thị.



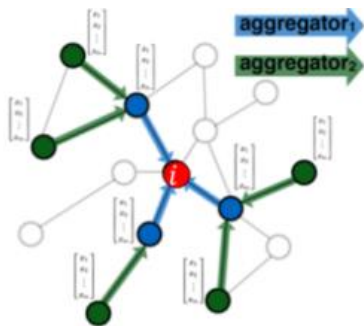
Neighborhood Aggregation



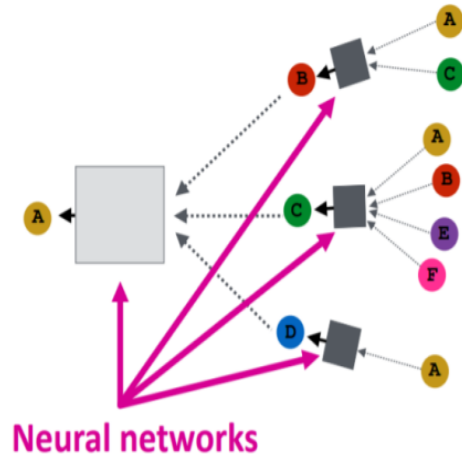
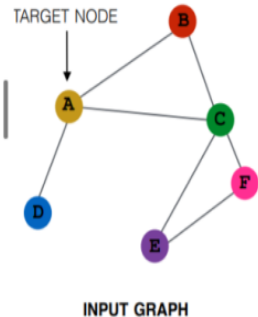
Key idea: Generate node embeddings based on local neighborhood



Determine node computation graph

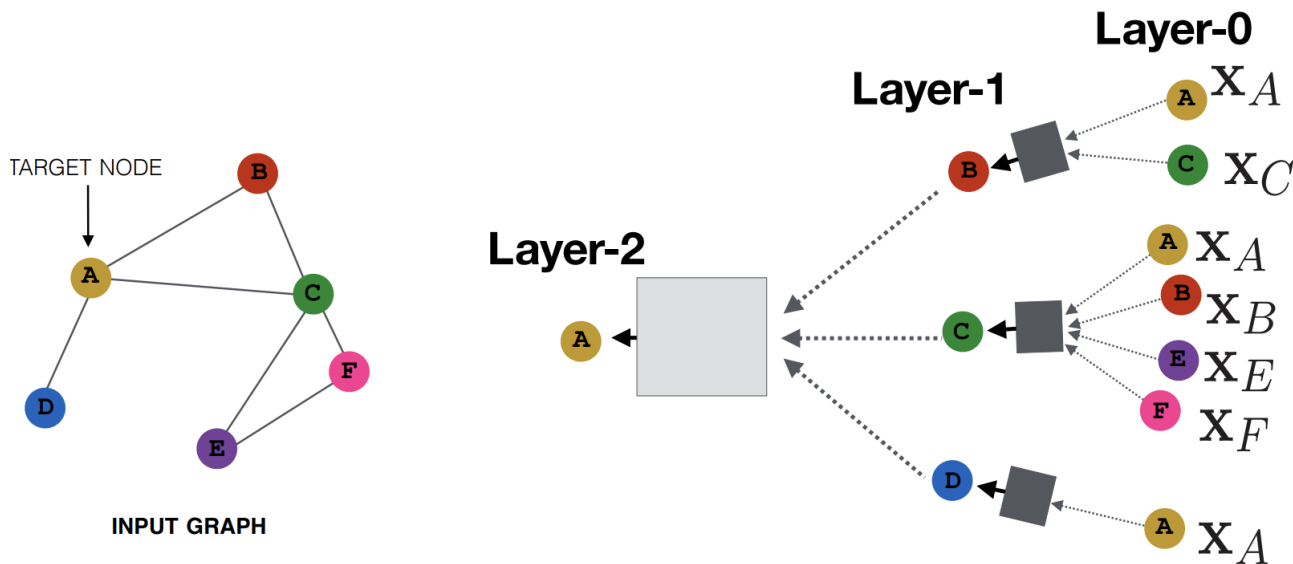


Propagate and transform information



Neighborhood Aggregation

- Node have embeddings each layer
- Layer 0 embedding of node u is its input feature



