FSAN/ELEG815: Statistical Learning

Gonzalo R. Arce

Department of Electrical and Computer Engineering
University of Delaware

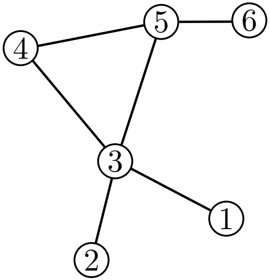
Graph Neural Networks

Graphs

Graphs are denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

$$\mathcal{V} \triangleq \text{Set of Vertices or Nodes}$$
$$\mathcal{E} \triangleq \text{Set of Edges}$$

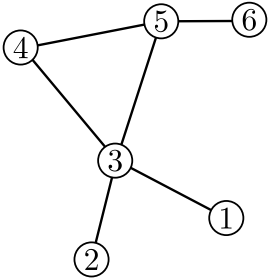
With $|\mathcal{V}| = n$, the *Adjacency Matrix* $\mathbb{A} \in \mathbb{R}^{n \times n}$ indicates if a pair of vertices is connected such as



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

Graphs

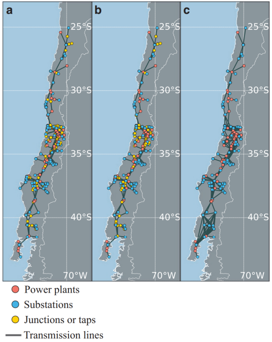
With $|\mathcal{V}| = n$, the *Degree Matrix* $\mathcal{D} \in \mathbb{R}^{n \times n}$ indicates how many edges terminate in each vertex



$$\mathcal{D} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Examples

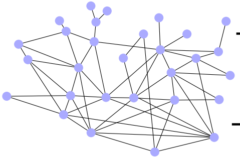
Graph representation of Chilean power grid



