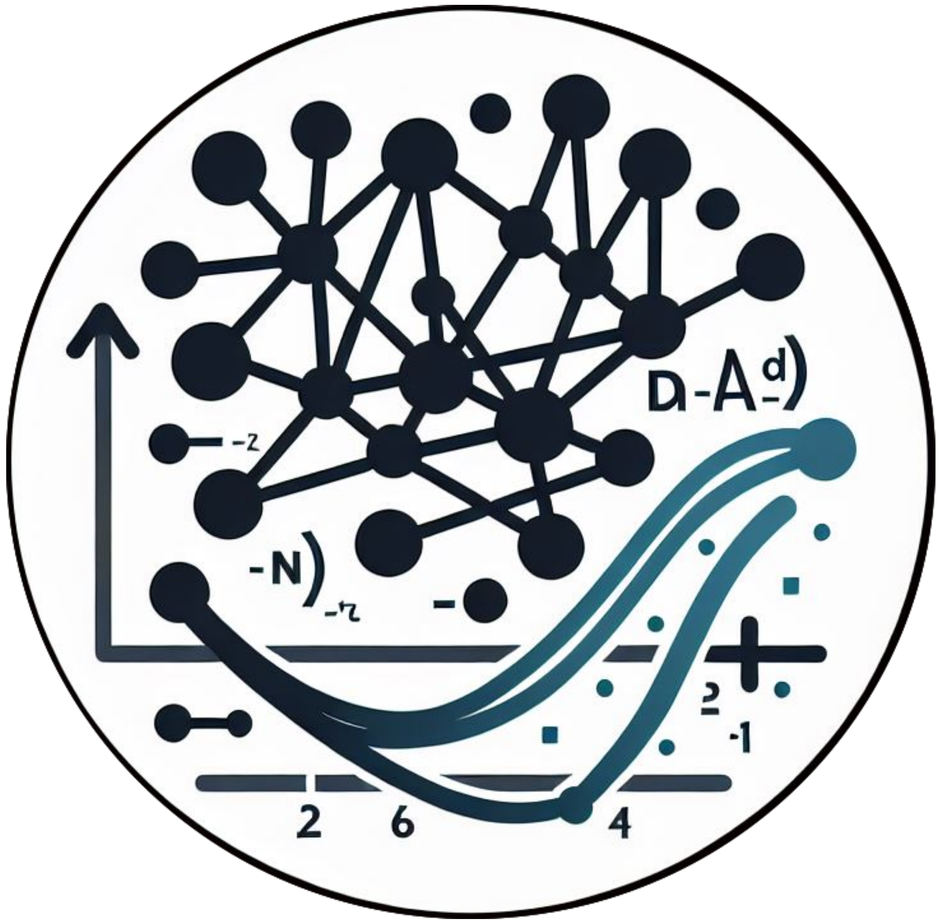
