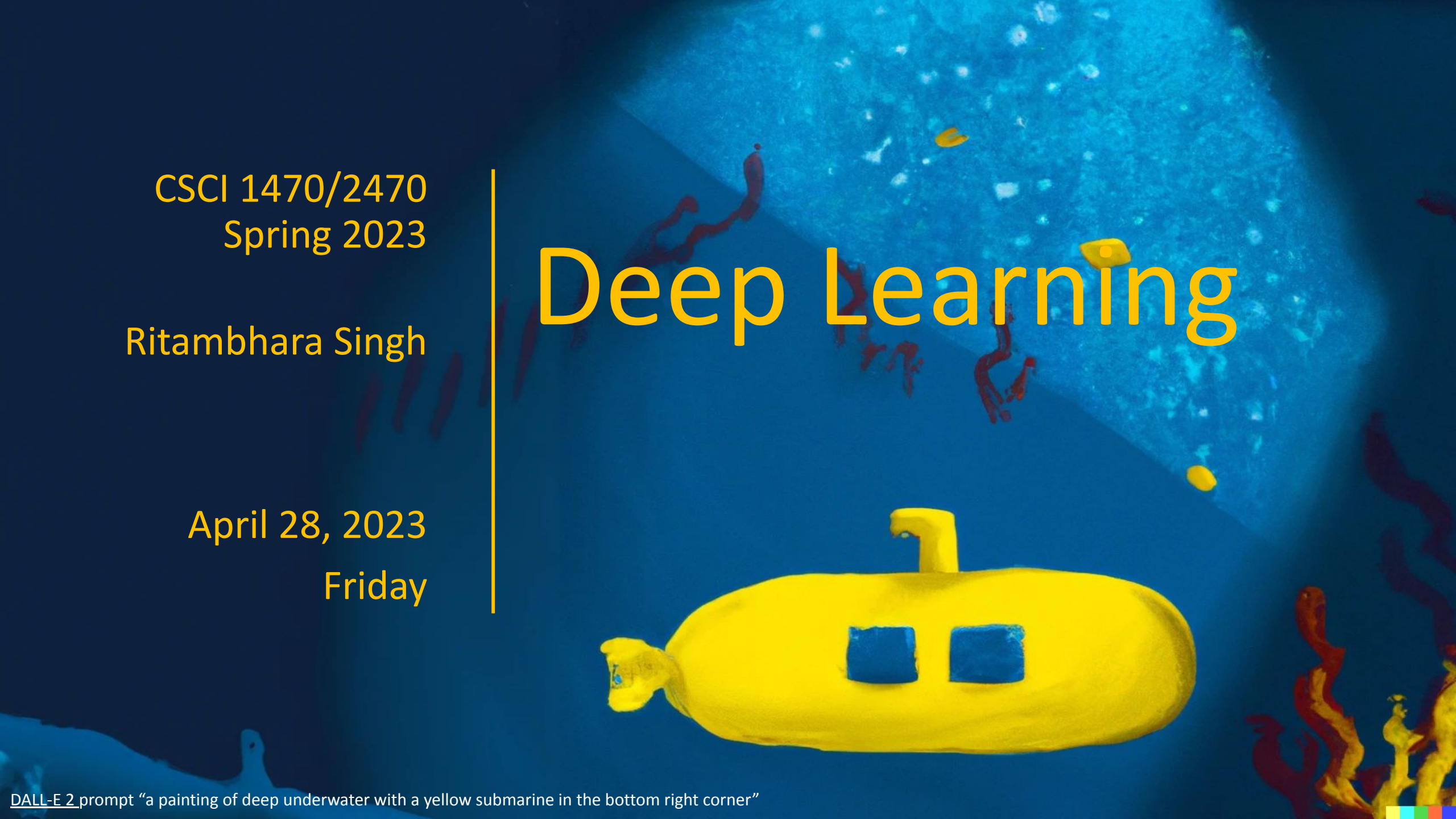
CSCI 1470/2470
Spring 2023

Ritambhara Singh

April 28, 2023

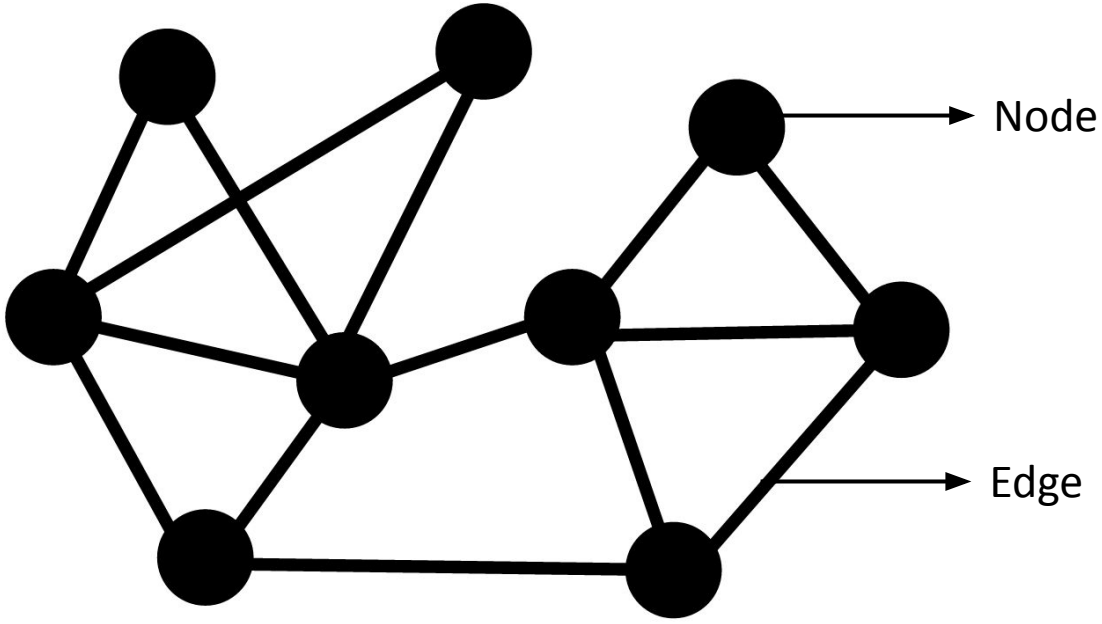
Friday

Deep Learning



DALL-E 2 prompt “a painting of deep underwater with a yellow submarine in the bottom right corner”

General language for describing entities with relations/interactions



Graph

Slide courtesy: http://web.stanford.edu/class/cs224w/slides/01-intro.pdf

Majority of real-world data can be described using graphs

Image credit: Medium

Social Networks

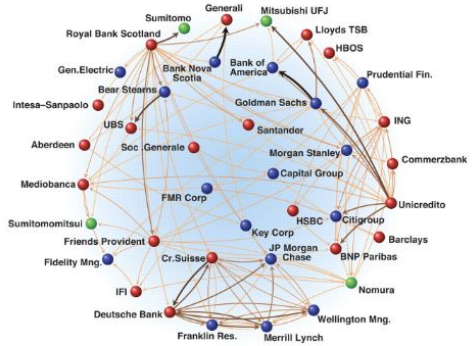
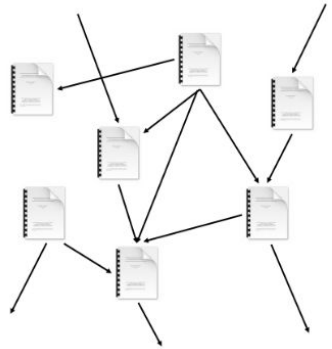


Image credit: Science

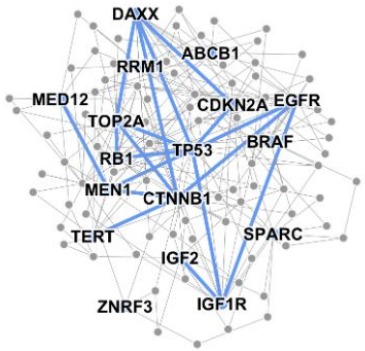
Economic Networks

Image credit: Lumen Learning

Communication Networks



Citation Networks



Disease Pathways

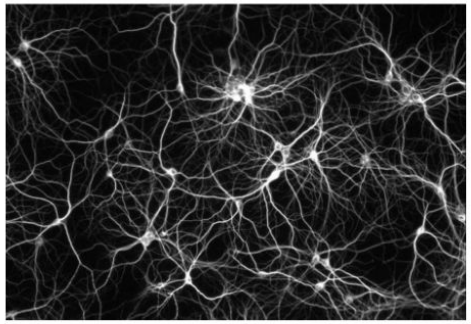
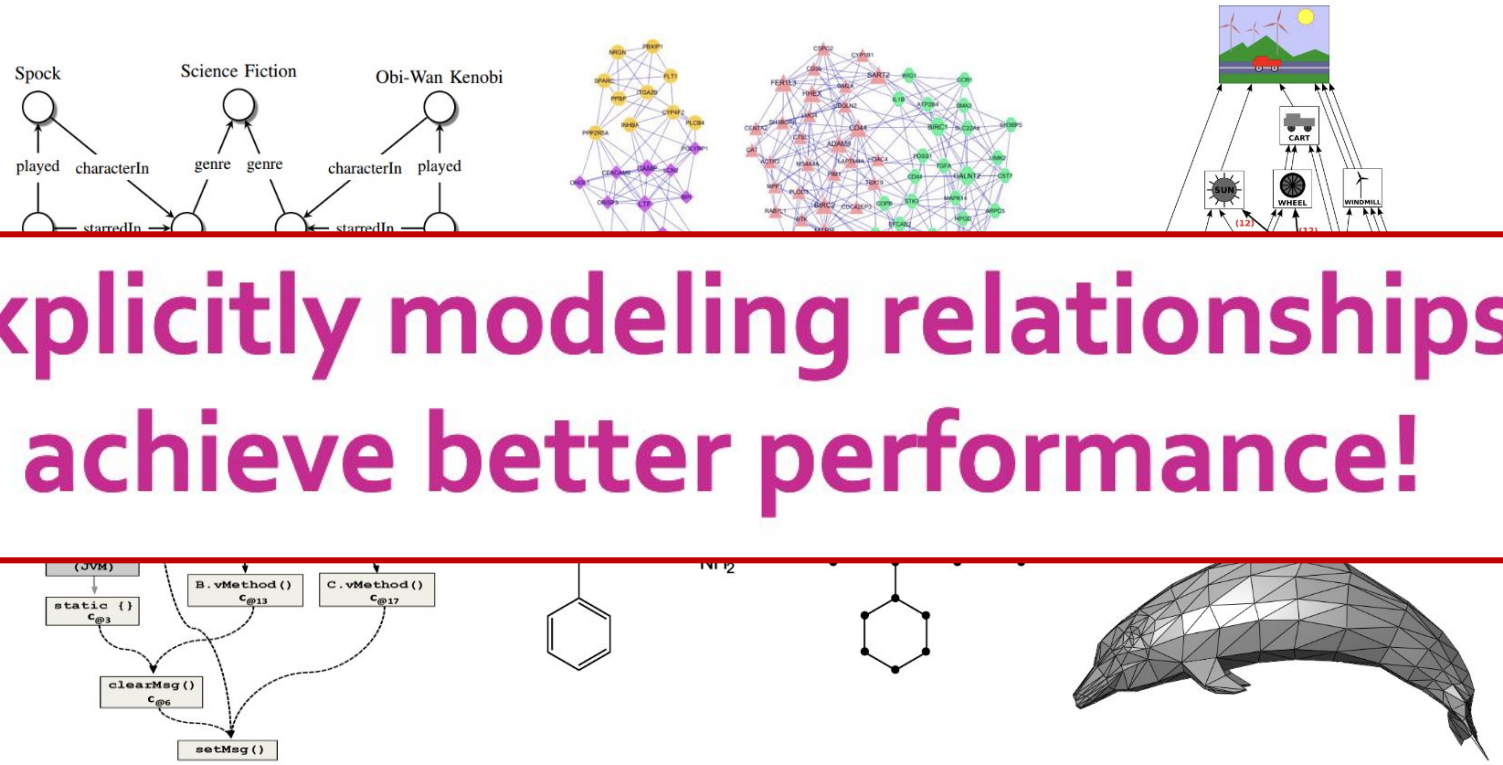


Image credit: The Conversation

Networks of Neurons

Slide courtesy: http://web.stanford.edu/class/cs224w/slides/01-intro.pdf

Majority of real-world data can be described using graphs



By explicitly modeling relationships we achieve better performance!

Image credit: ResearchGate

Code Graphs

Image credit: MDPI

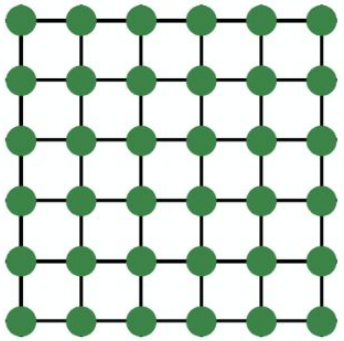
Molecules

Image credit: Wikipedia

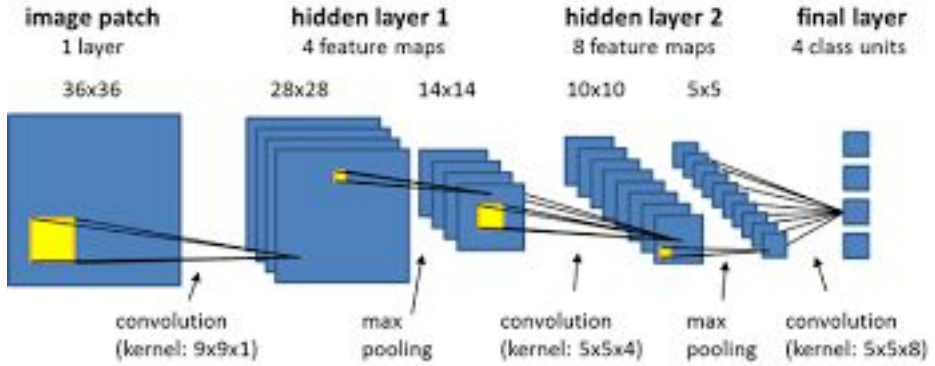
3D Shapes

Slide courtesy: http://web.stanford.edu/class/cs224w/slides/01-intro.pdf

Existing deep learning models designed for grids and sequences



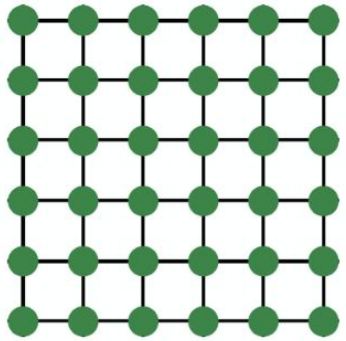
Images



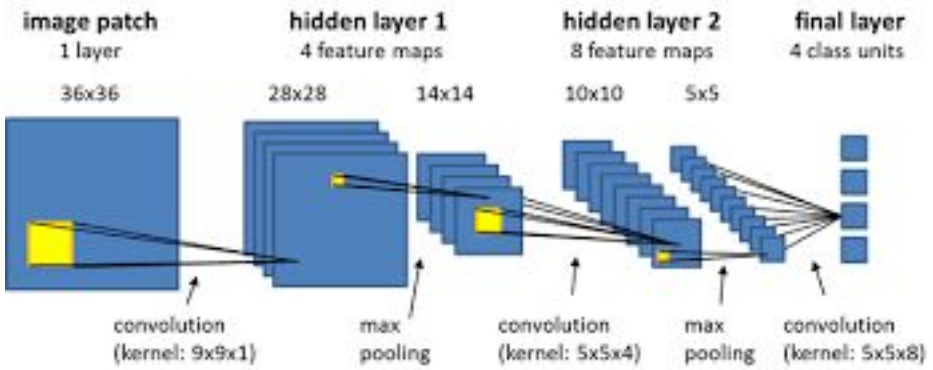
Convolutional Neural Network (CNN)

Slide courtesy: http://web.stanford.edu/class/cs224w/slides/01-intro.pdf
Image courtesy: https://docs.ecognition.com/v10.1.0/eCognition_documentation/User%20Guide%20Developer/8%20Classification%20-%20Deep%20Learning.htm

Existing deep learning models designed for grids and sequences

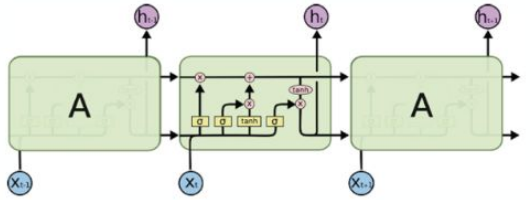


Images



Convolutional Neural Network (CNN)

Text/Speech



Recurrent Neural Network (RNN)

Slide courtesy: http://web.stanford.edu/class/cs224w/slides/01-intro.pdf
Image courtesy: https://docs.ecognition.com/v10.1.0/eCognition_documentation/User%20Guide%20Developer/8%20Classification%20-%20Deep%20Learning.htm

What's hard about NNs for graphs?

- Let's think about the problem of ***summarizing*** a data item
- We've done this for different types of data already:
  - Grids: Pass information from the neighboring pixels
  - Sequences: Pass information along the sequence (or, use a Transformer with positional embeddings)
- In both cases: there is an obvious way to ***direct*** the computation toward a single end point
  - Grid: center of the grid
  - Sequences: the end of the sequence

What's hard about NNs for graphs?