Simple neighborhood aggregation

- **Basic approach:** Average neighbor messages and apply a neural network

Initial layer 0 embeddings are equal to node feature

$$h_v^0 = X_v$$

$$h_v^k$$

$$= \sigma (W_k$$

$$\sum_{u \in N(v)} \frac{h_u^{k-1}}{|N(v)|}$$

$$+ B_k$$

$$h_u^{k-1})$$

$$, \forall k > 0$$

K^{th} layer embeddings of v

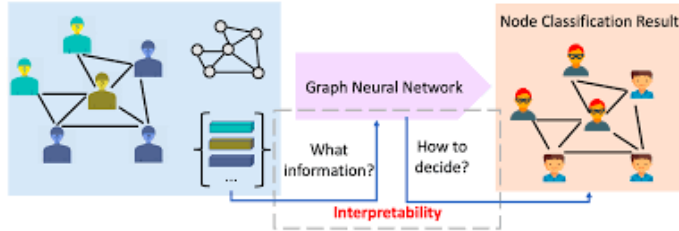
Average of neighbor previous layer embeddings

previous layer embeddings of v

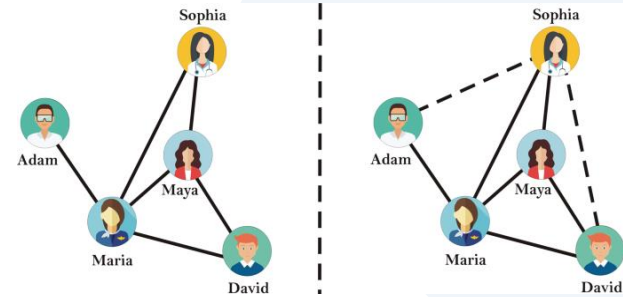
03

Application of GNNs

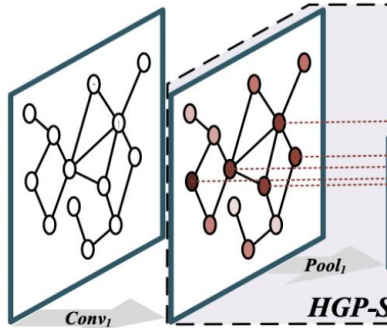
Application of GNNs



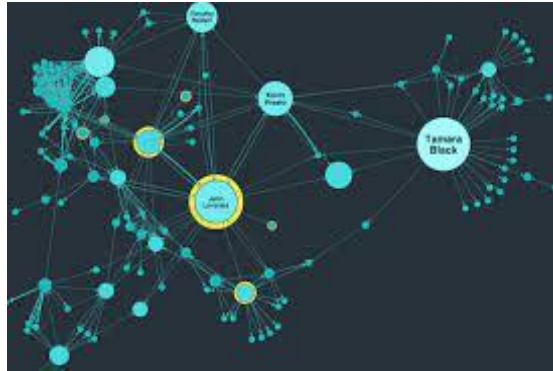
Node classification



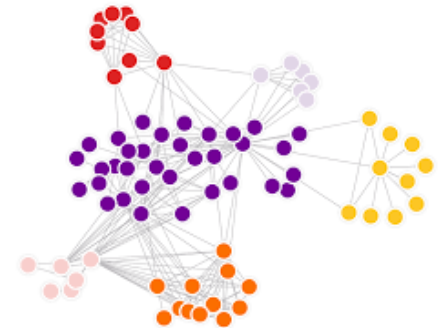
Link prediction



Graph classification

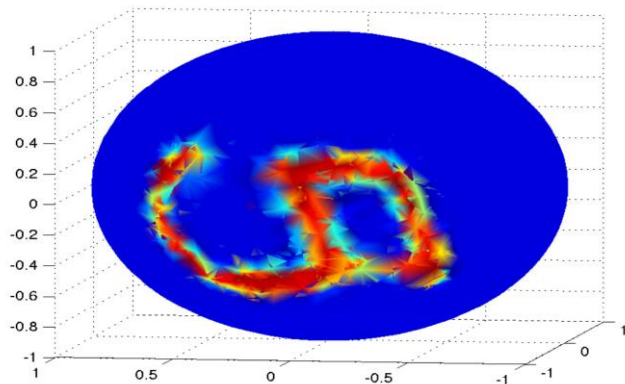


Graph visualization

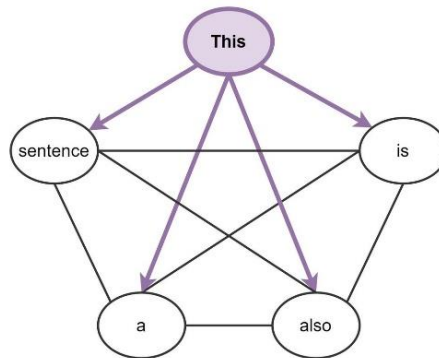


Graph clustering

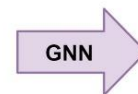
Application of **GNNs**



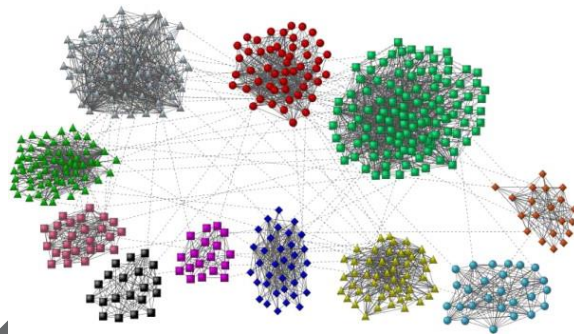
GNNs in computer vision



Natural Language Processing



- Translation?
- Sentiment?
- Next word?
- Part-of-speech tags?



Clustering & Community detection



Disadvantages of **GNNs**

The most basic limitations of GNN are:

- * Works of GNN limited to a fixed number of points,
- * Time and space complexity are higher.
- * Less handling of edges of graphs based on their types and relations

04