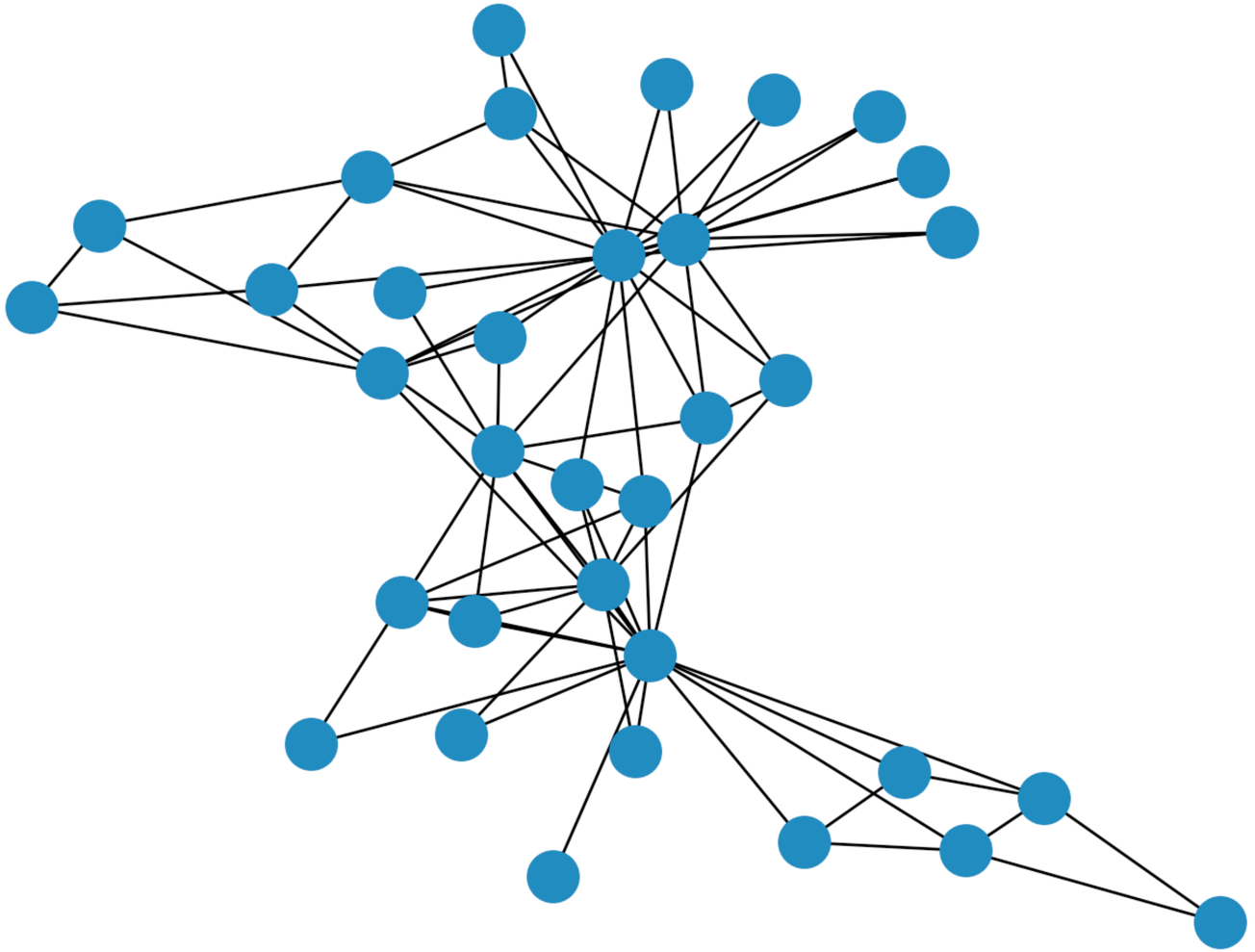
Graph Neural Nets