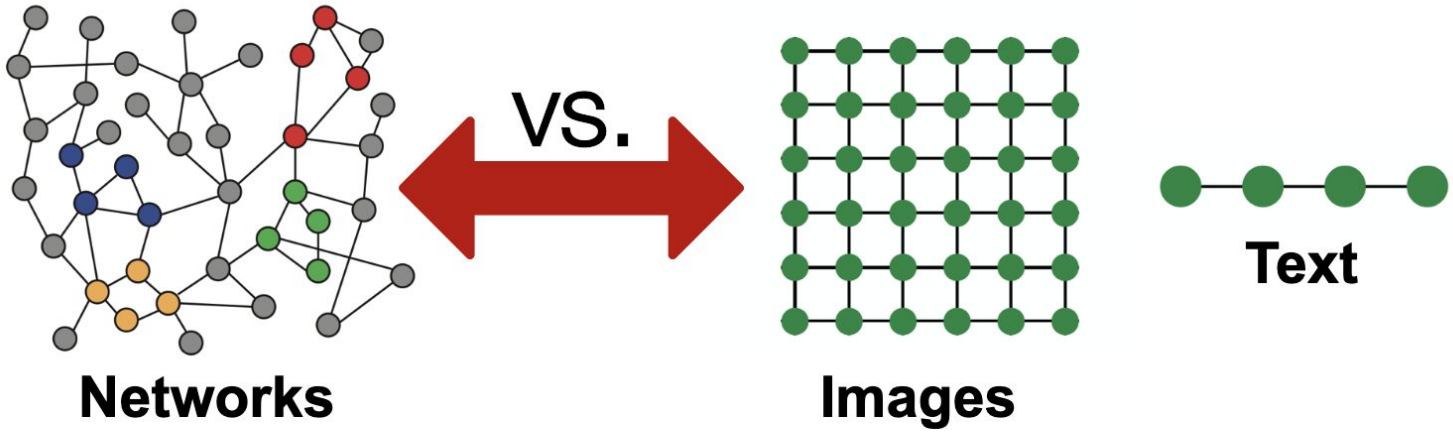
Networks are complex.

Slide courtesy: http://web.stanford.edu/class/cs224w/slides/01-intro.pdf

What's hard about NNs for graphs?

Networks are complex.

- Arbitrary size and complex topological structure (*i.e.*, no spatial locality like grids)

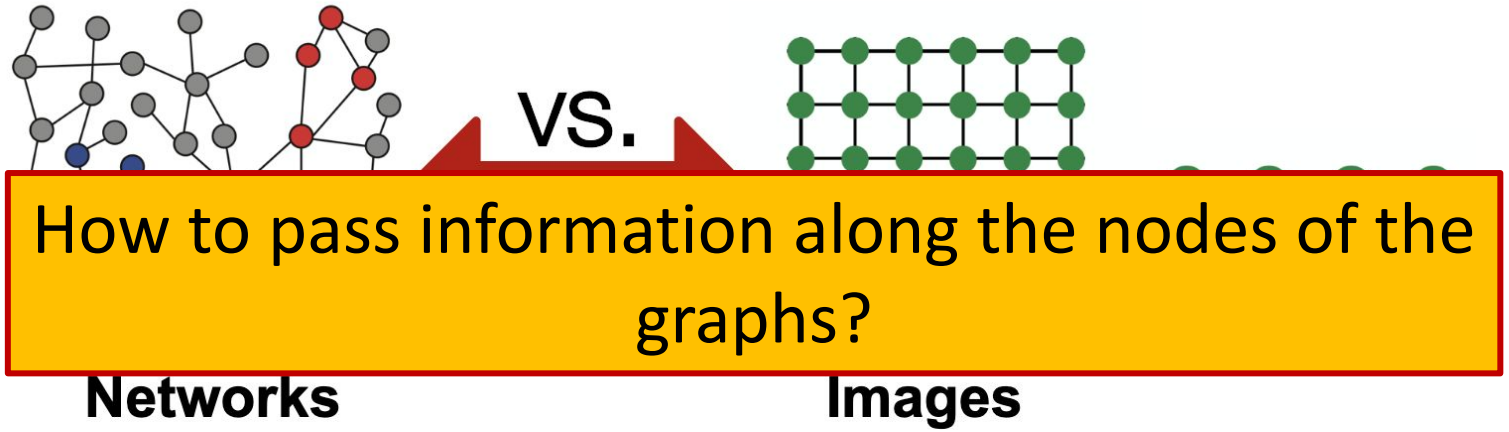


Slide courtesy: http://web.stanford.edu/class/cs224w/slides/01-intro.pdf

What's hard about NNs for graphs?

Networks are complex.

- Arbitrary size and complex topological structure (*i.e.*, no spatial locality like grids)



- No fixed node ordering or reference point

Slide courtesy: http://web.stanford.edu/class/cs224w/slides/01-intro.pdf

An alternate take on summarization

- If we look at our architectures for summarizing sequences and grids, they are actually doing two important things at once:
  1. Passing information along relationships between individual elements
  2. Aggregating per-element information into a summarized description

- Let's design a graph network that does both of these things

Today's goal – learn about graph-based neural networks (GNNs)

(1) Motivation

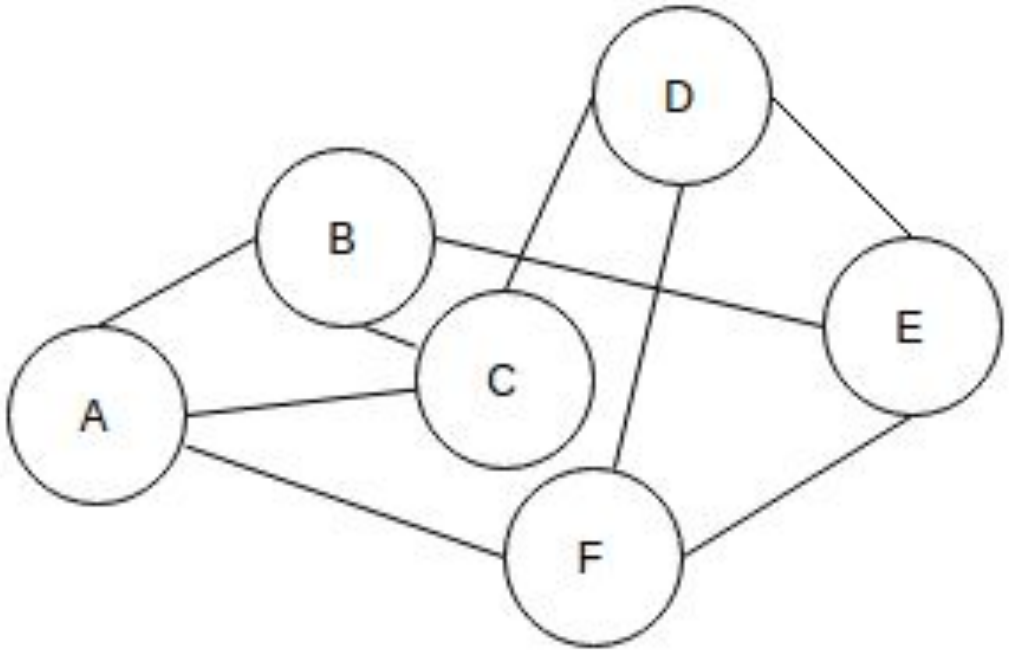
(2) **Passing information in Graphs (for NNs)**

(3) How to use GNNs?

(4) Graph Attention Networks

Why is passing information necessary?

- I.e. why not just compute some **descriptor on the nodes**, then aggregate them?



Node features: numerical values that describe the node

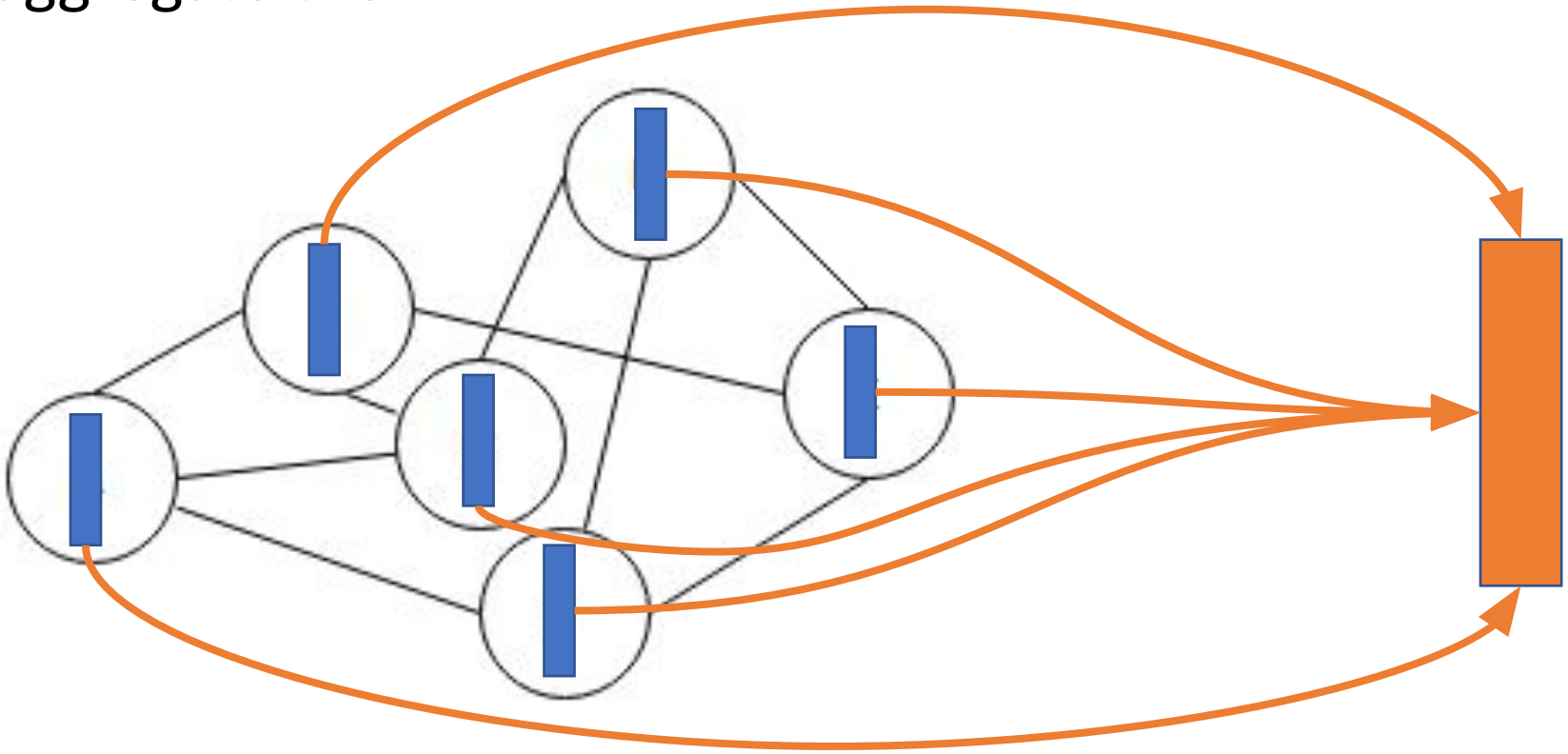
Image credit: Medium

Social Networks

Node features: [**24** (age), **1** (if employed),…]

Why is passing information necessary?

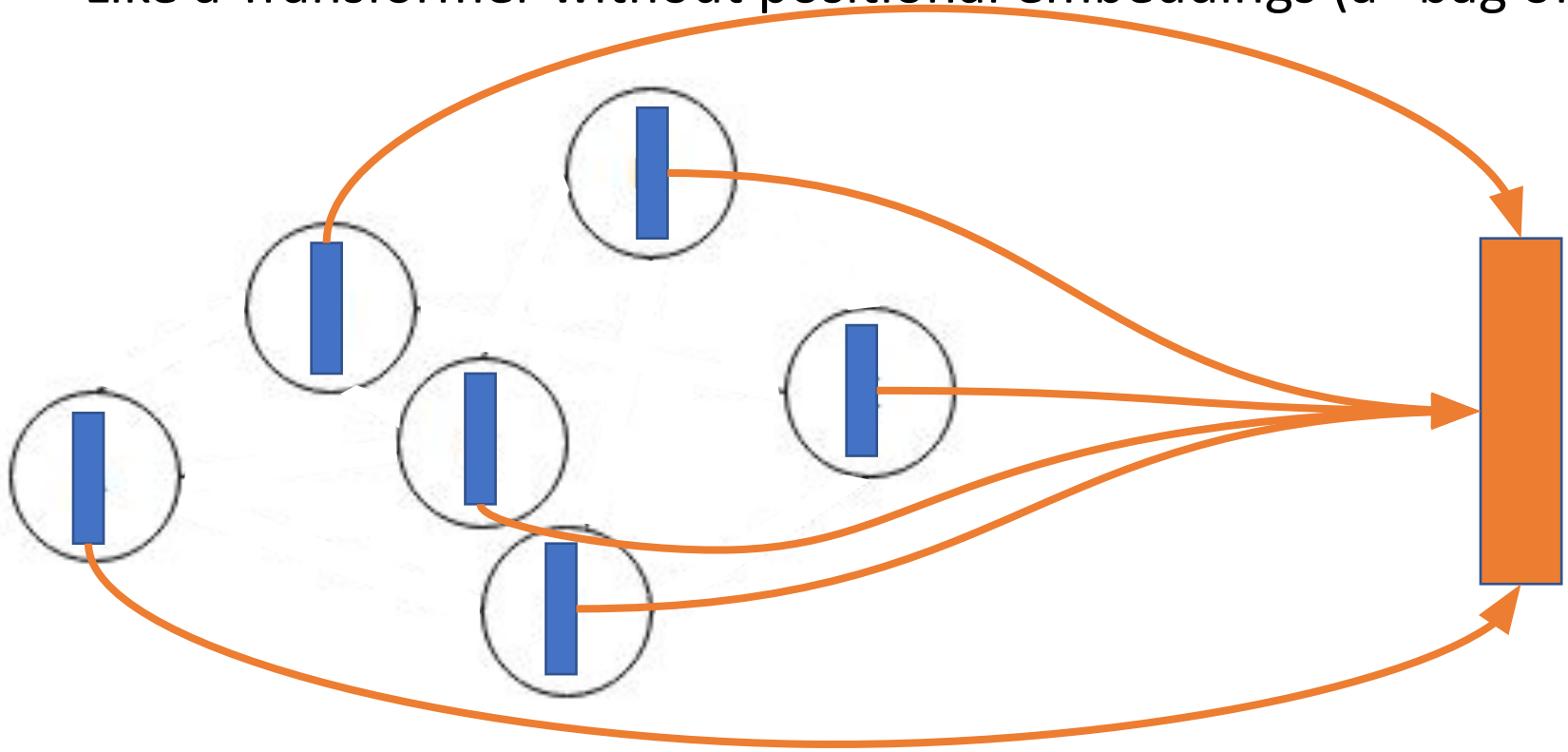
- I.e. why not just compute some descriptor on the nodes, then aggregate them?



What is the issue here?

Why is passing information necessary?

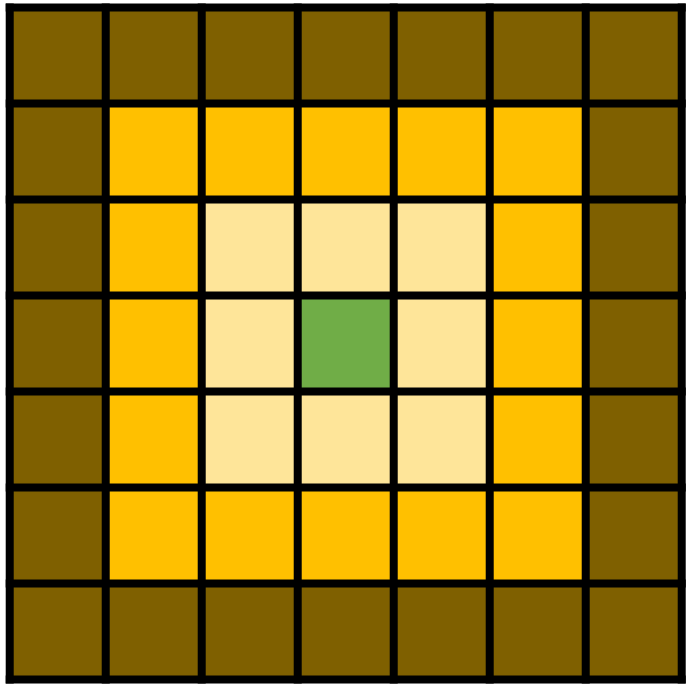
- This totally ignores the structure of the data!
  - Like a Transformer without positional embeddings (a “bag of nodes”)



So, how to pass information along edges?

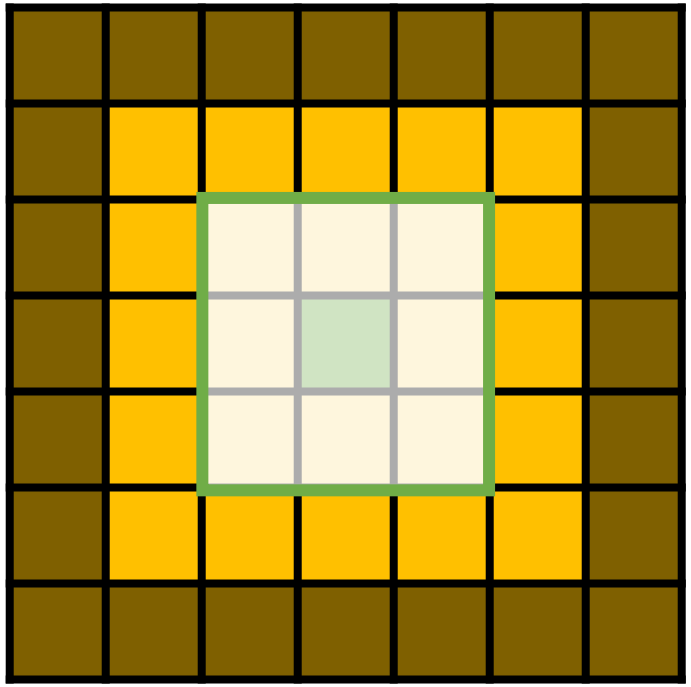
- We’re going to generalize an operation we’ve already seen, so that it operates on graphs...