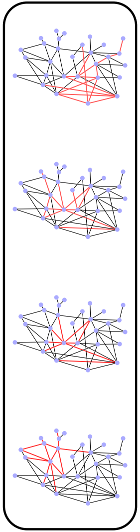
NH_2CNO_3

Chemical Formula

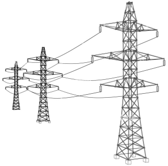
Topology

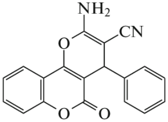


Graph NN



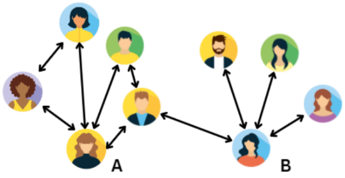
Optimal Power flow





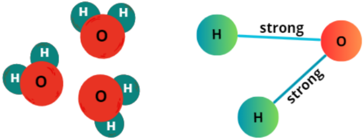
Structure and Transitions

Graphs are Everywhere



Social networks

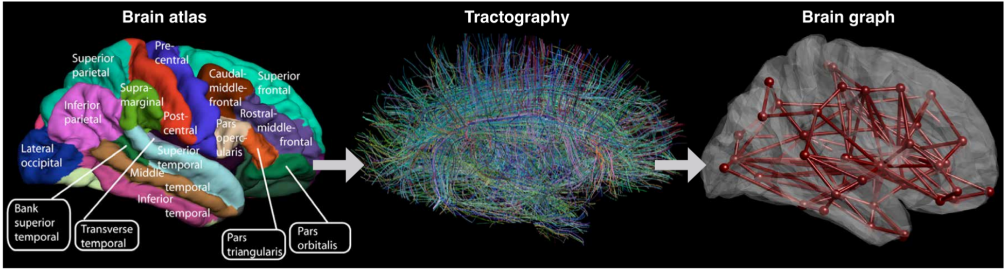
Recommender Systems



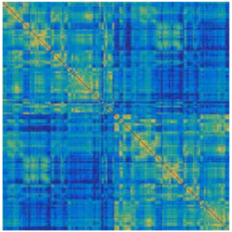
Chemical compounds

3D Games / Meshes

Graphs are Everywhere

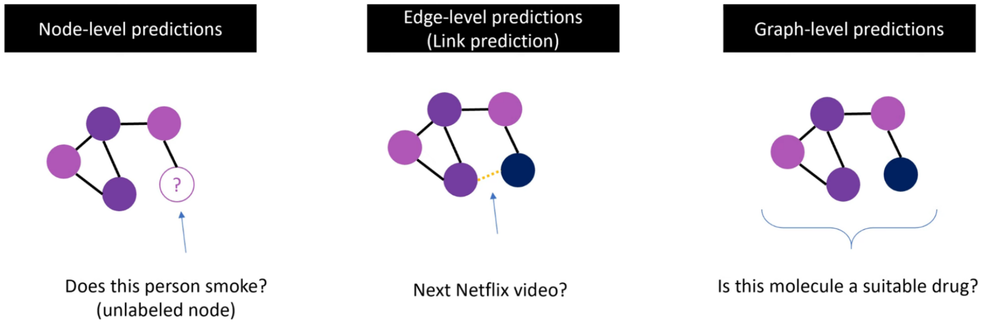


Adjacency Matrix.
Structural Connectome

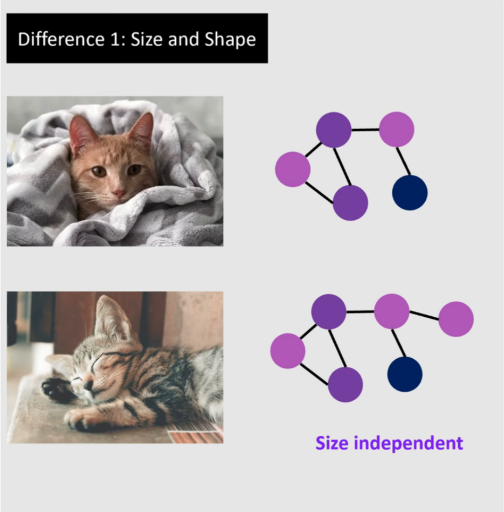


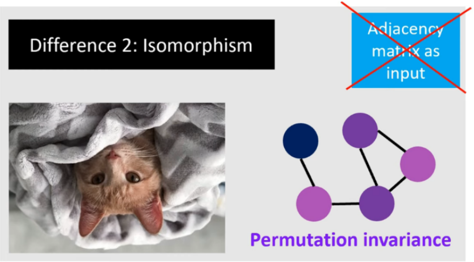
Connection between
Brain Regions

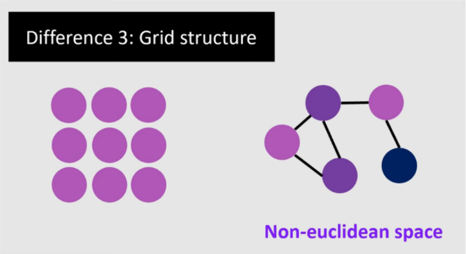
Machine Learning Problems With Graph Data