Deep Learning for Engineers
Andrew Ning
aning@byu.edu

Motivation

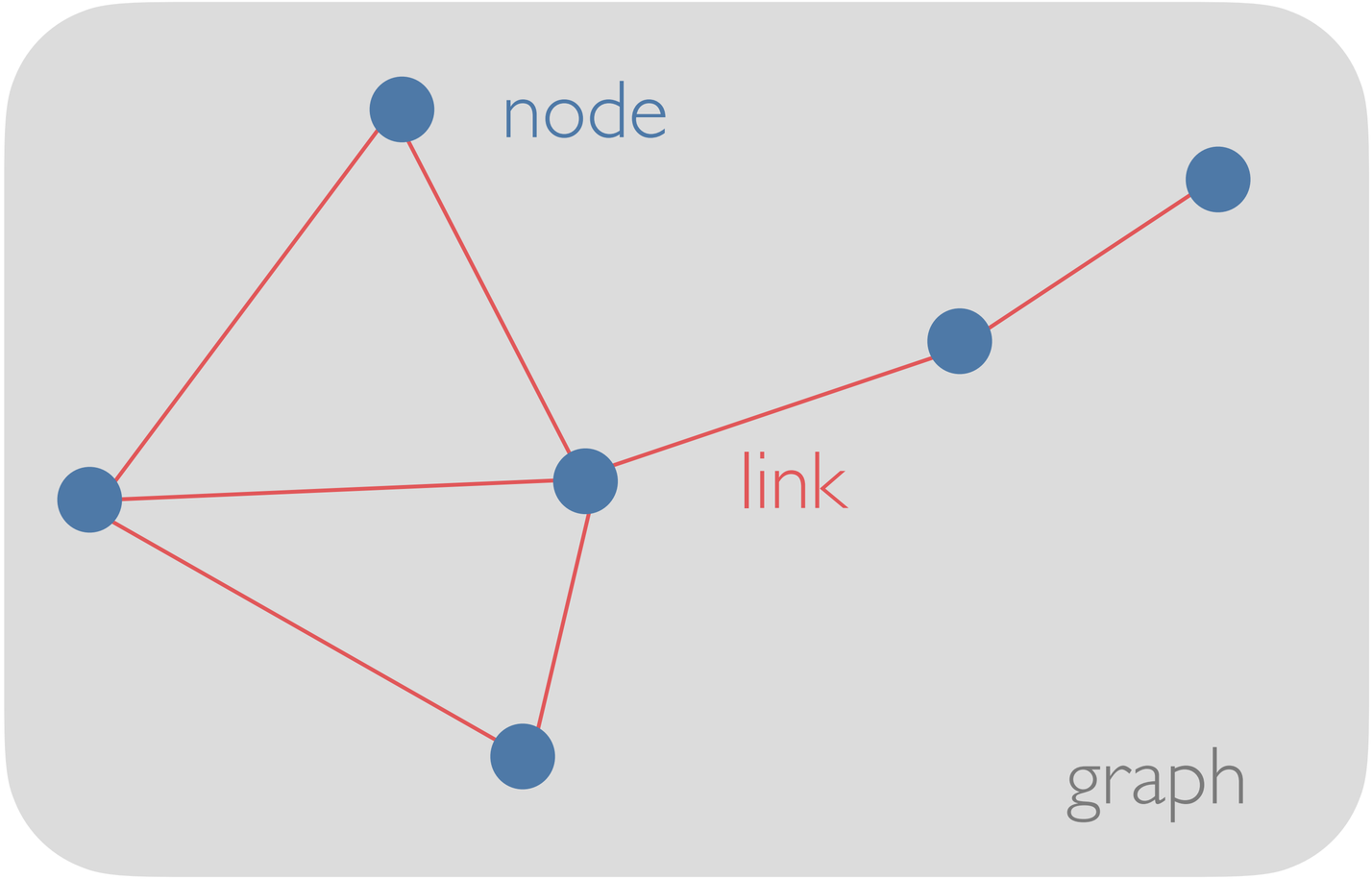


Challenges

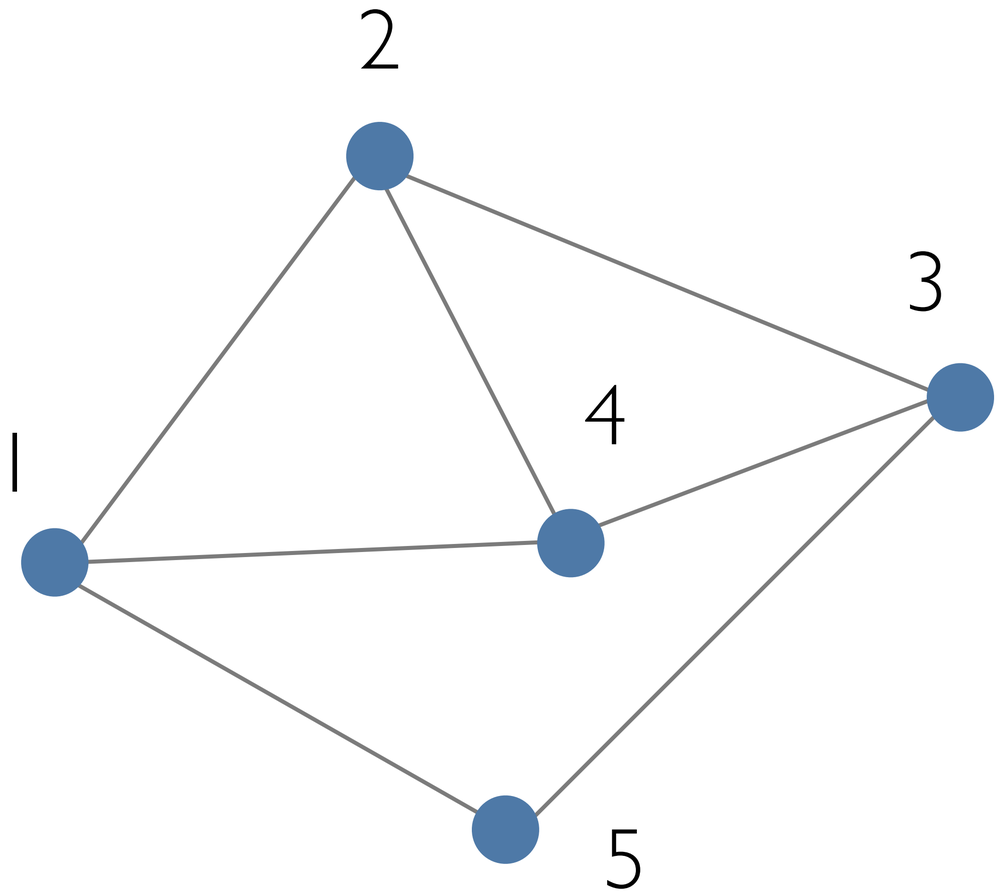
as compared to structured data

- structure is complex and requires more memory