For NNs on Graphs, we generalize ***convolution***



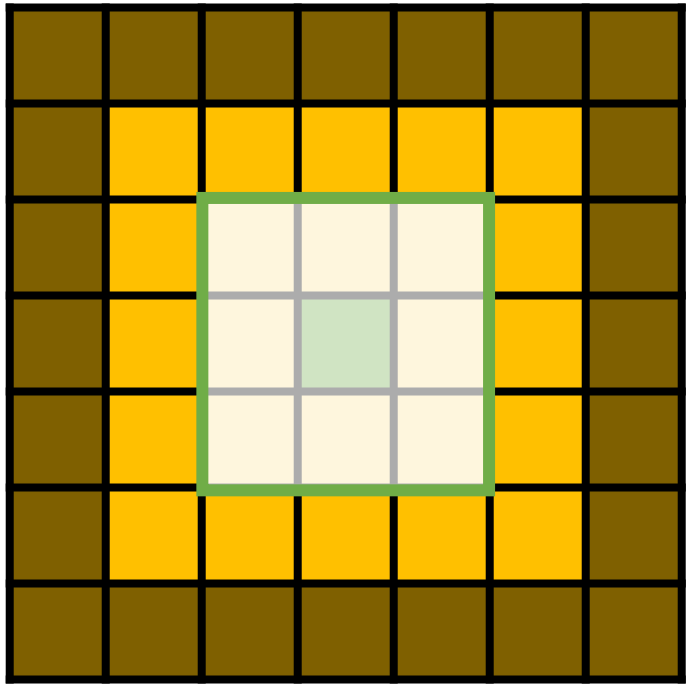
One pixel in an Image

For NNs on Graphs, we generalize ***convolution***

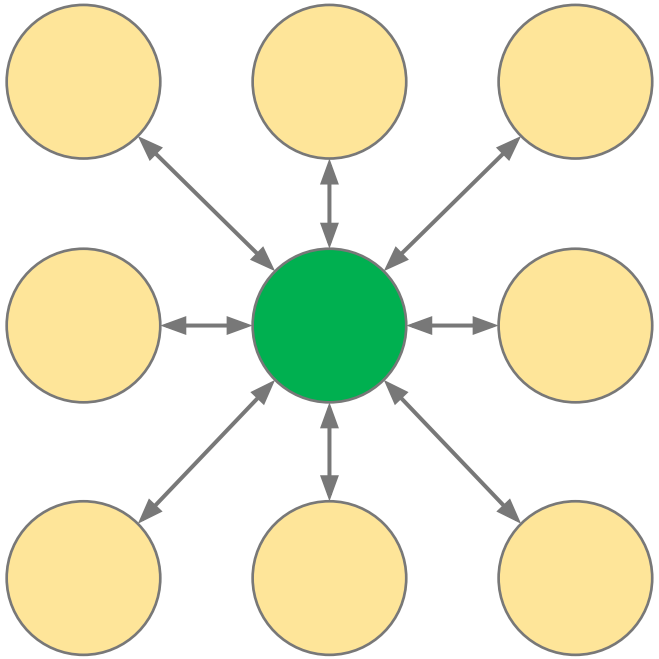


Convolution aggregates information from
the immediate neighborhood of a pixel

For NNs on Graphs, we generalize ***convolution***



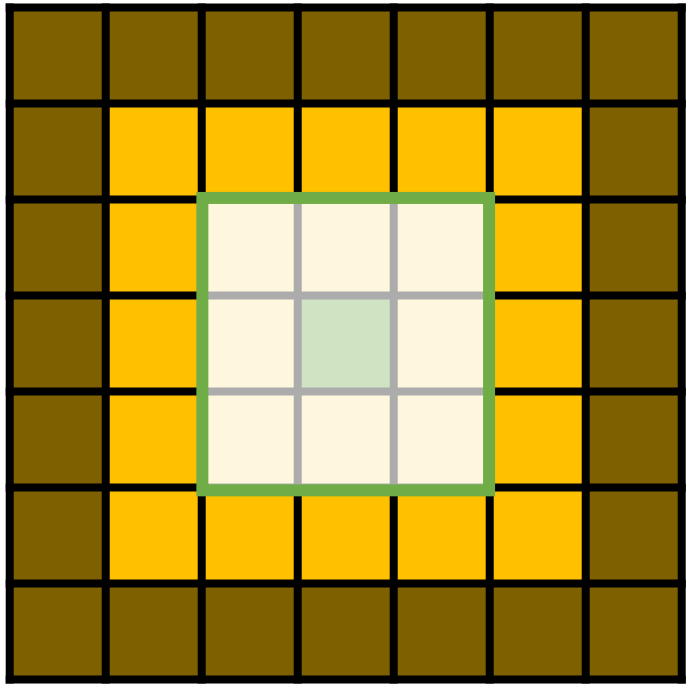
Convolution aggregates information from the immediate neighborhood of a pixel



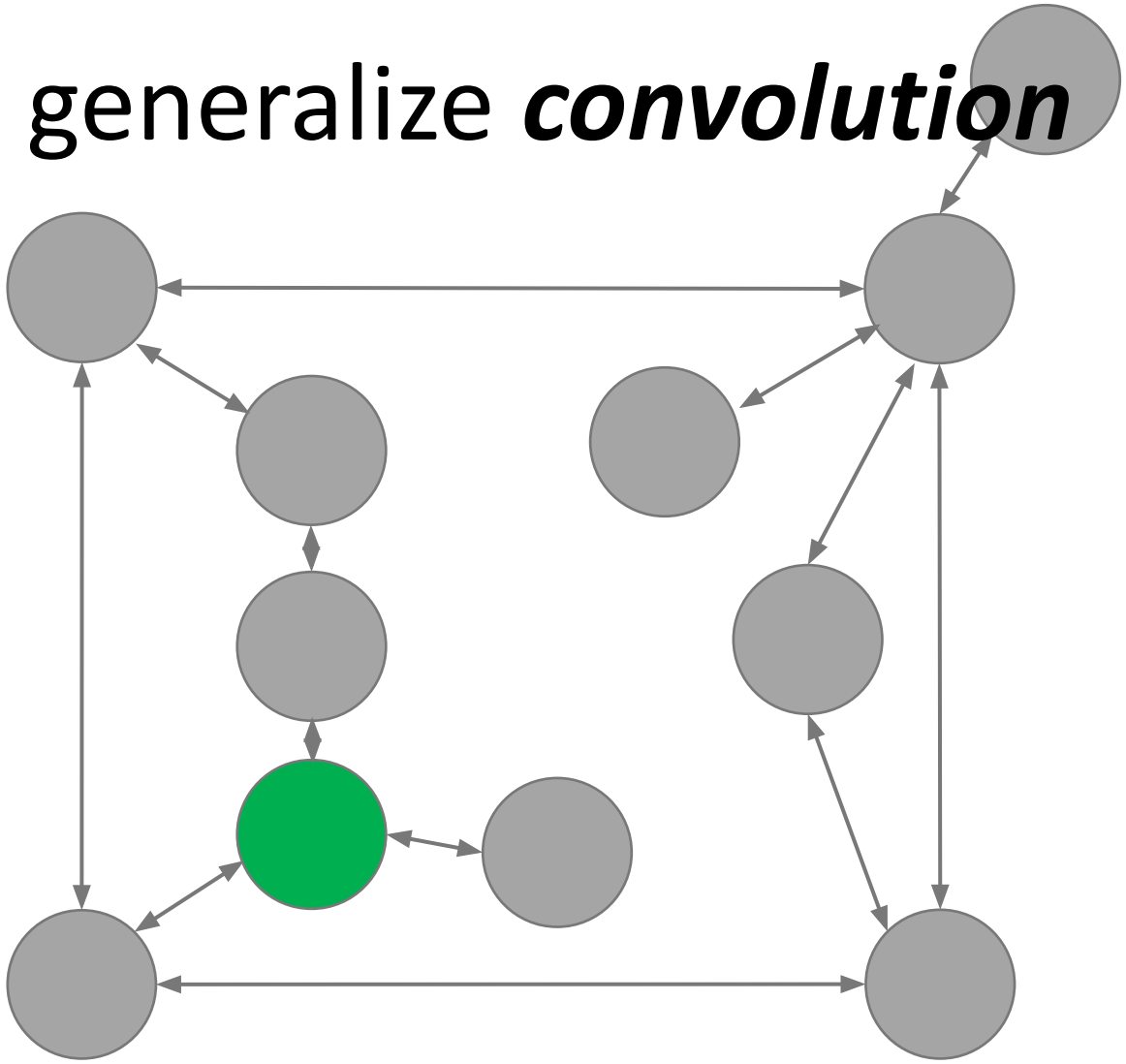
Another interpretation of this aggregation

Looks like a graph, no?

For NNs on Graphs, we generalize ***convolution***



Convolution aggregates information from the immediate neighborhood of a pixel



A more general graph

How do we define convolution on ***this***?

What other information do we need here?

For NNs on Graphs, we generalize ***convolution***

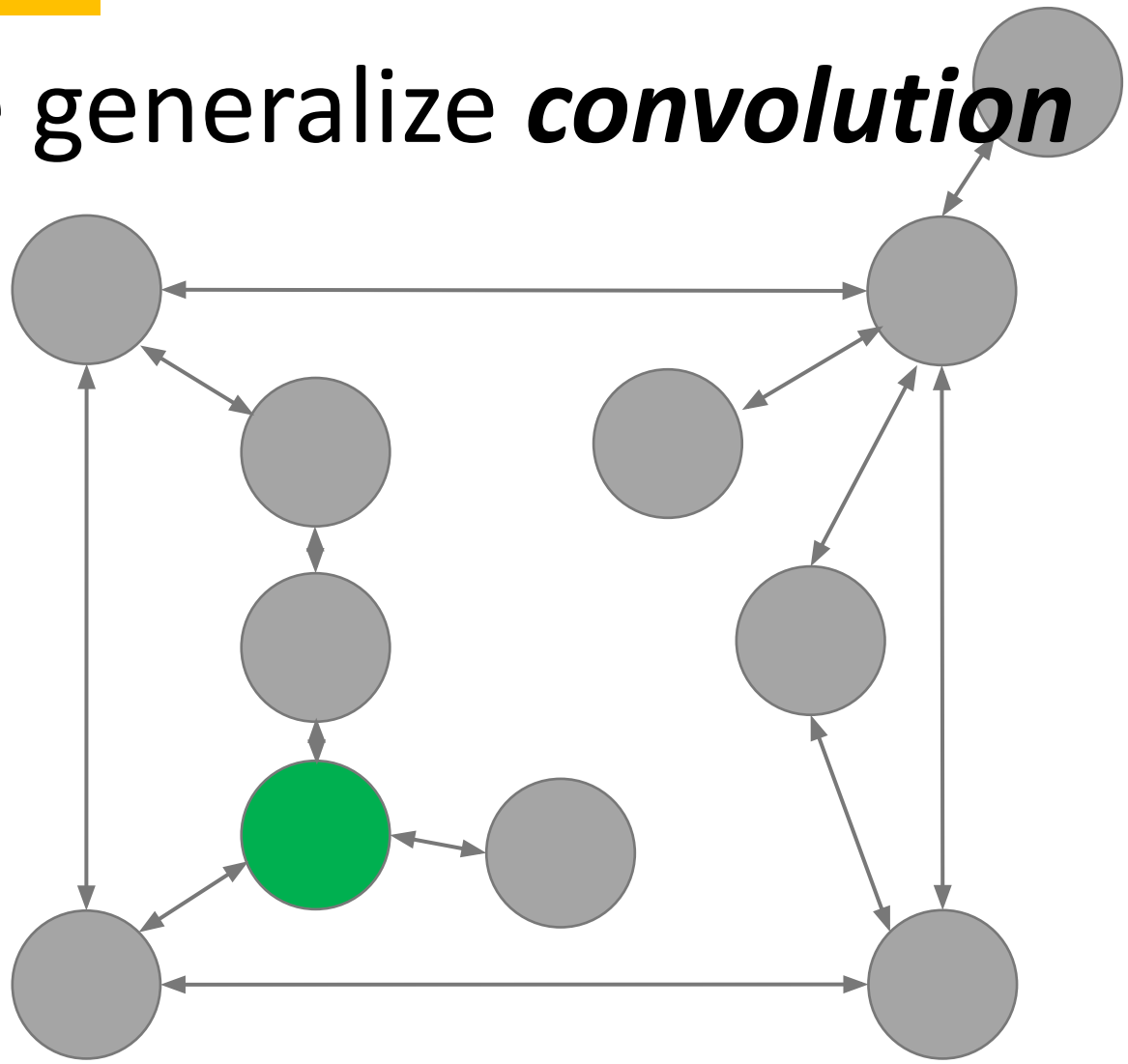
Inputs:
Node features +
Adjacency matrix

Image credit: Medium

Social Networks

Node features: [**24** (age), **1** (if employed),…]

Adjacency matrix: $\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$



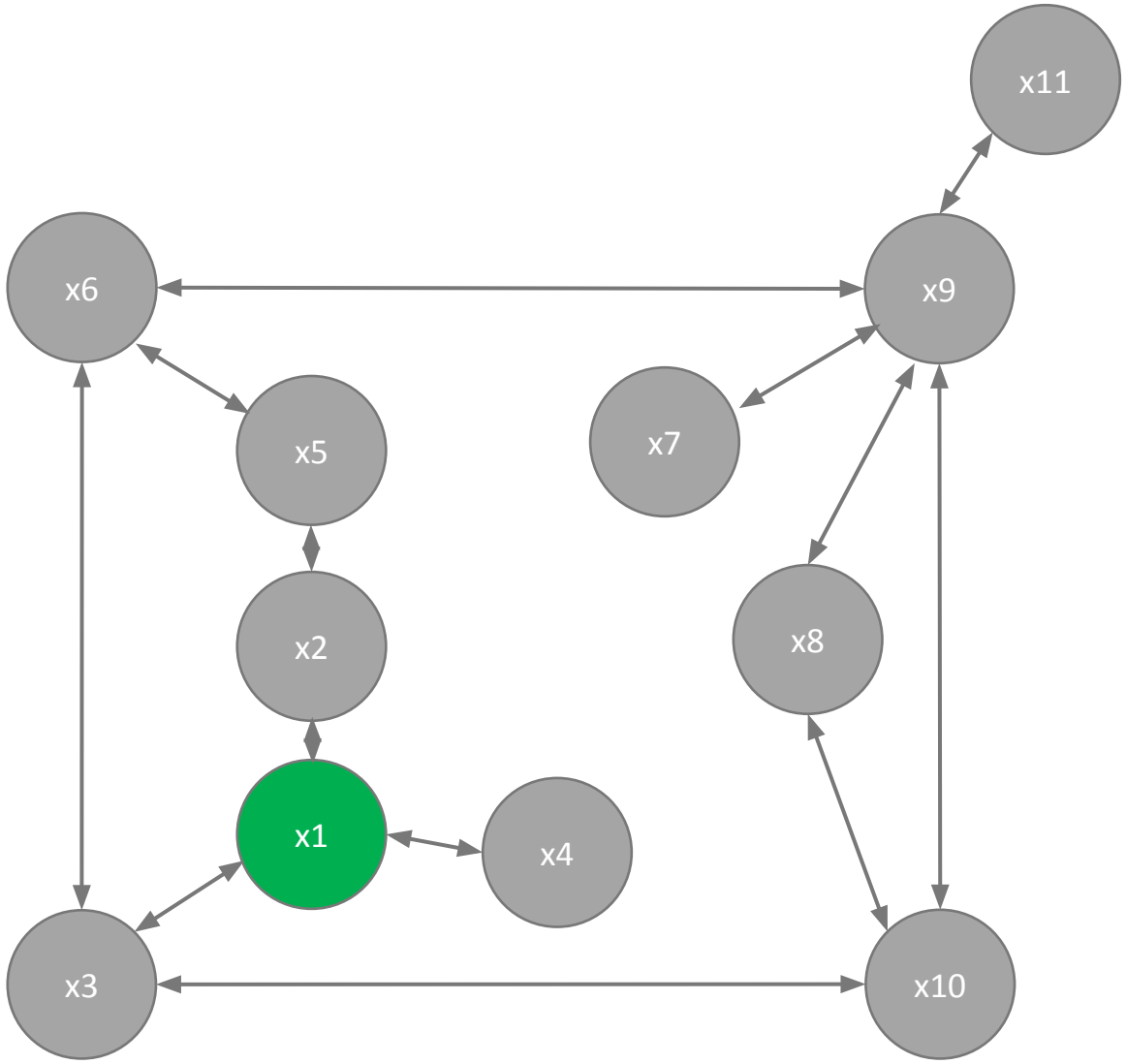
A more general graph

How do we define convolution on ***this***?