Graph Data Problems

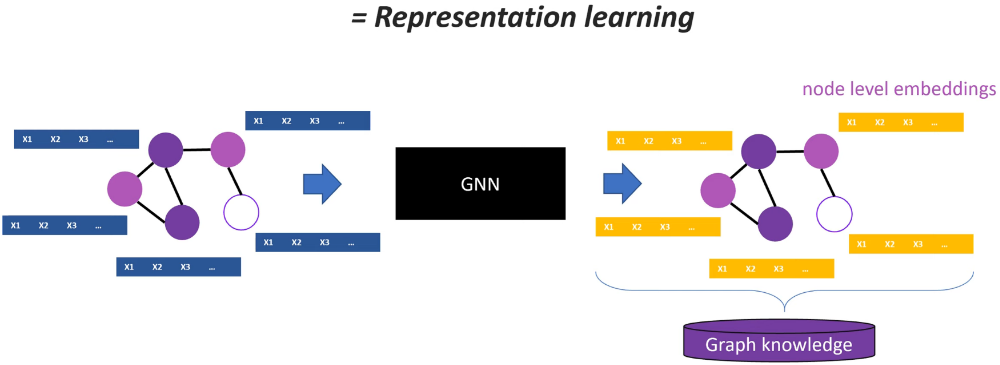




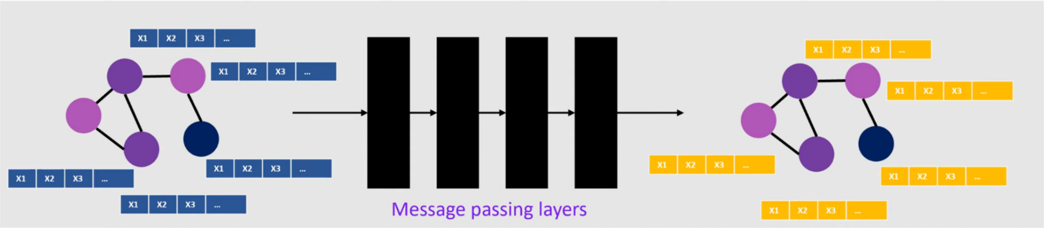


Fundamental Idea of GNNs

Learning a Neural Network suitable representation of graph data

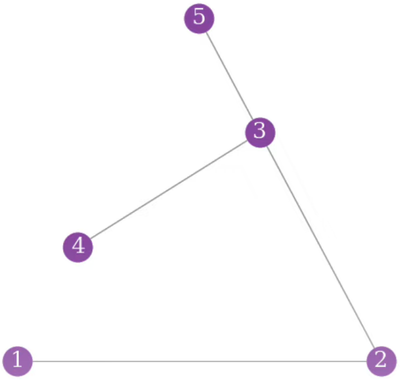


How GNNs Work



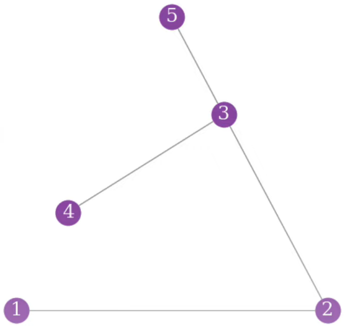
Adjacency Matrix

	$Node_1$	$Node_2$	$Node_3$	$Node_4$	$Node_5$
$Node_1$	0	1	0	0	0
$Node_2$	1	0	1	0	0
$Node_3$	0	1	0	1	1
$Node_4$	0	0	1	0	0
$Node_5$	0	0	1	0	0



Node Degree

	$Node_1$	$Node_2$	$Node_3$	$Node_4$	$Node_5$	
$Node_1$	0	1	0	0	0	1
$Node_2$	1	0	1	0	0	2
$Node_3$	0	1	0	1	1	3
$Node_4$	0	0	1	0	0	1
$Node_5$	0	0	1	0	0	1
	1	2	3	1	1	



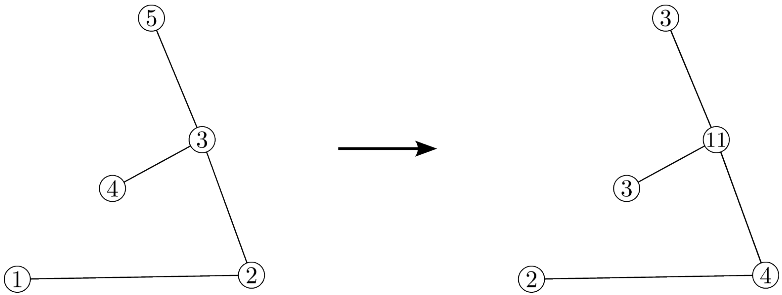
$$\mathcal{D} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Message Passing - Simple Form

$$\mathbb{A}\mathbf{h}$$

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} = \begin{bmatrix} (0)(1)+(1)(2)+(0)(3)+(0)(4)+(0)(5) \\ (1)(1)+(0)(2)+(1)(3)+(0)(4)+(0)(5) \\ (0)(1)+(1)(2)+(0)(3)+(1)(4)+(1)(5) \\ (0)(1)+(0)(2)+(1)(3)+(0)(4)+(0)(5) \\ (0)(1)+(0)(2)+(1)(3)+(0)(4)+(0)(5) \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 11 \\ 3 \\ 3 \end{bmatrix}$$