- no natural ordering (would also like functionality to not be sensitive to this ordering)
- less obvious how to deal with different size inputs (graphs)

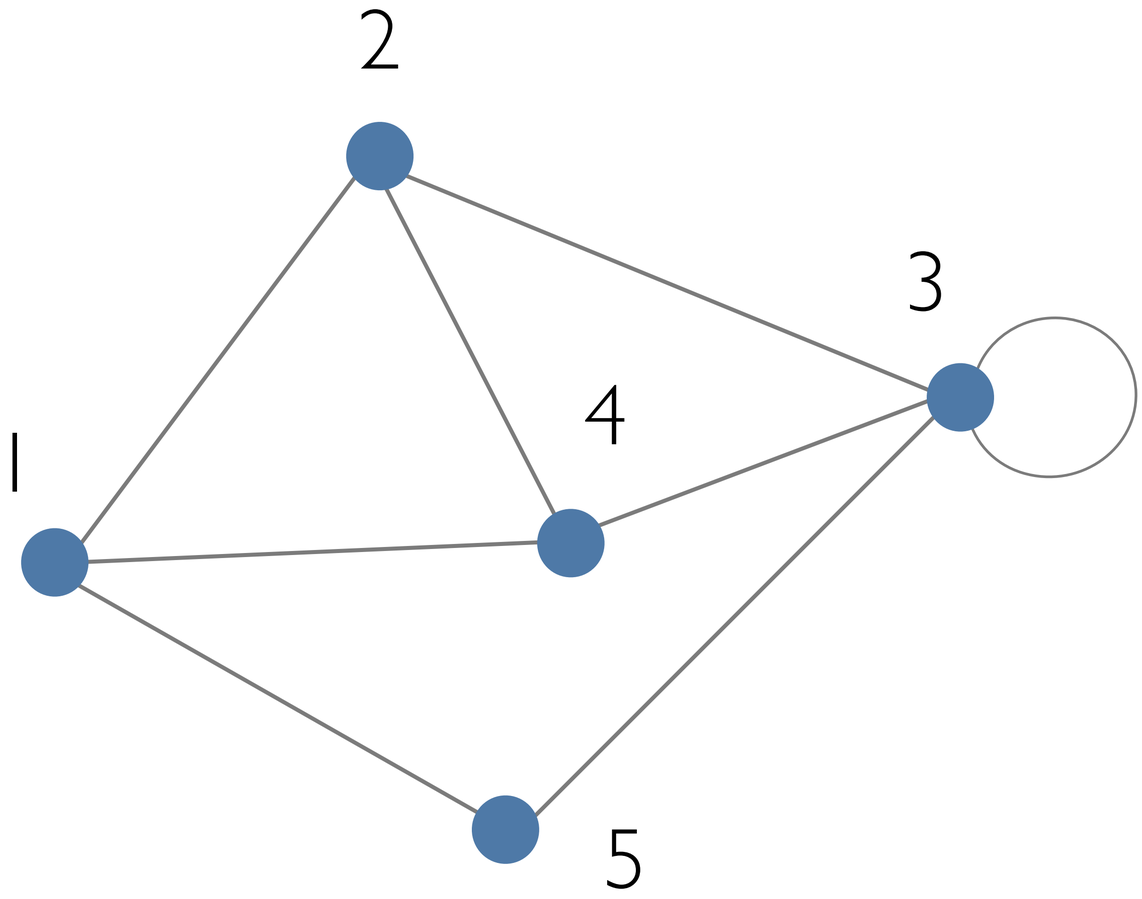
Node, link, and graph classification/regression