Message Passing

We start by mapping the features **x** at each node to a hidden state vector **h**

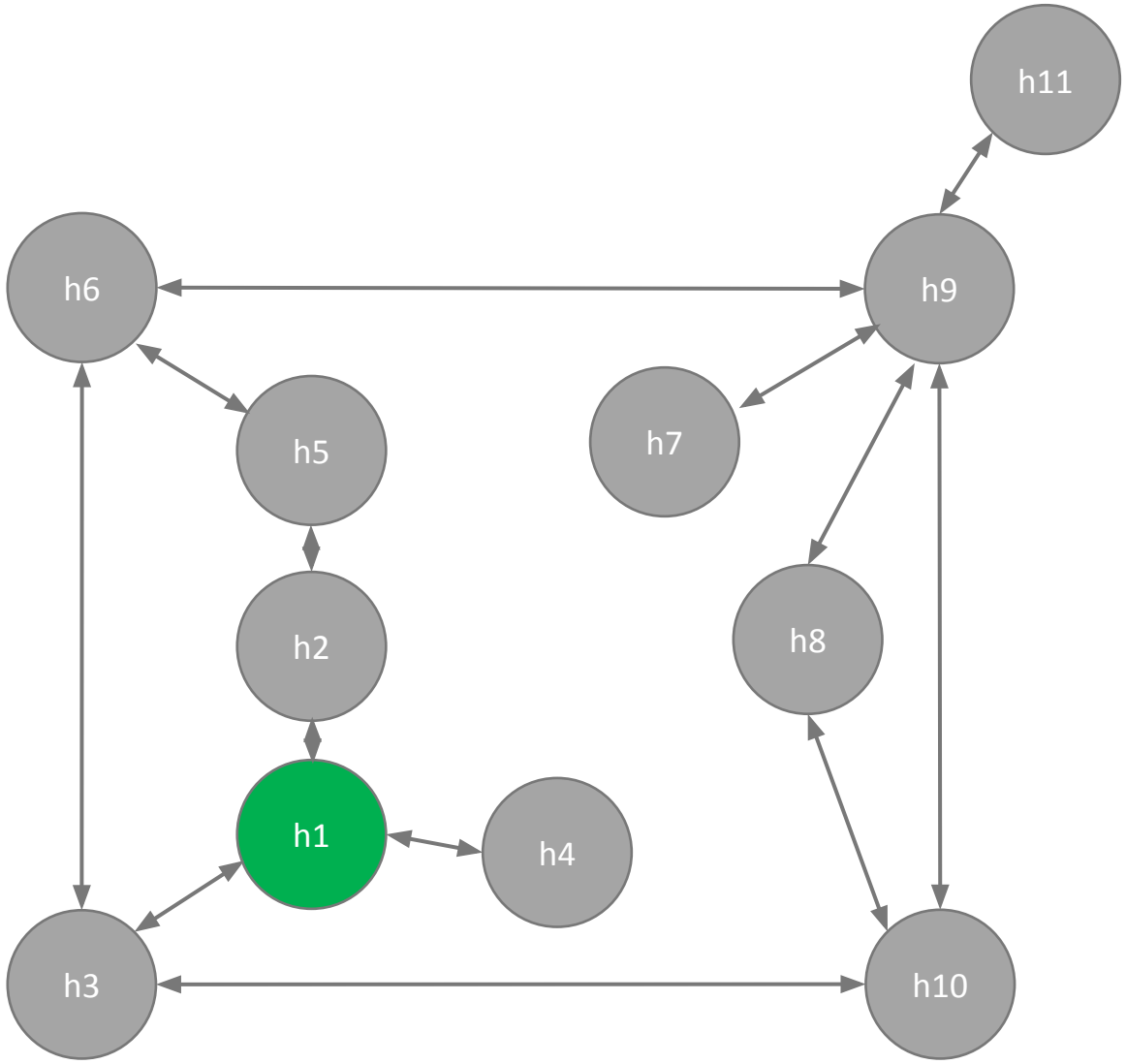
$$f_{\text{init}}(\mathbf{x}) = \mathbf{h}$$



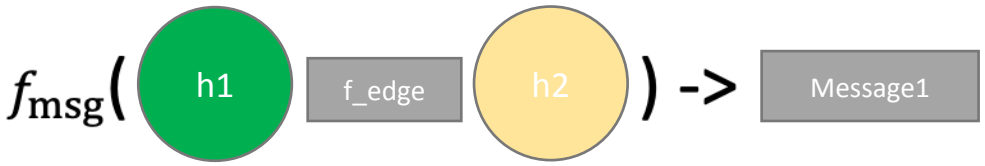
Message Passing

We start by mapping the features **x** at each node to a hidden state vector **h**

$$f_{\text{init}}(\mathbf{x}) = \mathbf{h}$$

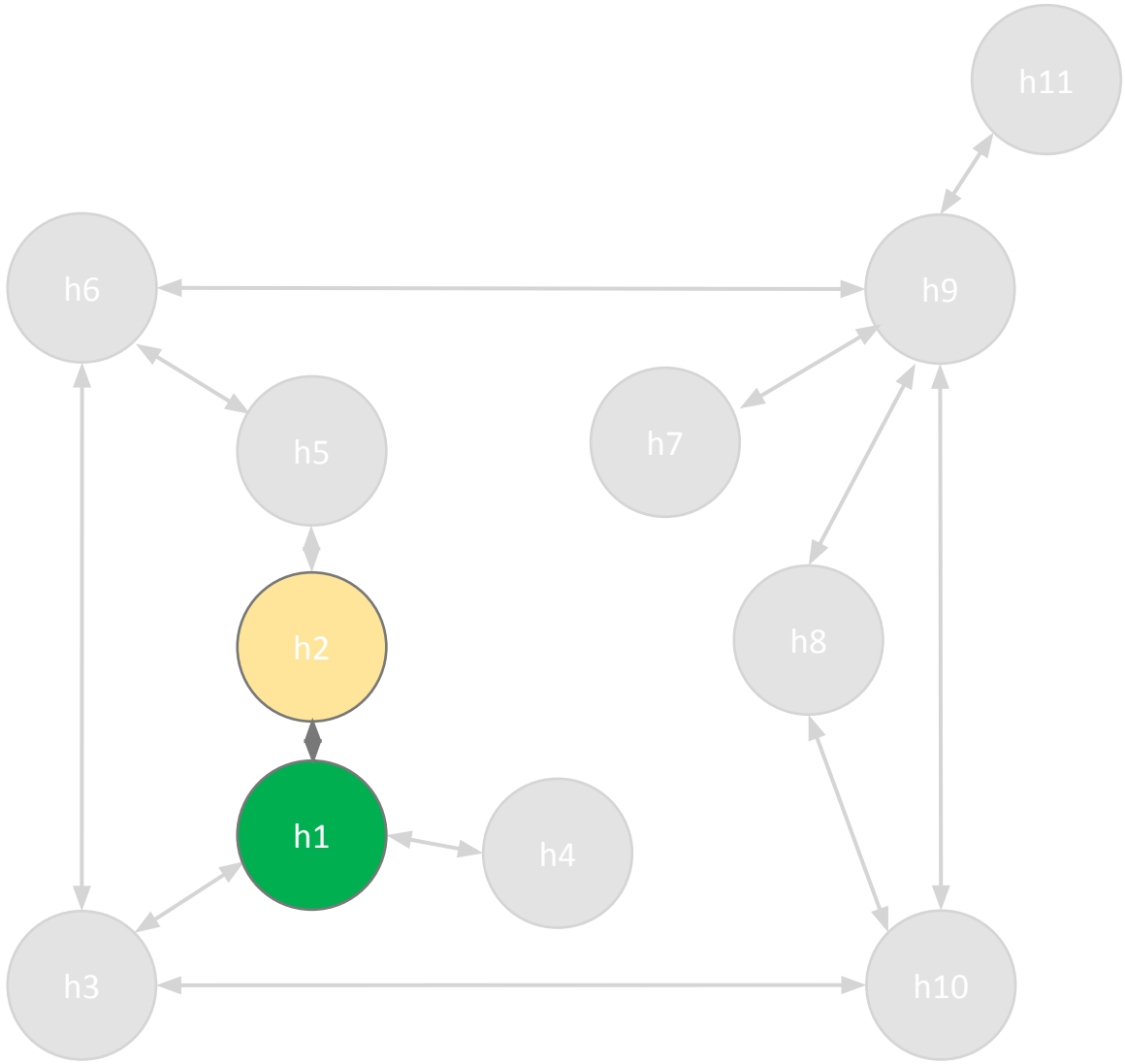


Message Passing

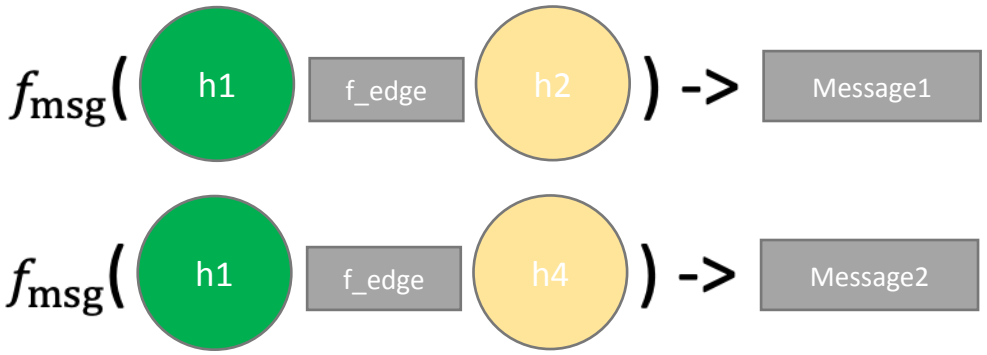


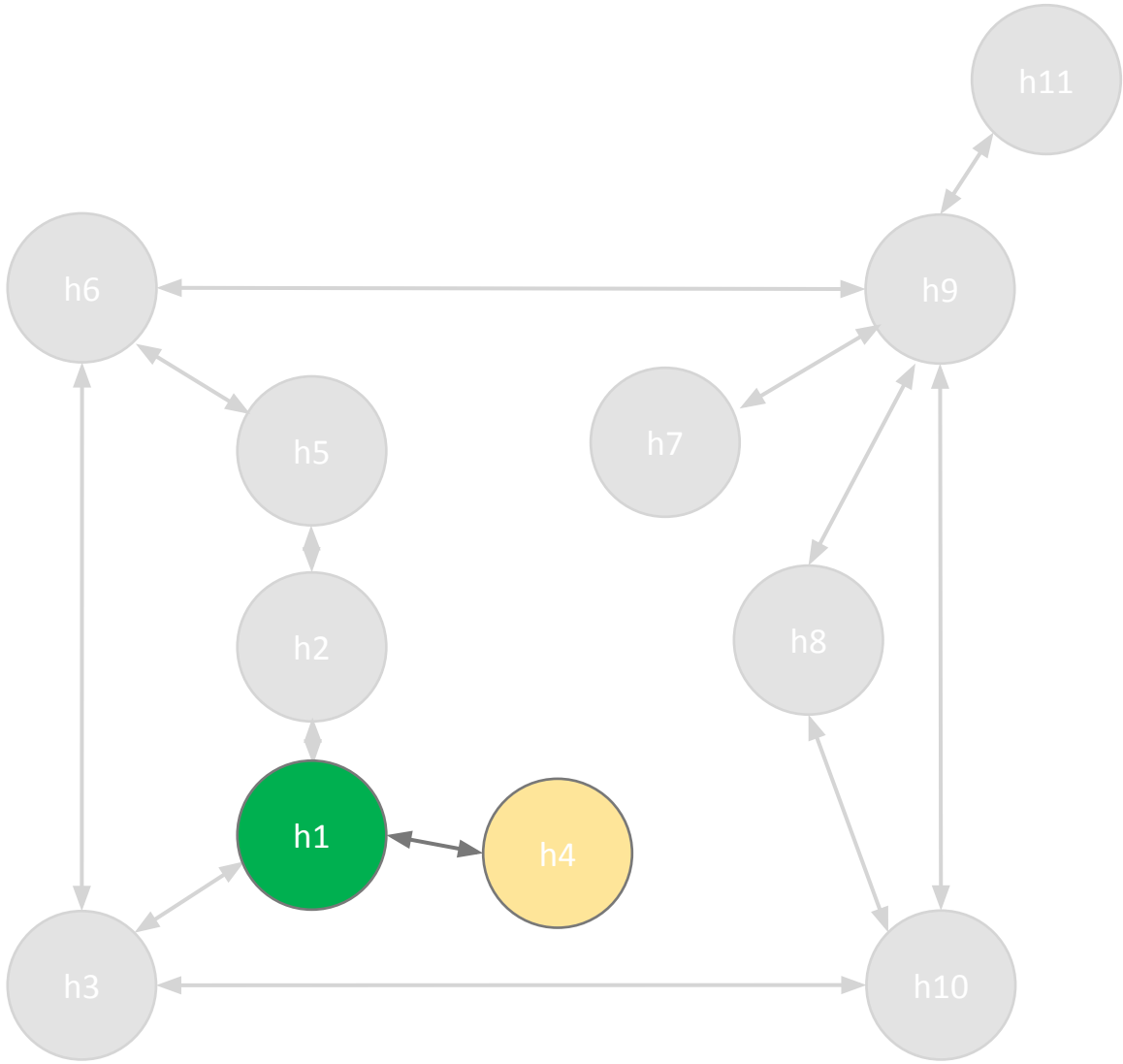
We use a neural network f_{msg} to compute a ***message*** between two nodes in the graph

- Inputs
  - Hidden states for each node
  - Features along the edge
- Output
  - Message vector