$\mathbf{h} \triangleq$ Features on nodes



Message Passing - Average of Messages

Degree Matrix

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Inverse Degree Matrix

$$D^{-1} = \begin{bmatrix} 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0.3 & 0 & 0 \\ 0 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 \end{bmatrix}$$

Message Passing - Average of Messages

Average Adjacency Matrix

$$D^{-1}A = A_{avg}$$

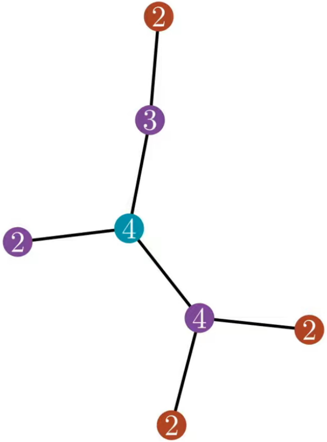
$$\begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.5 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.3 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.5 & 0.0 & 0.5 & 0.0 & 0.0 \\ 0.0 & 0.3 & 0.0 & 0.3 & 0.3 \\ 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \end{bmatrix}$$

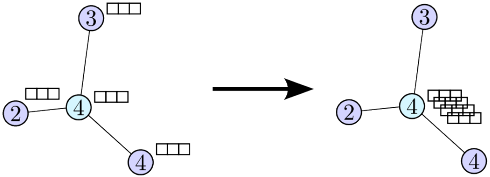
Message Passing - Self Connection for Message Passing

$$\tilde{A} = A + I \qquad \hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}} \qquad \hat{A}_{i,j} = \frac{1}{\sqrt{\tilde{d}_i \tilde{d}_j}}\tilde{A}_{i,j}$$

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

Graph Convolutional Networks - Simple Average



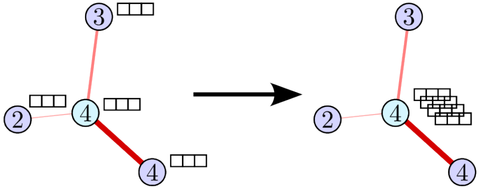


Simple Average Aggregation

$$\mathbf{h}_{\mathcal{N}(v)} = \frac{1}{|\mathcal{N}(v)|} \sum_{u \in \mathcal{N}(v)} \mathbf{h}_u$$

$\mathcal{N}(v)$ are the nodes in the neighborhood of node v

Graph Convolutional Networks - Weighted Average



Weighted Average Aggregation

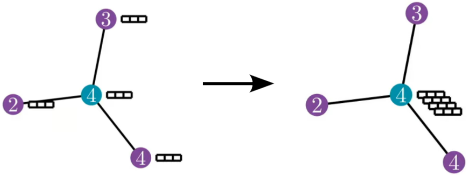
$$\mathbf{h}_{\mathcal{N}(v)} = \sum_{u \in \mathcal{N}(v)} w_{u,v} \mathbf{h}_u$$

$$= \sum_{u \in \mathcal{N}(v)} \sqrt{\frac{1}{d_v}} \sqrt{\frac{1}{d_u}} \mathbf{h}_u$$

$$= \sqrt{\frac{1}{d_v}} \sum_{u \in \mathcal{N}(v)} \sqrt{\frac{1}{d_u}} \mathbf{h}_u$$