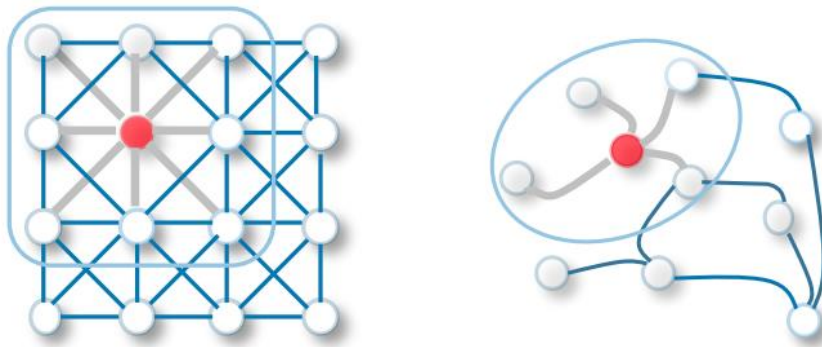
Graph Convolutional Network

Graph Convolutional Network

Graph Convolutional Network (GCN) là một mô hình học sâu được sử dụng để xử lý dữ liệu trên đồ thị.

GCN có thể học được các đặc trưng của các đỉnh và cạnh trong đồ thị, giúp cho việc phân loại, dự đoán và nhận dạng trên đồ thị trở nên dễ dàng hơn.

Key idea: GCN sử dụng các phép tính convolution trên đồ thị để học các đặc trưng của các đỉnh và cạnh trong đồ thị.



Graph Convolutional Network

GCN sử dụng ma trận trọng số để tính toán đặc trưng của các đỉnh trong đồ thị.

Các đặc trưng của các đỉnh được tính toán bằng cách lấy trung bình có trọng số của các đặc trưng của các đỉnh kề cận.

Các đặc trưng của các cạnh trong đồ thị cũng được tính toán bằng cách lấy trung bình có trọng số của các đặc trưng của các đỉnh kề cận.

$$\text{GCN: } \mathbf{h}_v^k = \sigma \left(\mathbf{W}_k \sum_{u \in N(v) \cup v} \frac{\mathbf{h}_u^{k-1}}{\sqrt{|N(u)| |N(v)|}} \right)$$

$$\text{GNN: } \mathbf{h}_v^k = \sigma \left(\mathbf{W}_k \sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{|N(v)|} + \mathbf{B}_k \mathbf{h}_v^{k-1} \right), \forall k > 0$$



Graph Convolutional Network

- 1 hidden layer có thể được biểu diễn như sau: $H^{(k+1)} = f(H^k, A)$
- Trong đó hàm f có thể biểu diễn đơn giản bằng công thức sau: $H^{(k+1)} = \sigma(D^{-\frac{1}{2}} A D^{-\frac{1}{2}} H^k W_k)$