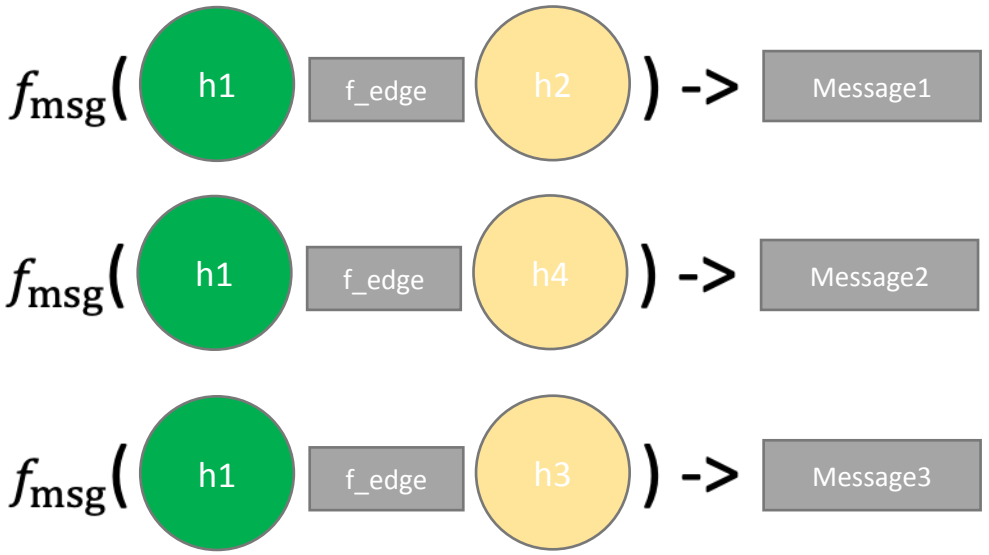


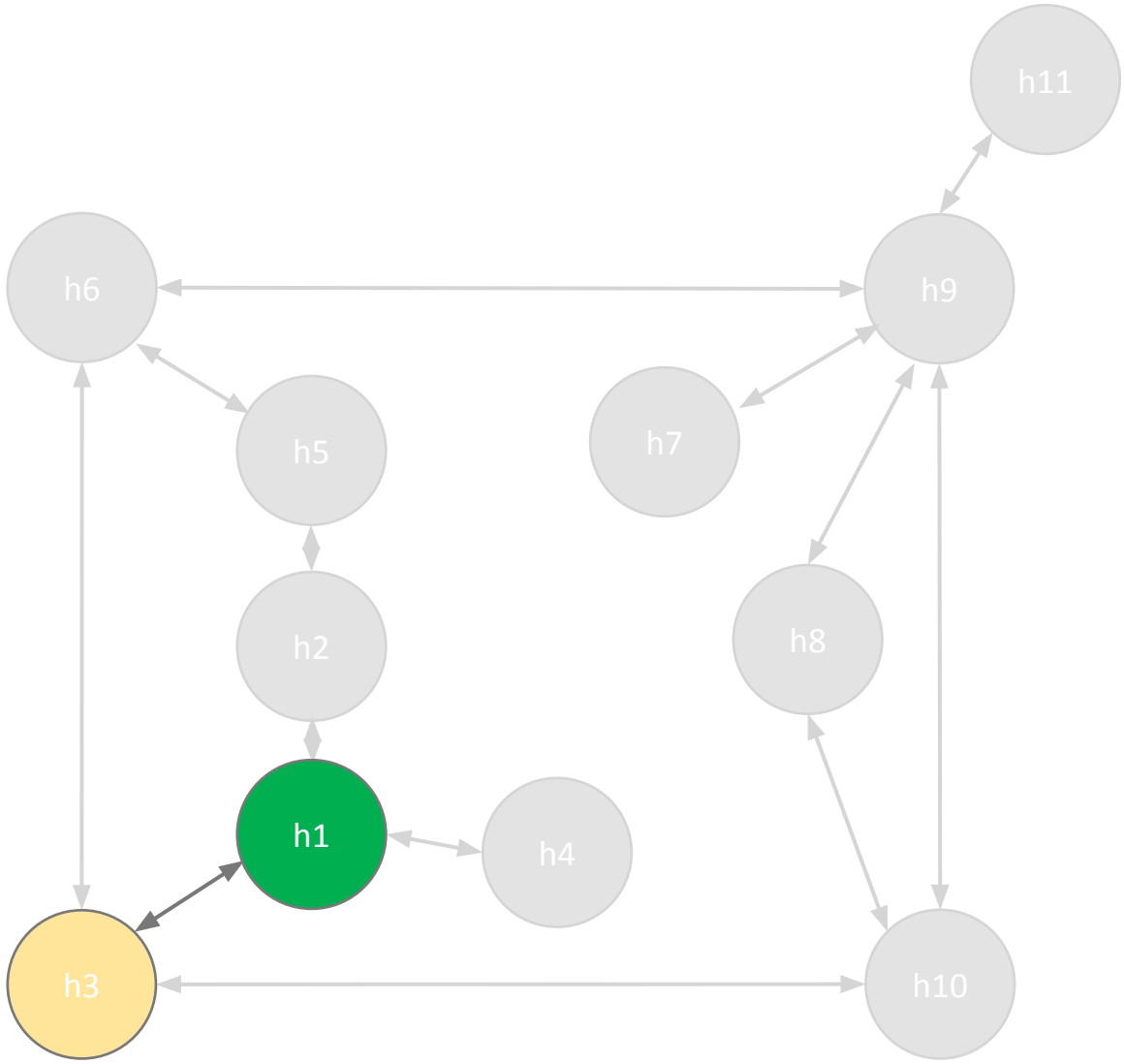
Message Passing



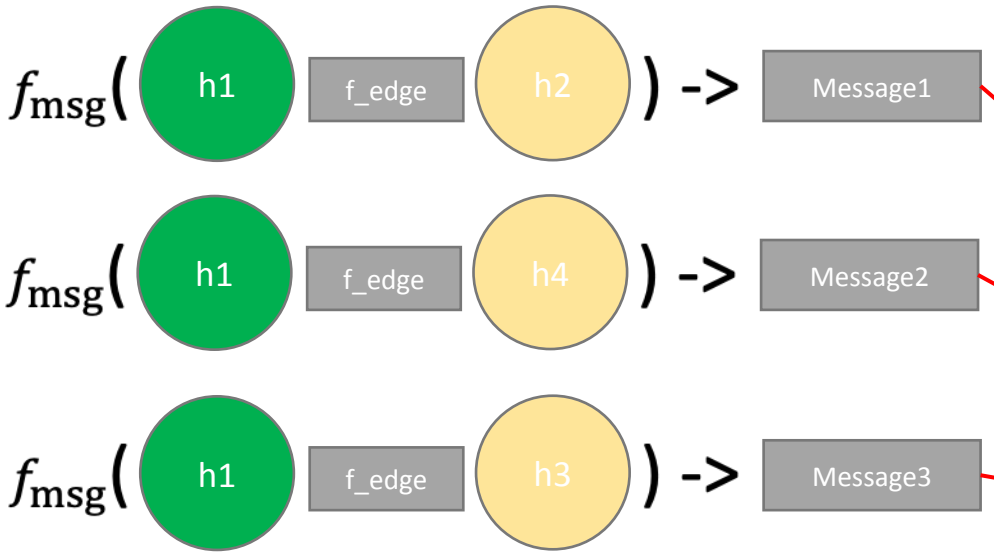


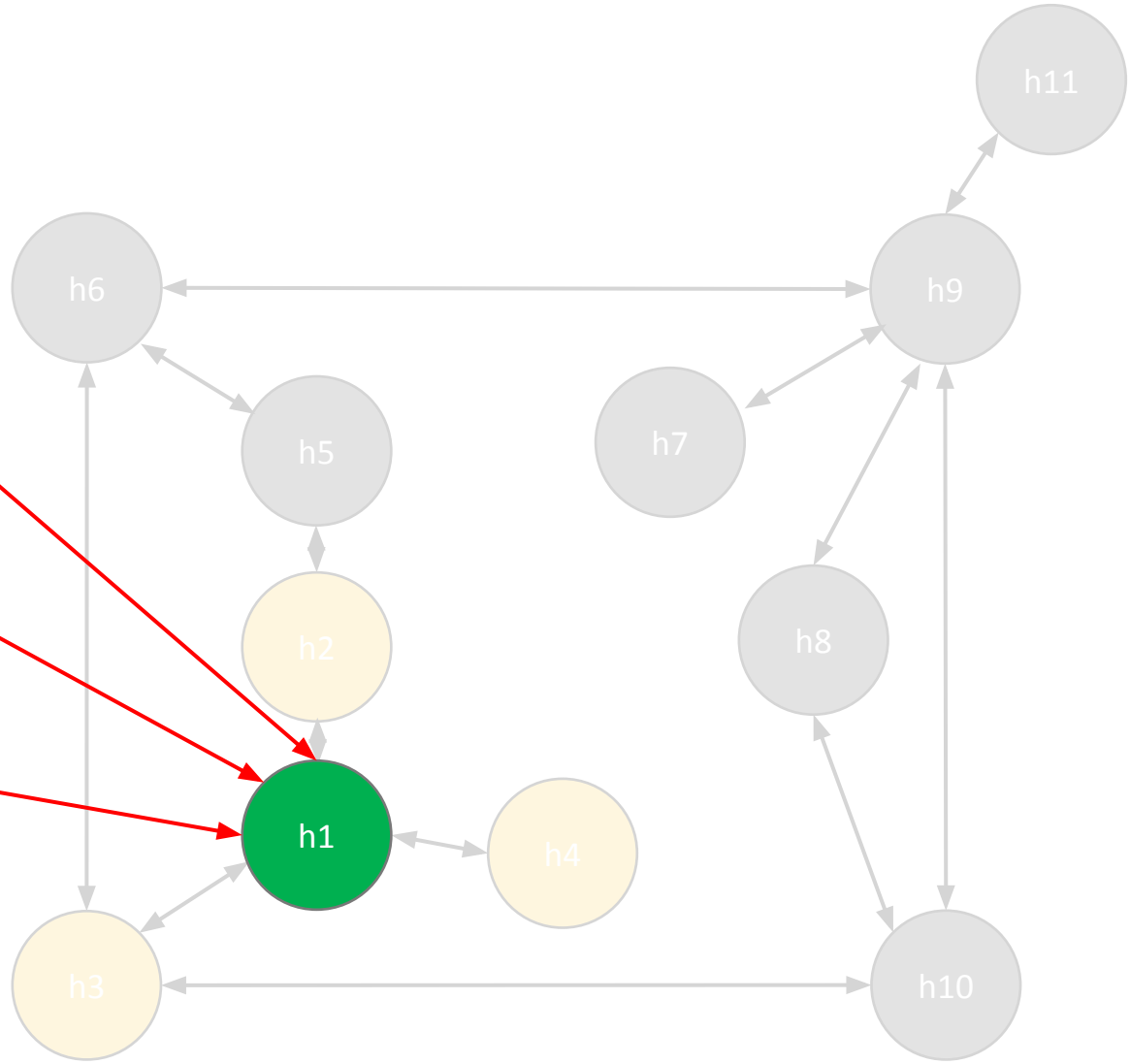
Message Passing





Message Passing



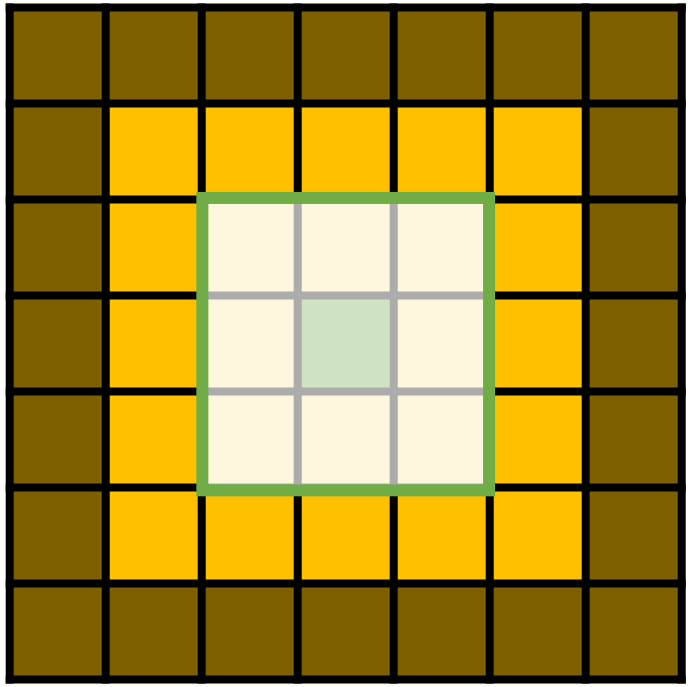


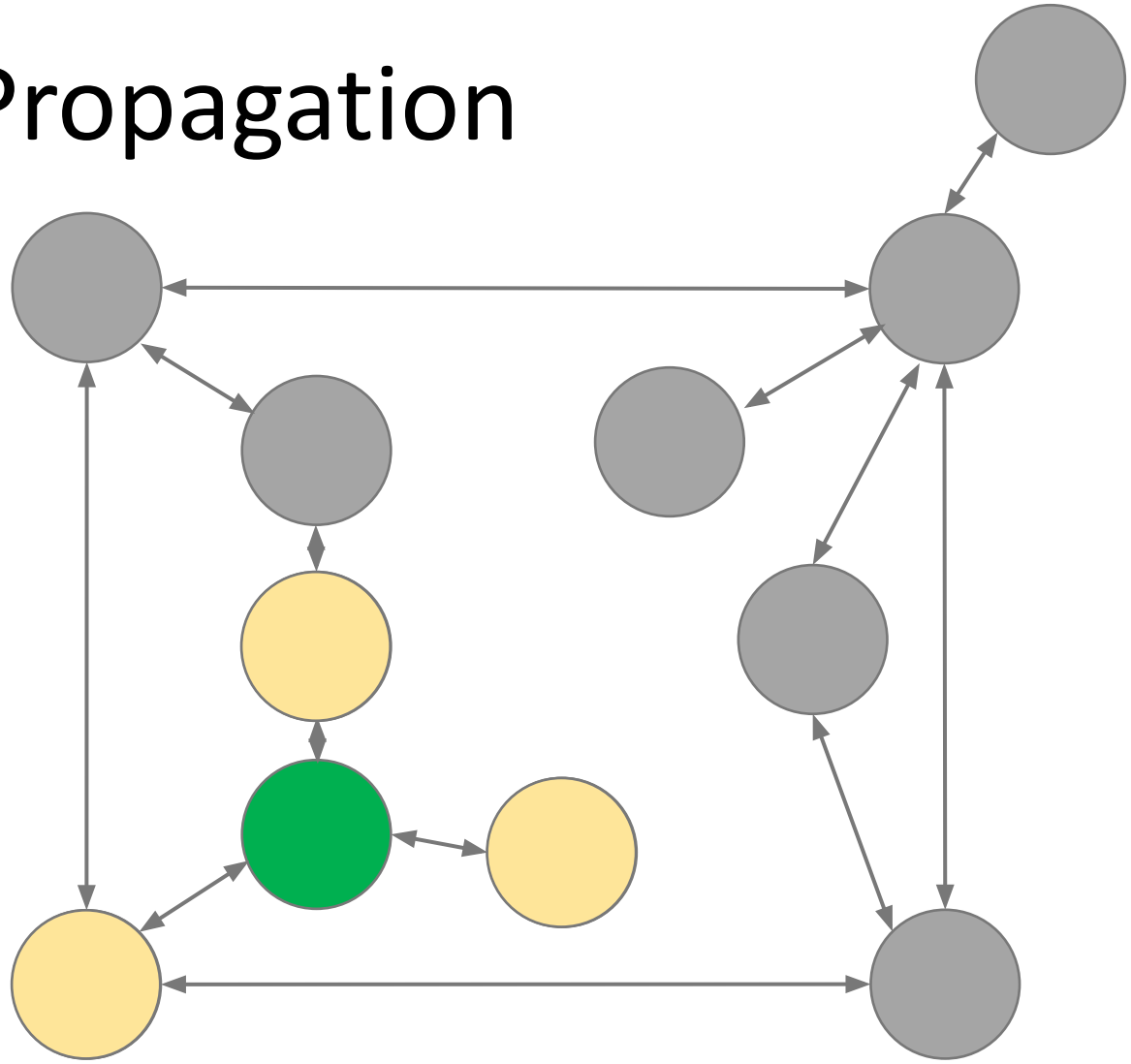
Aggregate all the messages to compute the output hidden state vector for $\mathbf{h}_1$

(e.g. sum up the messages)

NOTE: The aggregation function can differ leading to different formulations of GNNs

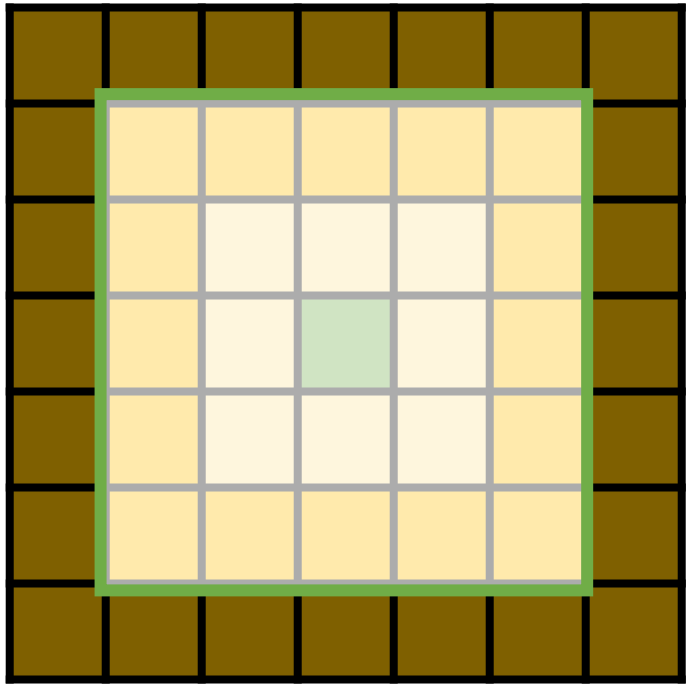
Multiple Rounds of Propagation



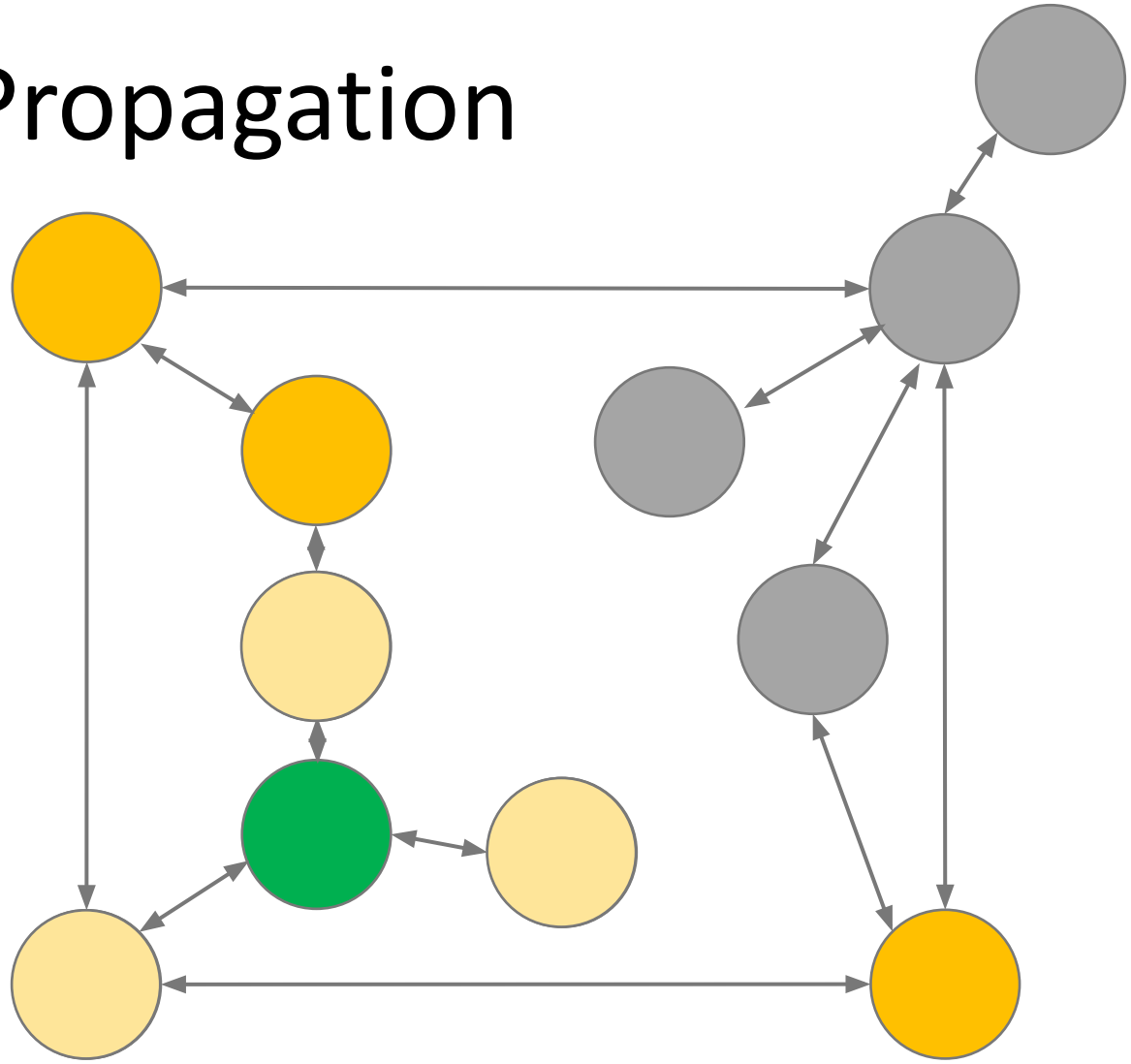


Just as multiple steps of convolution increases the “receptive field” of an image convolutional network...