Adjacency matrix



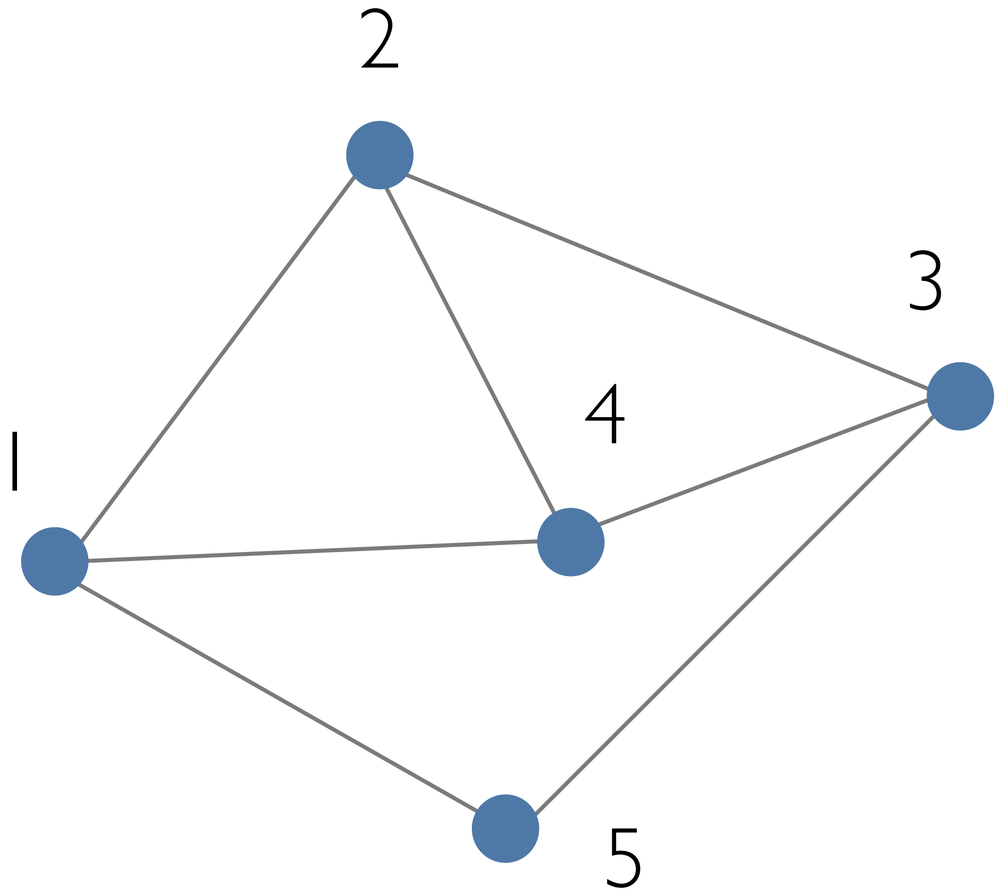
Adjacency matrix



$$A_{ij} = \begin{cases} 1 & \text{connected} \\ 0 & \text{otherwise} \end{cases}$$