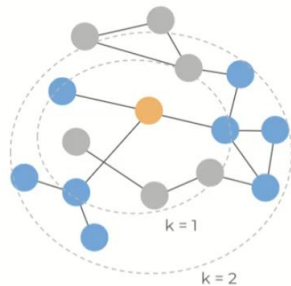
05

GraphSage

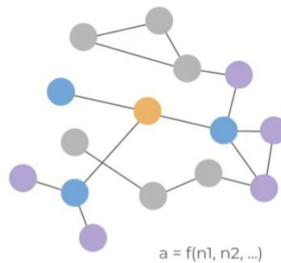
GraphSAGE

GraphSAGE = Graph + Sample + Aggregate

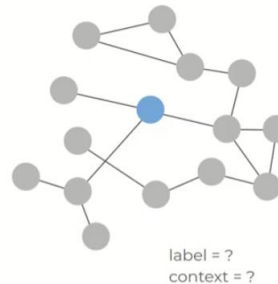
Sample



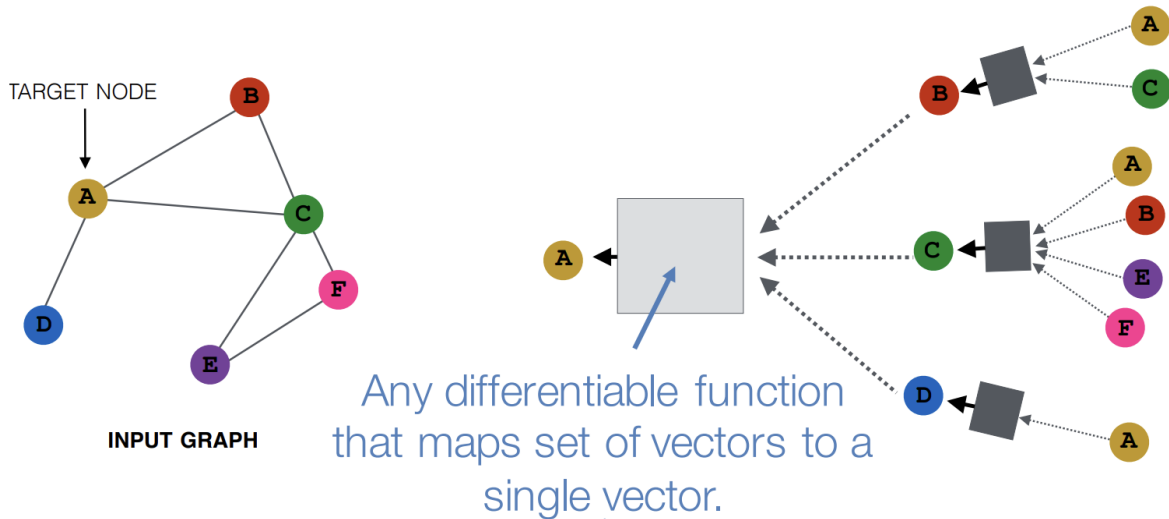
Aggregate



Predict



GraphSAGE



$$h_v^k = \sigma(W_k \cdot AGG(\{h_u^{k-1}, \forall u \in N(v)\}), h_v^{k-1})$$

GraphSAGE

- Input: đồ thị $G = (V, E)$, các node feature $\mathbf{x}_v \in X, \forall v \in V, \forall k \in 1, \dots, K$, với K là số aggregator function được áp dụng liên tiếp nhau
- Output: embedding vector $\mathbf{z}_v, \forall v \in V$

```
1  $\mathbf{h}_v^0 \leftarrow \mathbf{x}_v, \forall v \in \mathcal{V};$   
2 for  $k = 1 \dots K$  do  
3   for  $v \in \mathcal{V}$  do  
4      $\mathbf{h}_{\mathcal{N}(v)}^k \leftarrow \text{AGGREGATE}_k(\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\});$   
5      $\mathbf{h}_v^k \leftarrow \sigma(\mathbf{W}^k \cdot \text{CONCAT}(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k))$   
6   end  
7    $\mathbf{h}_v^k \leftarrow \mathbf{h}_v^k / \|\mathbf{h}_v^k\|_2, \forall v \in \mathcal{V}$   
8 end  
9  $\mathbf{z}_v \leftarrow \mathbf{h}_v^K, \forall v \in \mathcal{V}$ 
```