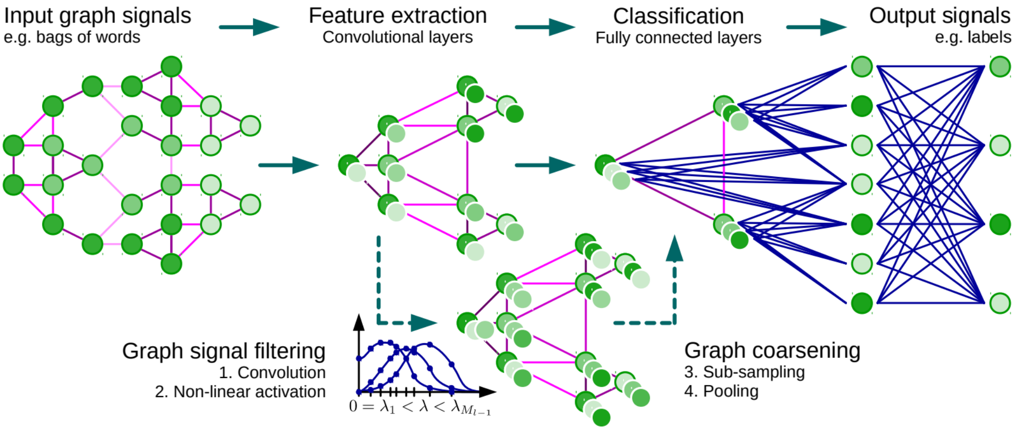
d_u and d_v are the degree of node u and v respectively

Graph Convolutional Networks - Weighted Average



$$\mathbf{h}_{\mathcal{N}(0)} = \frac{1}{\sqrt{4}}\left(\frac{\mathbf{h}_0}{\sqrt{4}} + \frac{\mathbf{h}_1}{\sqrt{3}} + \frac{\mathbf{h}_2}{\sqrt{2}} + \frac{\mathbf{h}_3}{\sqrt{4}}\right)$$

Graph Convolutional Networks



Graph Attention Networks

$$\mathbf{h}_{\mathcal{N}(v)} = \sum_{u \in \mathcal{N}(v)} w_{u,v} \mathbf{h}_u \longrightarrow \mathbf{h}_{\mathcal{N}(v)} = \sum_{u \in \mathcal{N}(v)} a(\mathbf{h}_u, \mathbf{h}_v) \mathbf{h}_u$$

$a(h_u, h_v)$ is the attention coefficient between nodes u and v

Original Paper Suggestion

$$\mathbf{h}_{\mathcal{N}(v)} = \sum_{u \in \mathcal{N}(v)} \text{softmax}_u \left(a(\mathbf{h}_u, \mathbf{h}_v) \right) \mathbf{h}_u$$

$\alpha_{u,v} = \text{softmax}(a(\mathbf{h}_u, \mathbf{h}_v))$ is the normalized attention coefficient.

$$\alpha_{u,v} = \frac{\exp(a(\mathbf{h}_u, \mathbf{h}_v))}{\sum_{k \in \mathcal{N}(v)} \exp(a(\mathbf{h}_u, \mathbf{h}_v))}$$

GAT - Attention values

$$\mathbf{h}_{\mathcal{N}(v)} = \sum_{u \in \mathcal{N}(v)} \overbrace{\text{softmax}_u\left(a(\mathbf{h}_u, \mathbf{h}_v)\right)}^{\alpha_{u,v}} \mathbf{h}_u$$

$$a(\mathbf{h}_u, \mathbf{h}_v) = \sigma(\mathbf{a}^T \cdot [\mathbf{W}\mathbf{h}_u || \mathbf{W}\mathbf{h}_v])$$

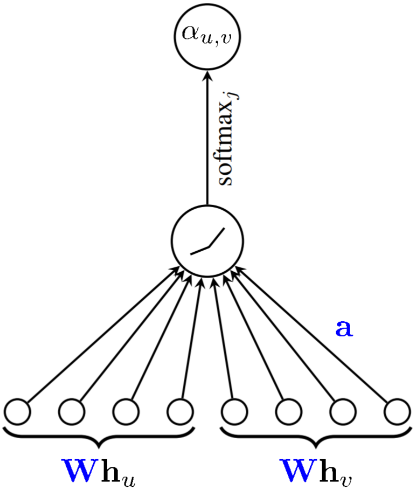
W is a learnable weight matrix that projects the feature vector of nodes u and v.

a^T is a learnable parameter vector that determines the importance of different parts of the concatenated input.