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

Features and (Node) Targets/Labels



$$x_3 = \begin{bmatrix} 2.1 \\ 1.6 \\ -1.2 \\ 4.1 \end{bmatrix}$$

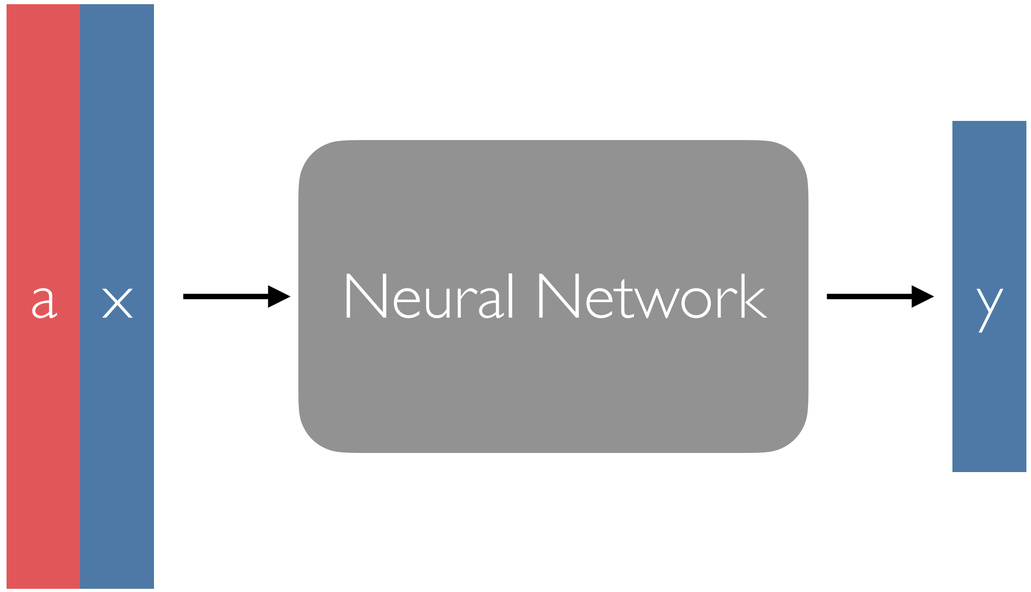
$$y_3 = \begin{bmatrix} 0.3 \\ -1.1 \end{bmatrix}$$