GraphSAGE Variants

Mean

$$\text{AGG} = \sum_{u \in N(v)} \frac{h_u^{k-1}}{|N(v)|}$$

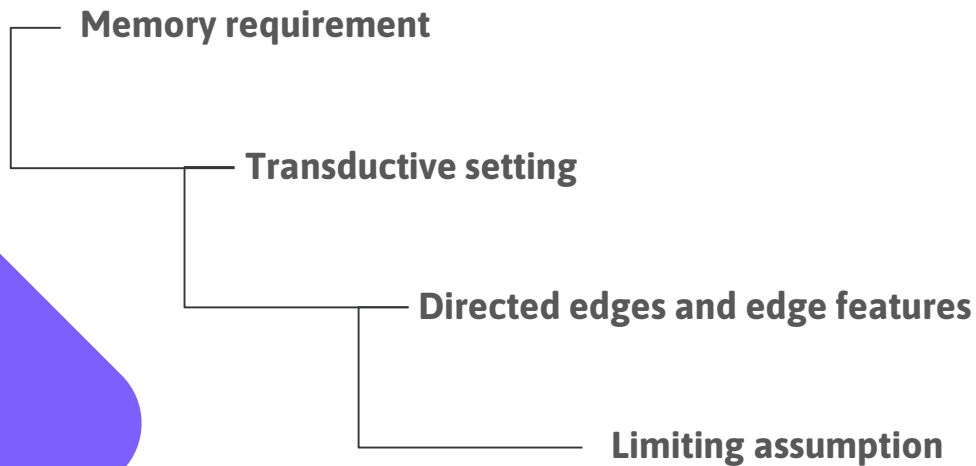
Pool

$$\text{AGG} = \gamma(\{Qh_u^{k-1}, \forall u \in N(n)\})$$

LSTM

$$\text{AGG} = \text{LSTM}([h_u^{k-1}, \forall u \in \pi(N(n))])$$

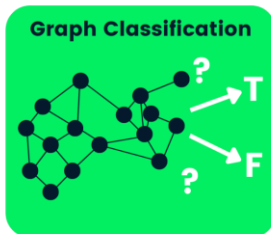
Disadvantages of Graph Neural Networks



Types of Graph Neural Networks Tasks

Graph
Classification

1

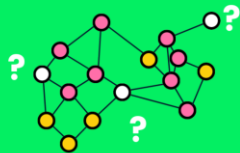


Node
Classification

2



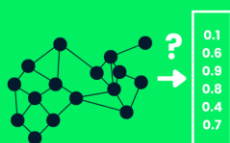
Node Classification



Link Prediction



Graph Embedding



Graph Generation



Link Prediction

3

6

Graph
Generation

5

Graph
Embedding

4

Community
Detection



Bài Tập



1

Hãy giải thích cách GNN áp dụng đặc trưng của các đỉnh và cạnh để tính toán đặc trưng của một đỉnh trong đồ thị.

Cách GNN áp dụng đặc trưng của các đỉnh và cạnh để tính toán đặc trưng của một đỉnh trong đồ thị là sử dụng một mạng nơ-ron để tính toán đặc trưng của mỗi đỉnh dựa trên đặc trưng của đỉnh đó và các đặc trưng của các đỉnh và cạnh kề. Quá trình tính toán đặc trưng này được thực hiện thông qua các lớp của GNN

2

Hãy nêu ví dụ về các loại lớp GNN khác nhau và giải thích sự khác nhau giữa chúng.

Graph Convolutional Network (GCN): Sử dụng tích chập đồ thị để lấy trung bình trọng số của các đặc trưng của các đỉnh và cạnh kề để tính toán đặc trưng của mỗi đỉnh.

GraphSAGE: Sử dụng phép tổng hợp cục bộ (local aggregation) để lấy thông tin đặc trưng từ các đỉnh và cạnh kề để tính toán đặc trưng của mỗi đỉnh.

Gated Graph Neural Network (GGNN): Sử dụng cổng đầu vào để xác định độ quan trọng của các đỉnh và cạnh kề để tính toán đặc trưng của mỗi đỉnh.

3

Hãy giải thích cách GNN giải quyết vấn đề về sự thay đổi kích thước đồ thị (graph size) trong quá trình huấn luyện.

GNN giải quyết vấn đề về sự thay đổi kích thước đồ thị trong quá trình huấn luyện bằng cách sử dụng các lớp chuyển đổi (pooling, unpooling) để điều chỉnh kích thước đồ thị.