GAT - Attention values

$$a(\mathbf{h}_u, \mathbf{h}_v) = \sigma(\mathbf{a}^T \cdot [\mathbf{W}\mathbf{h}_u || \mathbf{W}\mathbf{h}_v])$$

$$\alpha_{u,v} = \text{softmax}(a(\mathbf{h}_u, \mathbf{h}_v))$$



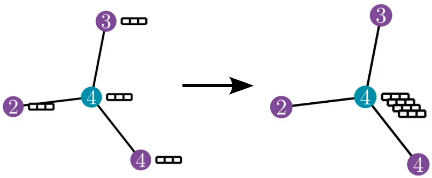
GAT - Attention values

$$a(\mathbf{h}_u, \mathbf{h}_v) = \sigma(\mathbf{a}^T \cdot [\mathbf{W}\mathbf{h}_u || \mathbf{W}\mathbf{h}_v])$$

$$a(\mathbf{h}_u, \mathbf{h}_v) = \text{LeakyReLU}(\mathbf{a}^T \cdot [\mathbf{W}\mathbf{h}_u || \mathbf{W}\mathbf{h}_v])$$

$$\alpha_{uv} = \frac{\exp\left(\text{LeakyReLU}(\mathbf{a}^T \cdot [\mathbf{W}\mathbf{h}_u || \mathbf{W}\mathbf{h}_v])\right)}{\sum_{k \in \mathcal{N}(u)} \exp(\text{LeakyReLU}(\mathbf{a}^T \cdot [\mathbf{W}\mathbf{h}_u || \mathbf{W}\mathbf{h}_k]))}$$

GAT - Attention Aggregation

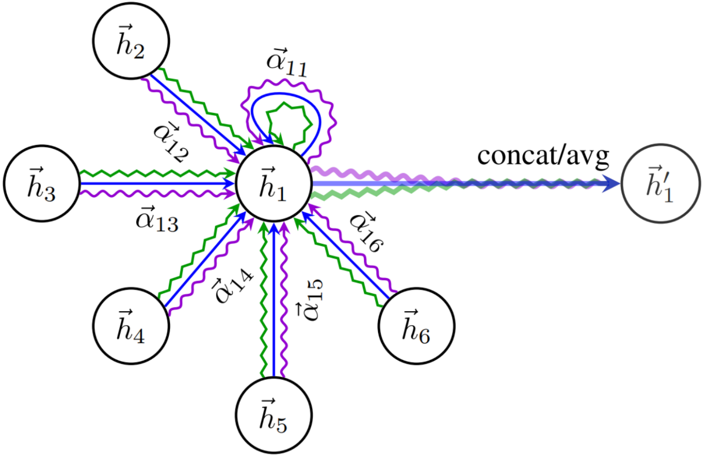


$$\mathbf{h}_u = \sigma\left(\sum_{v \in \mathcal{N}_u} \alpha_{uv} \mathbf{W} \mathbf{h}_v\right)$$

Multi-Head Graph Attention Network

$$\mathbf{h}_u = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{v \in \mathcal{N}_u} \alpha_{uv}^{k} \mathbf{W}^{k} \mathbf{h}_v\right)$$

GAT - Attention Aggregation



Multihead-Attention (with $k = 3$ heads) by node 1 on its neighborhood. Different arrow styles and colors denote independent attention computations. The aggregated features from each head are concatenated or averaged to obtain $\vec{h'_1}$

GNN - Examples

Knowledge Graph for Social Relationship Understanding