$$X \in \mathcal{R}^{n \times d}$$

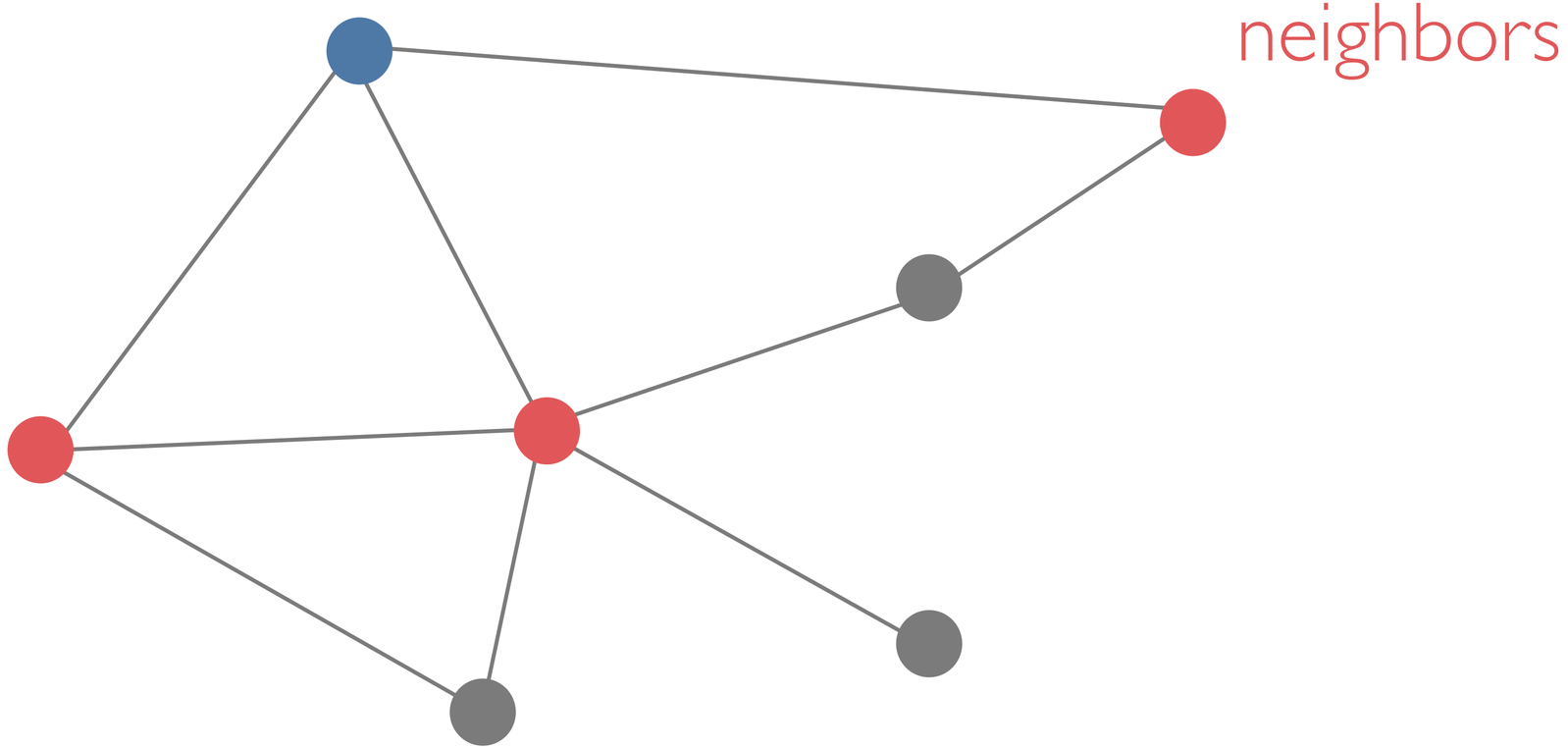
n: number of nodes

d: dimension of x