Multiple Rounds of Propagation



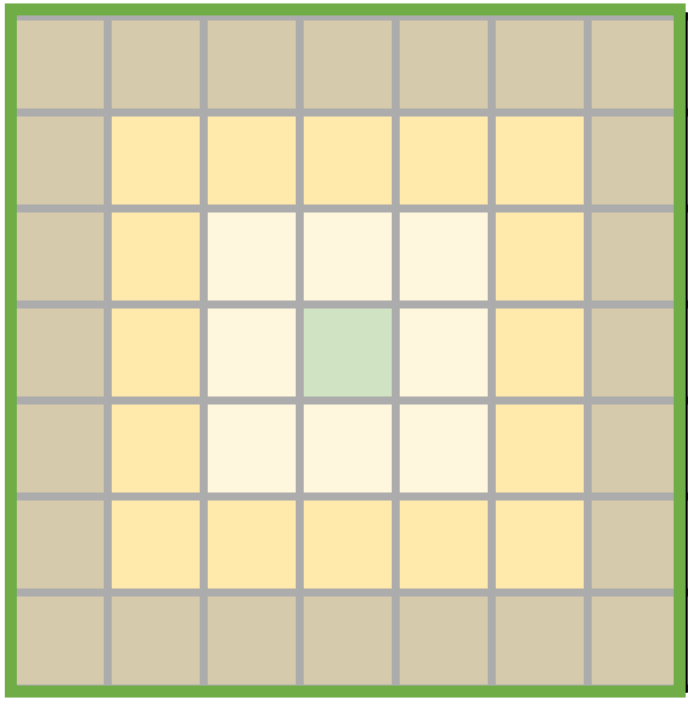
Just as multiple steps of convolution increases the “receptive field” of an image convolutional network...



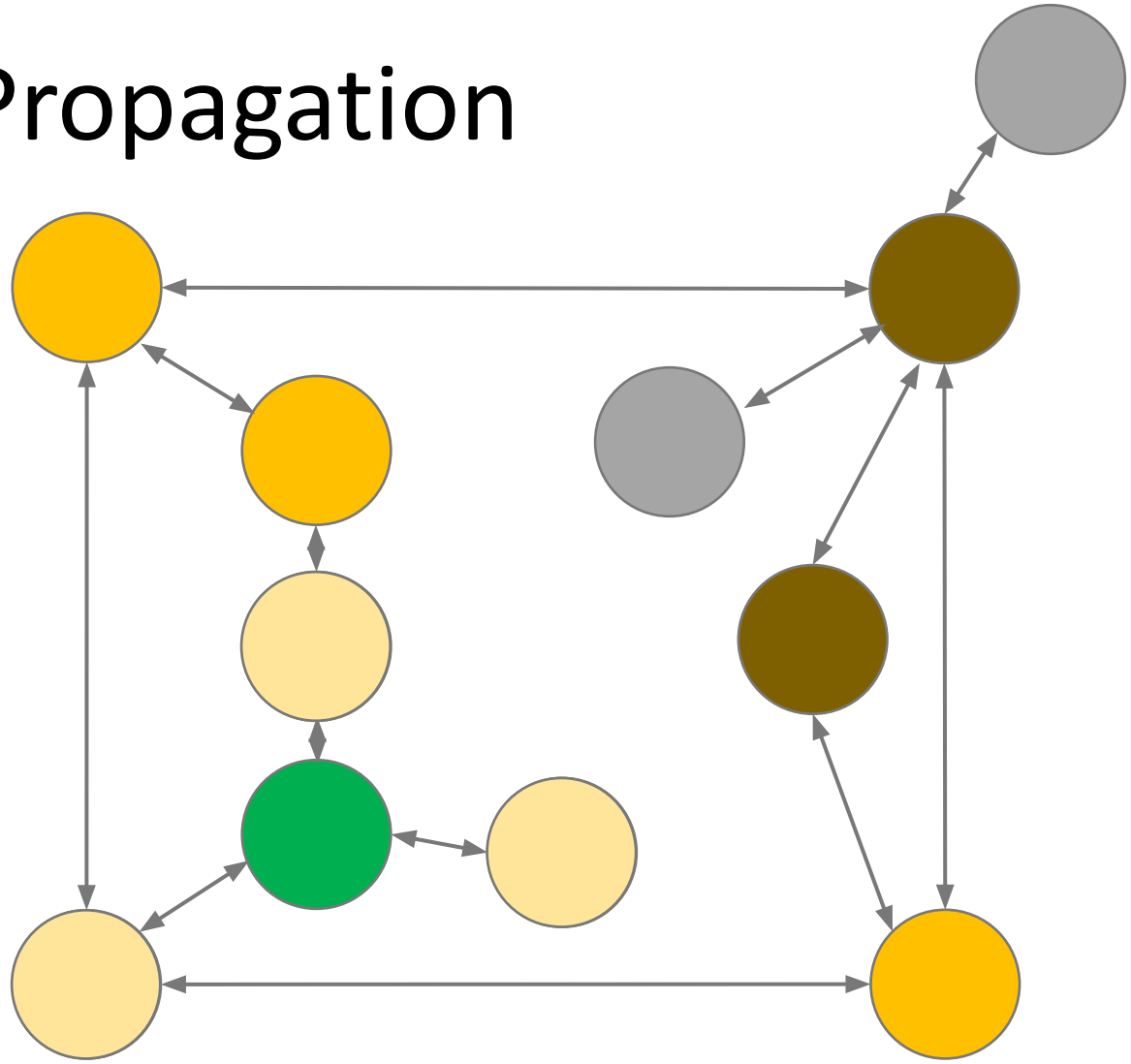
...multiple steps of message passing increases the range of information spread in the graph

Why is that?

Multiple Rounds of Propagation



Just as multiple steps of convolution increases the “receptive field” of an image convolutional network...



...multiple steps of message passing increases the range of information spread in the graph

Example: GraphSAGE formulation

Any questions?

$$\mathbf{h}_v^0 = \mathbf{x}_v$$

$$\mathbf{h}_v^k = \sigma\left(\mathbf{W}_k \sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{|N(v)|} + \mathbf{B}_k \mathbf{h}_v^{k-1}\right), \quad \forall k \in \{1, ..., K\}$$

$$\mathbf{z}_v = \mathbf{h}_v^K$$

Initial 0-th layer embeddings are equal to node features

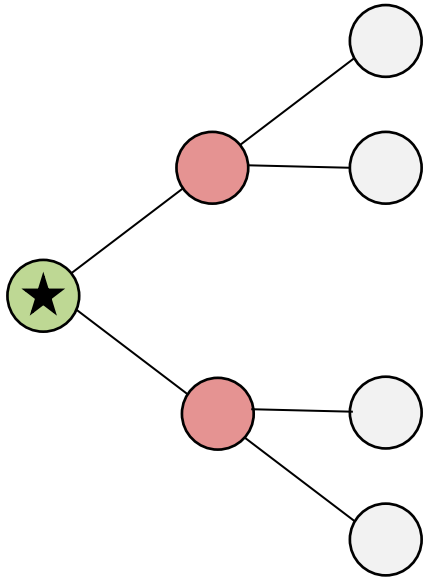
Previous layer embedding of v

Average of neighbor's previous layer embeddings

Non-linearity (e.g., ReLU)

Embedding after K layers of neighborhood aggregation

W and B are learnable parameters



Inductive Representation Learning on Large Graphs

Courtesy: http://snap.stanford.edu/deepnetbio-ismb/slides/deepnetbio-part2-gcn.pdf

Today's goal – learn about graph-based neural networks (GNNs)

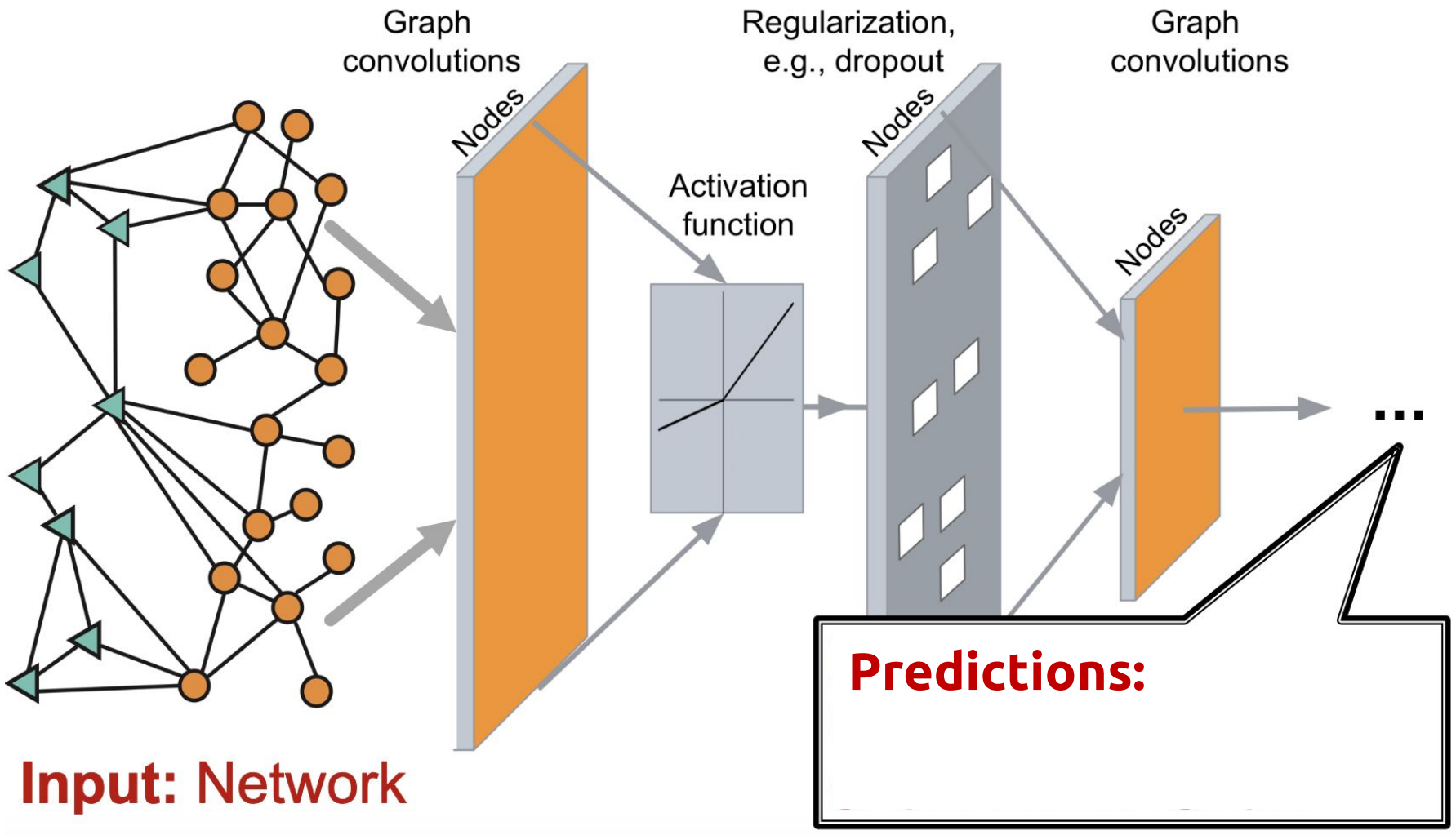
(1) Motivation

(2) Passing information in Graphs (for NNs)

(3) How to use GNNs?

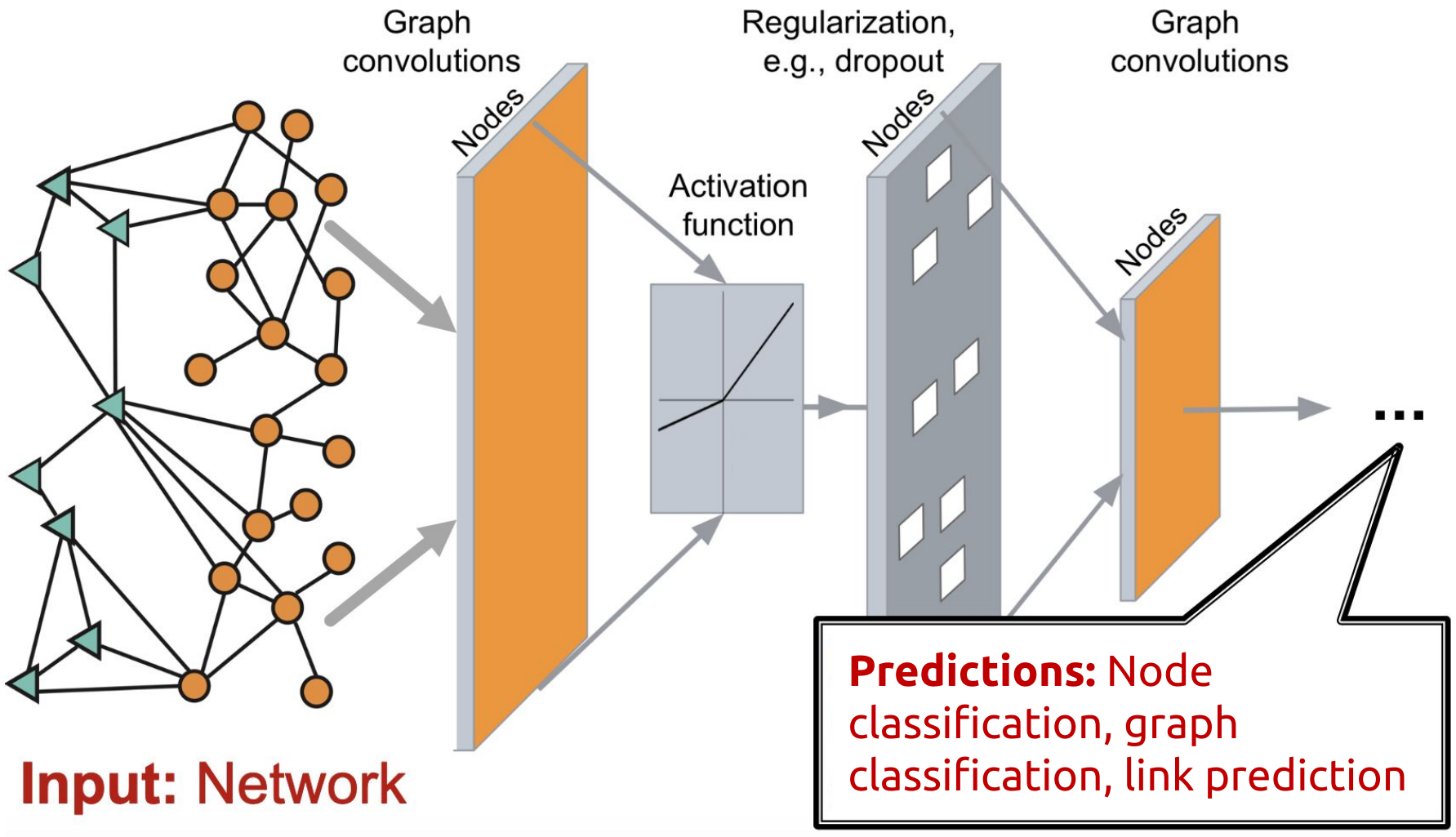
(4) Graph Attention Networks

Deep learning framework for graphs



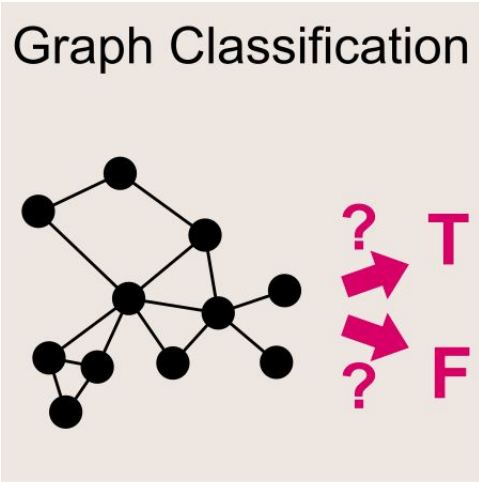
Slide courtesy: http://web.stanford.edu/class/cs224w/slides/01-intro.pdf

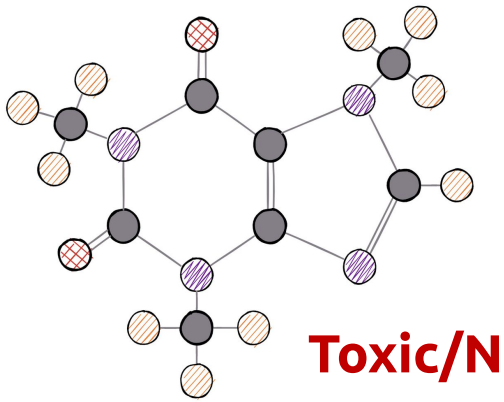
Deep learning framework for graphs



Slide courtesy: http://web.stanford.edu/class/cs224w/slides/01-intro.pdf

Prediction tasks using GNNs





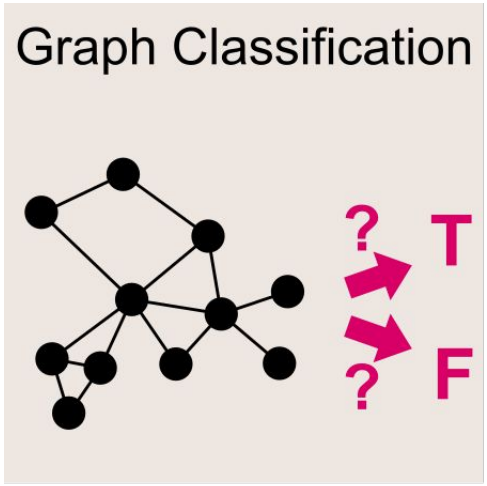
Toxic/Not toxic?

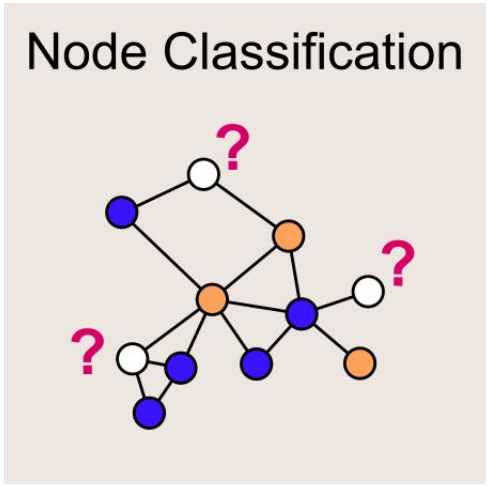
Image courtesy: https://towardsdatascience.com/graph-convolutional-networks-deep-99d7fee5706f

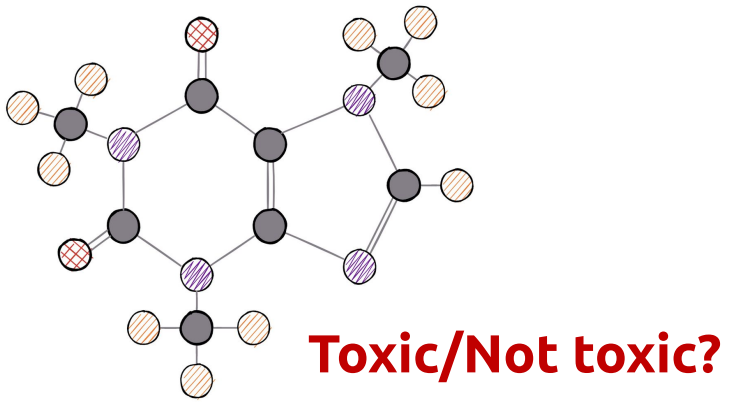
https://memgraph.com/docs/mage/algorithms/machine-learning-graph-analytics/graph-classification-algorithm

https://aws.amazon.com/blogs/machine-learning/graph-based-recommendation-system-with-neptune-ml-an-illustration-on-social-network-link-prediction-challenges/

Prediction tasks using GNNs







Toxic/Not toxic?

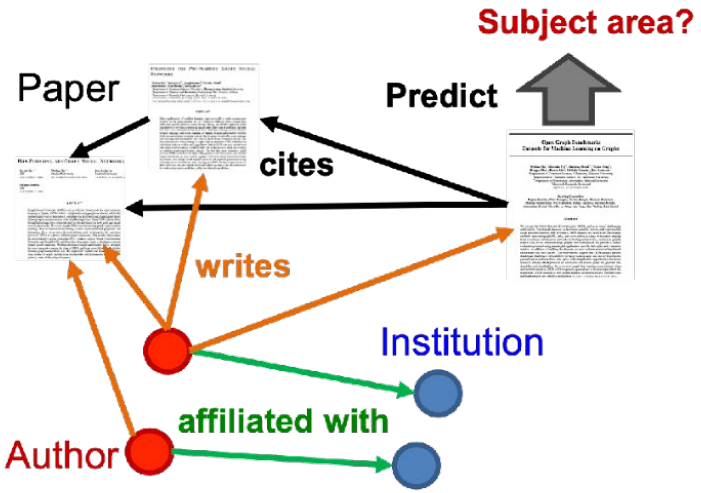
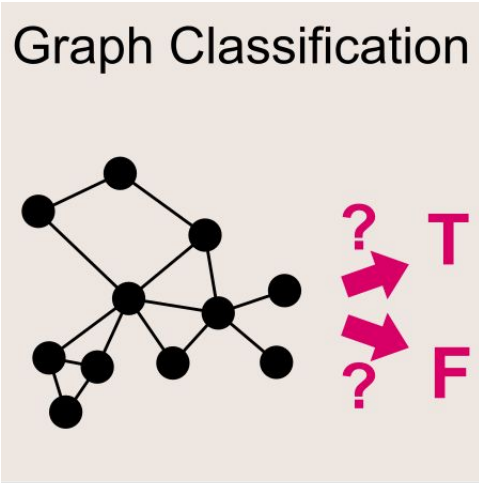


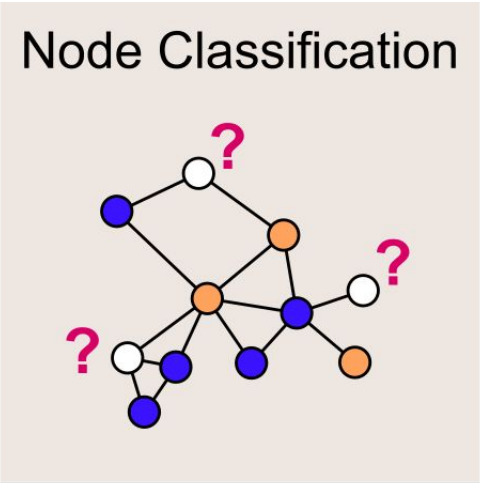
Image courtesy: https://towardsdatascience.com/graph-convolutional-networks-deep-99d7fee5706f

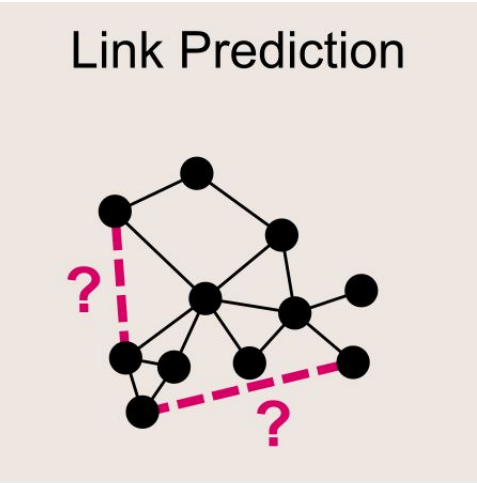
https://memgraph.com/docs/mage/algorithms/machine-learning-graph-analytics/graph-classification-algorithm

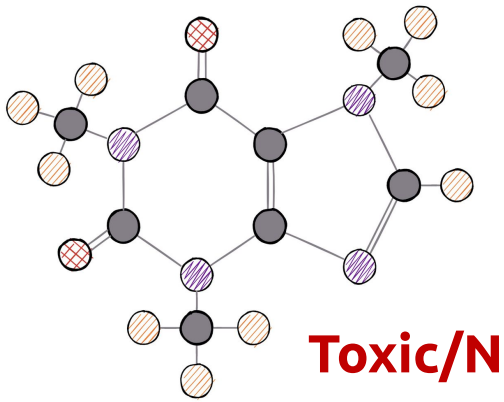
https://aws.amazon.com/blogs/machine-learning/graph-based-recommendation-system-with-neptune-ml-an-illustration-on-social-network-link-prediction-challenges/

Prediction tasks using GNNs

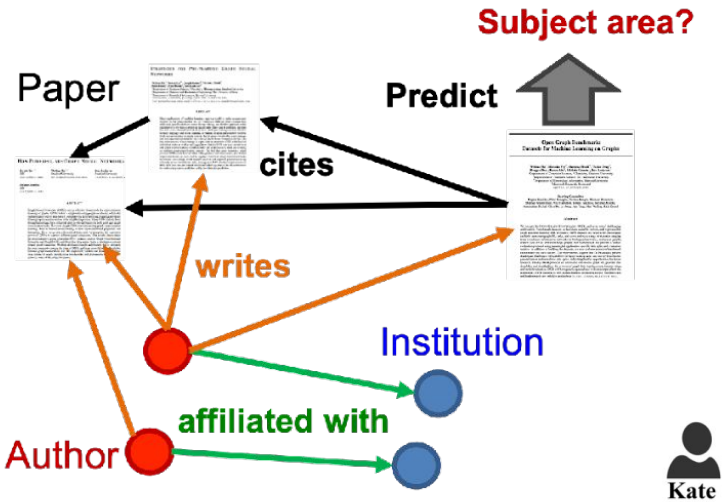








Toxic/Not toxic?



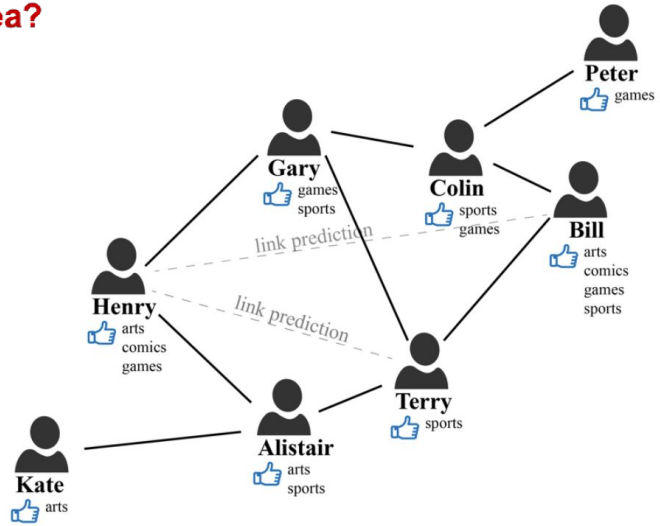
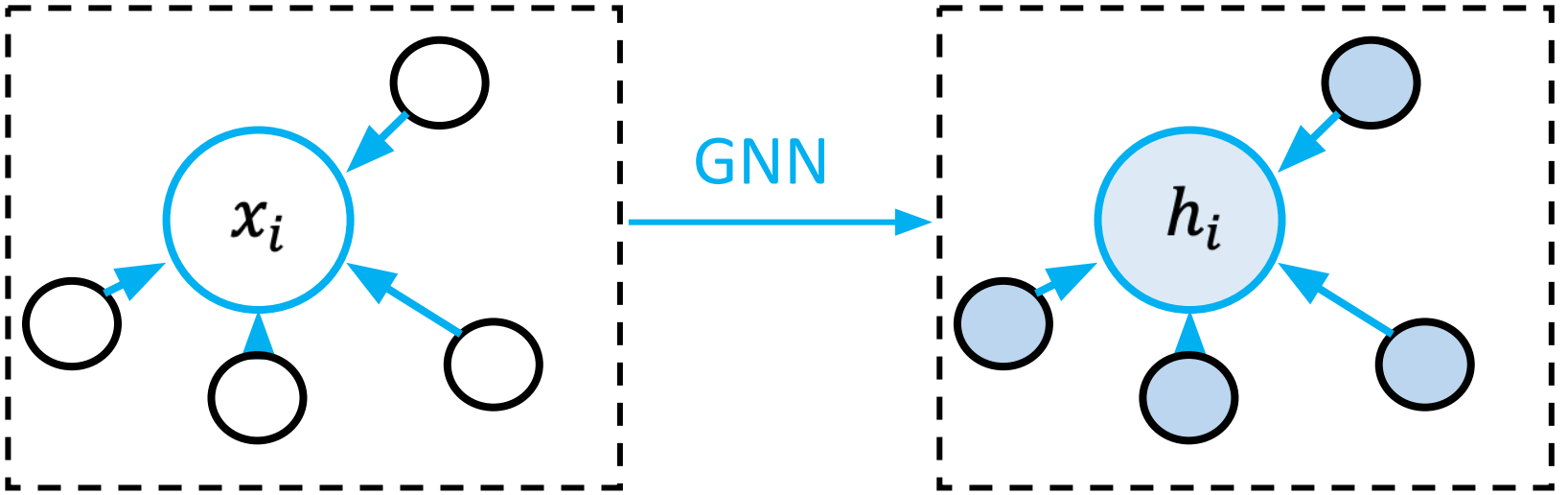


Image courtesy: https://towardsdatascience.com/graph-convolutional-networks-deep-99d7fee5706f

https://memgraph.com/docs/mage/algorithms/machine-learning-graph-analytics/graph-classification-algorithm

https://aws.amazon.com/blogs/machine-learning/graph-based-recommendation-system-with-neptune-ml-an-illustration-on-social-network-link-prediction-challenges/

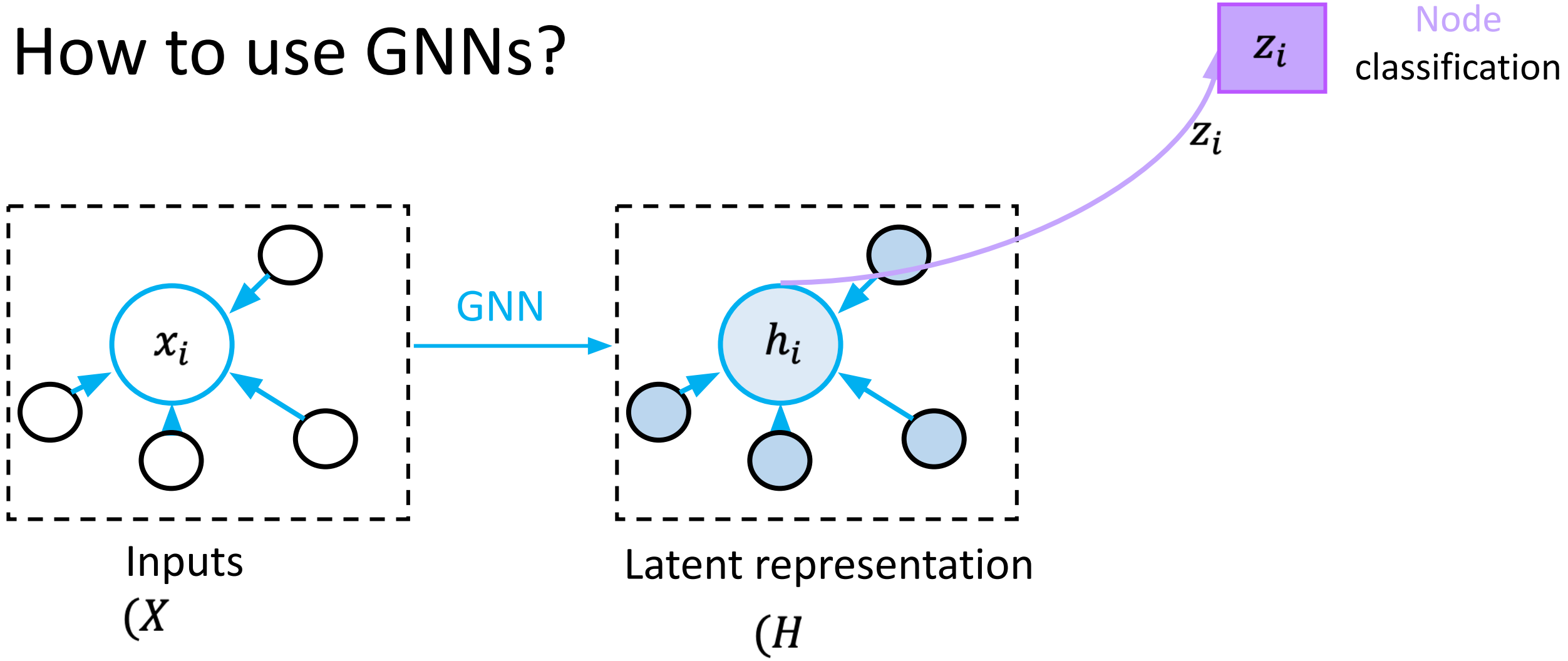
How to use GNNs?



Think-pair-share:
Given the learned representation of the nodes, what do we do next to perform:
(1) Node classification,
(2) graph classification,
(3) link prediction

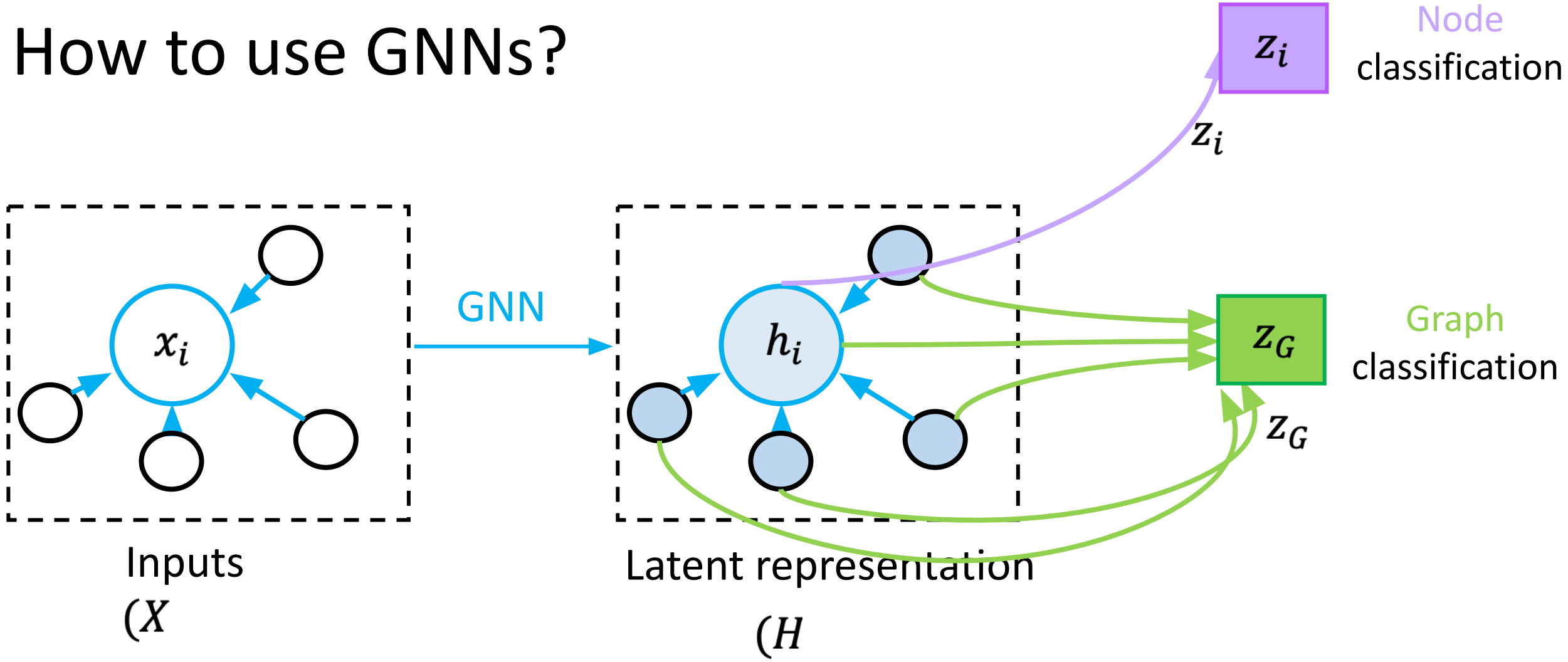
Slide design borrowed from **Petar Veličković's talk**

How to use GNNs?



Slide design borrowed from **Petar Veličković's talk**

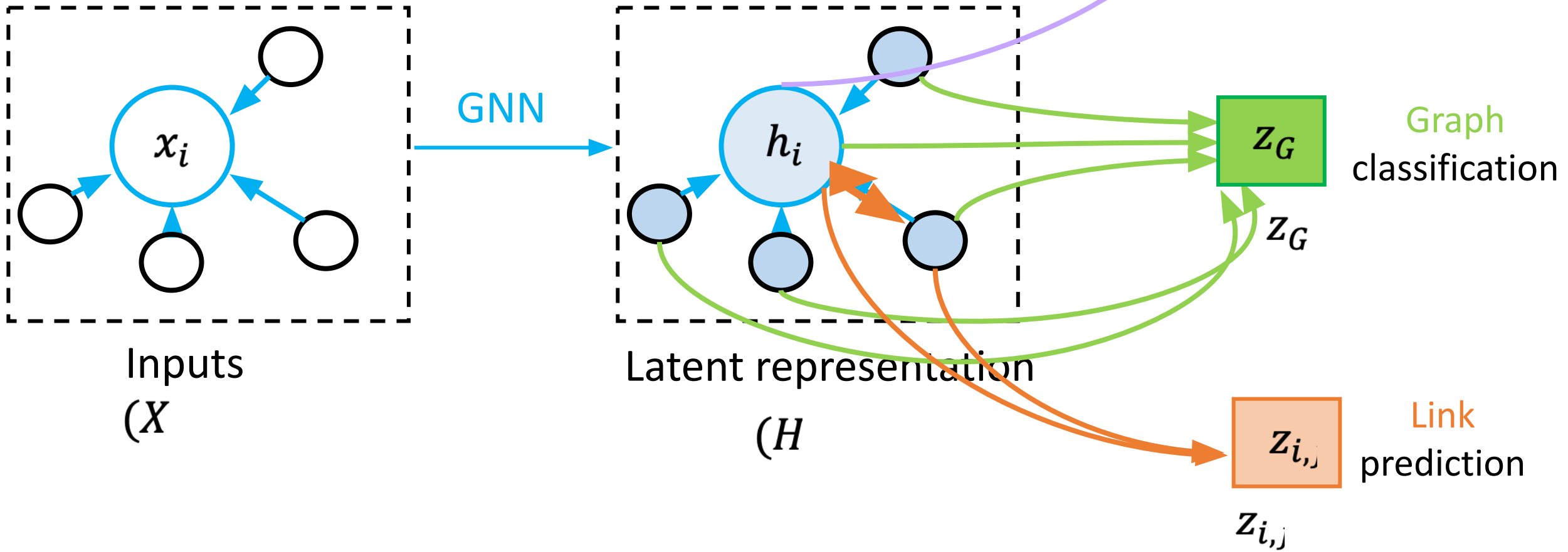
How to use GNNs?



Slide design borrowed from **Petar Veličković's talk**

How to use GNNs?

Any questions?



Slide design borrowed from **Petar Veličković's talk**

Today's goal – learn about graph-based neural networks (GNNs)

(1) Motivation

(2) Passing information in Graphs (for NNs)

(3) How to use GNNs?

(4) Graph Attention Networks

Our formulation so far

We give equal importance to all nodes through normalized aggregation and learn the weights implicitly during training

What if we could attend to different nodes based on their relevance (explicit weighting) ?

$$\mathbf{h}_v^0 = \mathbf{x}_v$$

$$\mathbf{h}_v^k = \sigma\left(\mathbf{W}_k \sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{|N(v)|} + \mathbf{B}_k \mathbf{h}_v^{k-1}\right), \quad \forall k \in \{1, ..., K\}$$

$$\mathbf{z}_v = \mathbf{h}_v^K$$

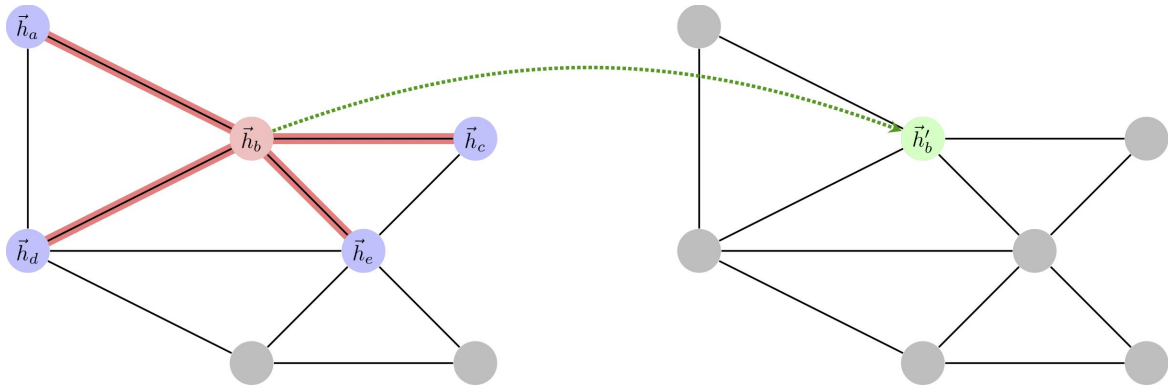
Initial 0-th layer embeddings are equal to node features

Previous layer embedding of v

Average of neighbor's previous layer embeddings

Non-linearity (e.g., ReLU)

Embedding after K layers of neighborhood aggregation



https://petar-v.com/GAT/

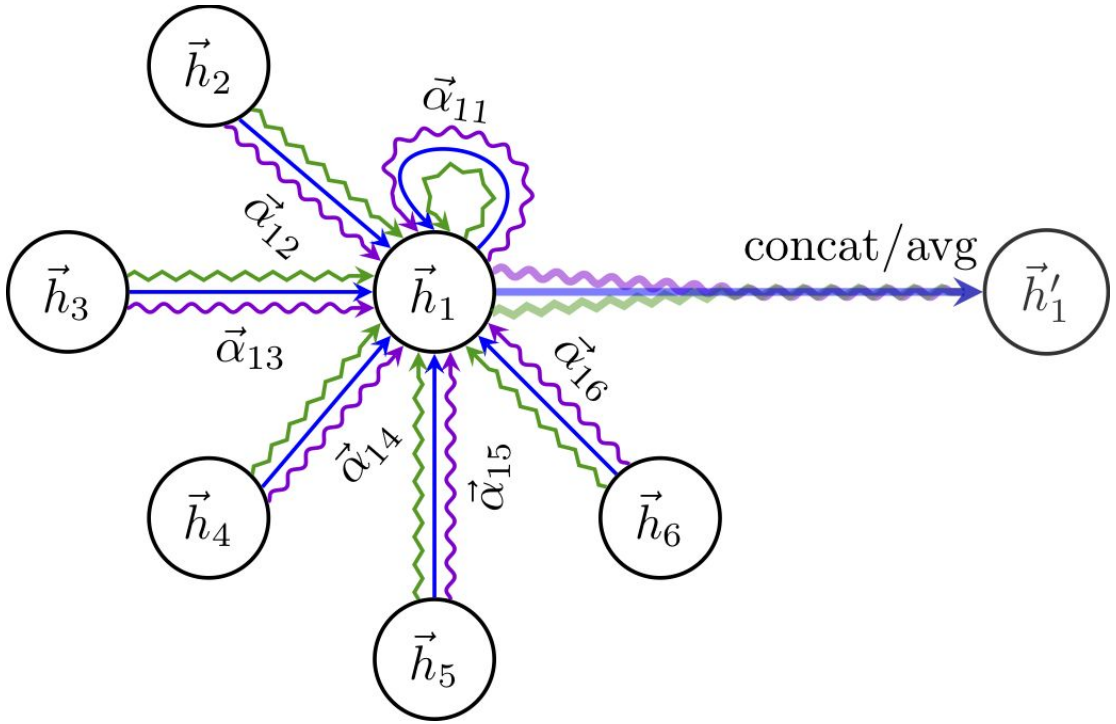
Graph Attention Networks (GAT)

Let's walk through attention steps:

(1) Alignment score $e_{ij} = a(\vec{h}_i, \vec{h}_j)$

(2) Softmax $\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}$

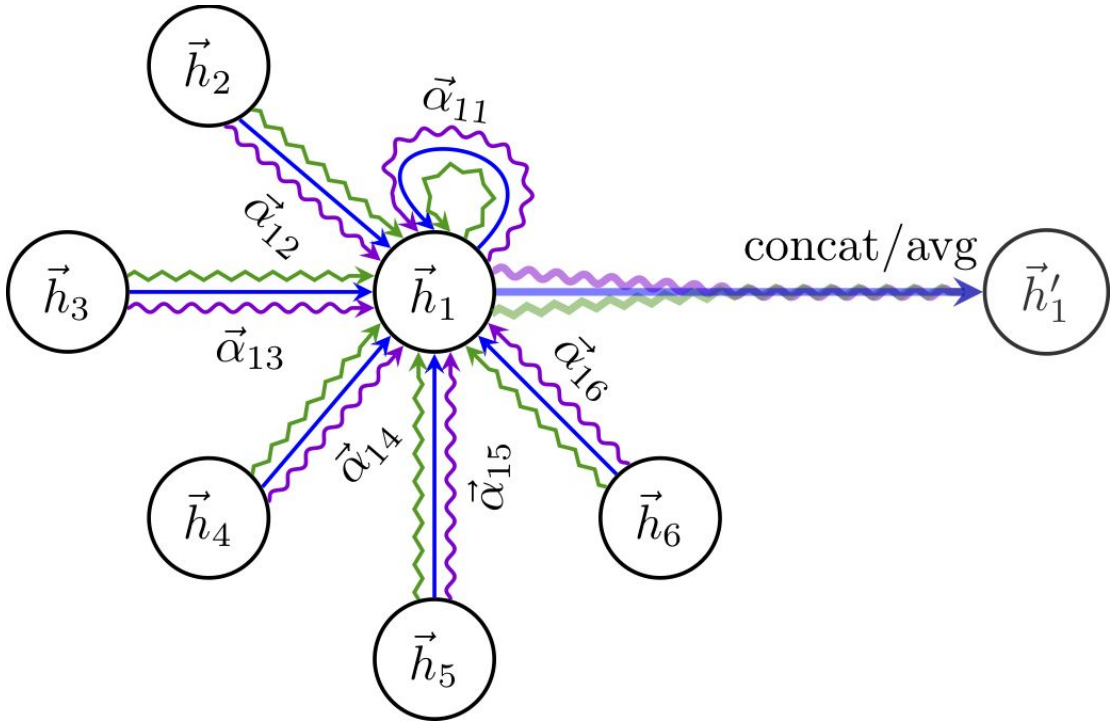
(3) Weighted aggregation $\vec{h}'_i = \Big\|_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \vec{h}_j\right)$



https://petar-v.com/GAT/

Graph Attention Networks (GAT)

Method	PPI
Random	0.396
MLP	0.422
GraphSAGE-GCN (Hamilton et al., 2017)	0.500
GraphSAGE-mean (Hamilton et al., 2017)	0.598
GraphSAGE-LSTM (Hamilton et al., 2017)	0.612
GraphSAGE-pool (Hamilton et al., 2017)	0.600
GraphSAGE*	0.768
Const-GAT (ours)	0.934 ± 0.006
GAT (ours)	$\mathbf{0.973} \pm 0.002$

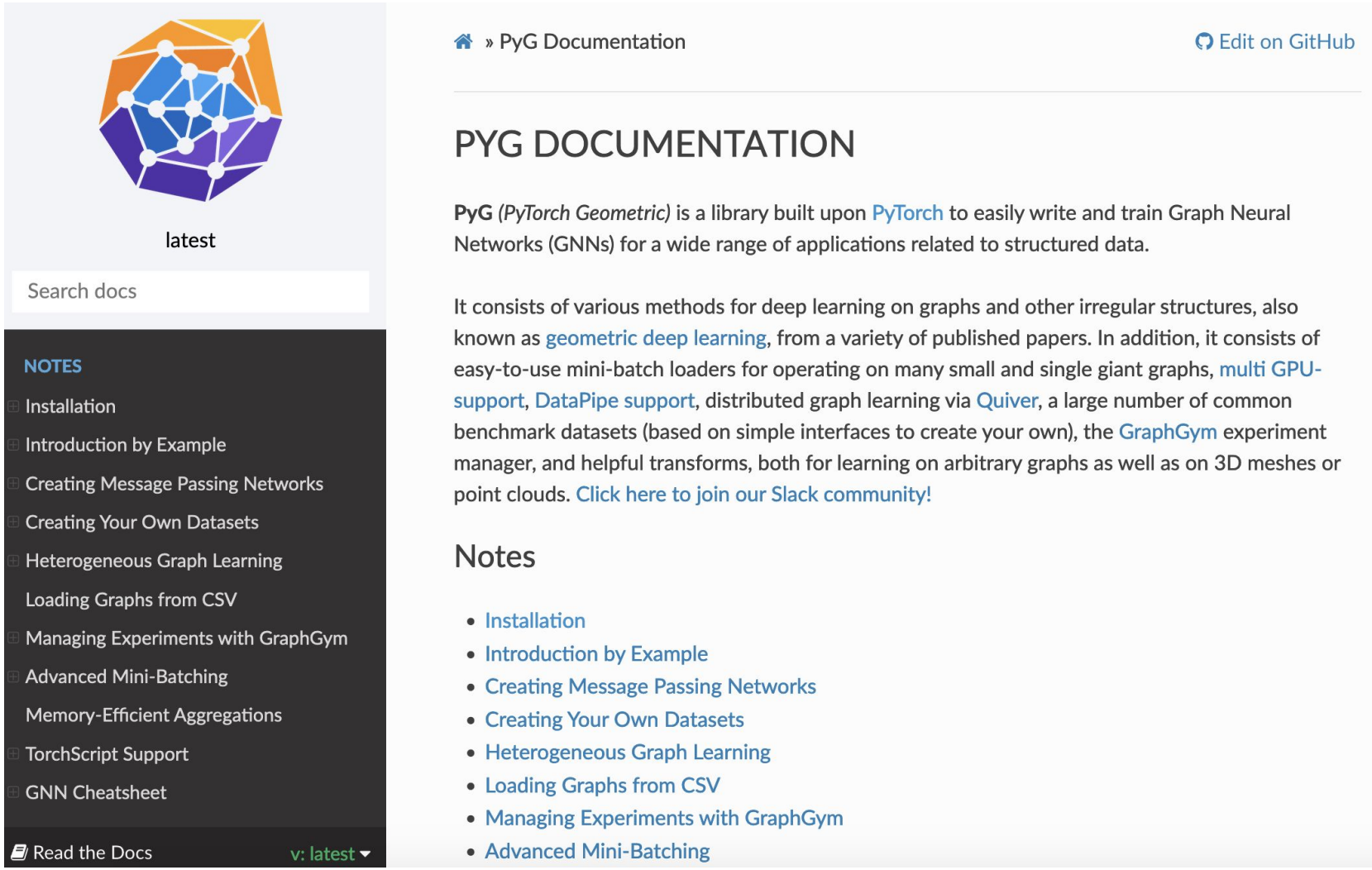


Protein-Protein Interaction dataset:

- consists of graphs corresponding to different human tissue
- contains 20 graphs for training, 2 for validation and 2 for testing.

How to implement a GNN?

https://pytorch-geometric.readthedocs.io/en/latest/



Recap

Graph-based Neural Networks (GNNs)

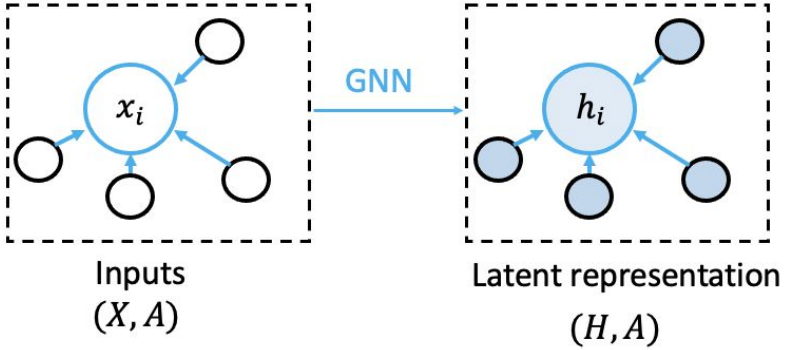
- Motivation
- Message Passing
- GraphSAGE formulation

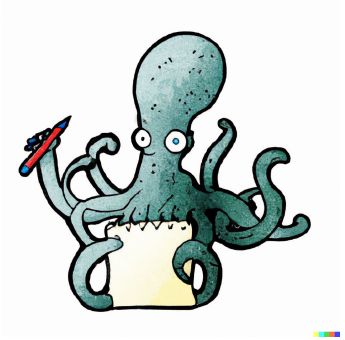
Image credit: Medium

Social Networks

Practical aspects of GNNs

- Prediction tasks
- How to use GNNs
- Graph Attention Networks





Extra reading material

- Summary of graph convolution using graph Fourier transform
- Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering
- Semi-Supervised Classification with Graph Convolutional Networks