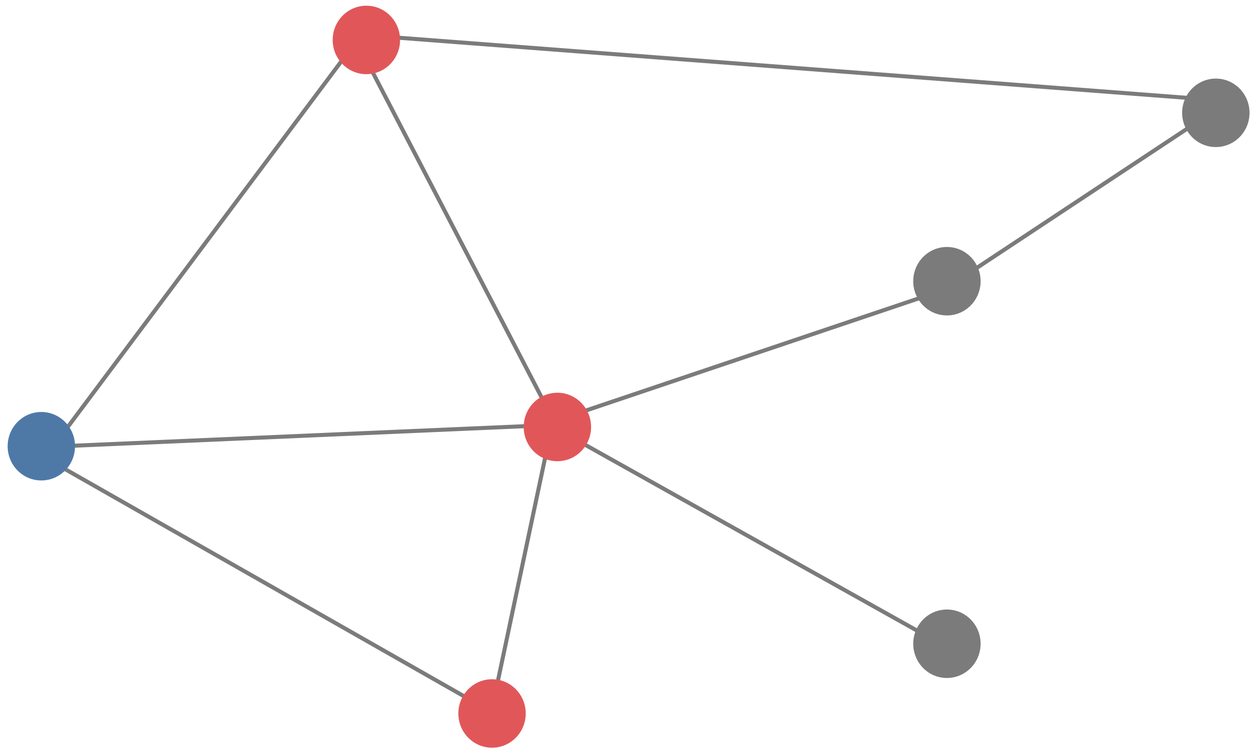
First Approach



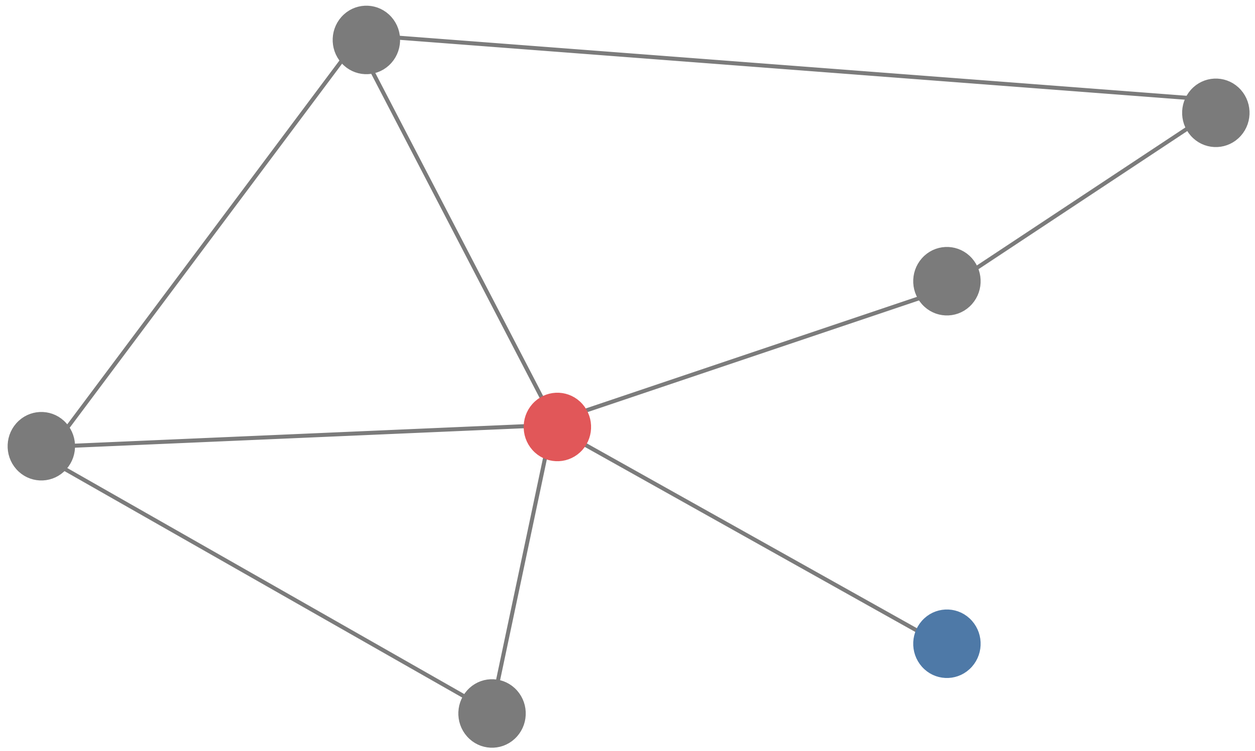
Neighbors



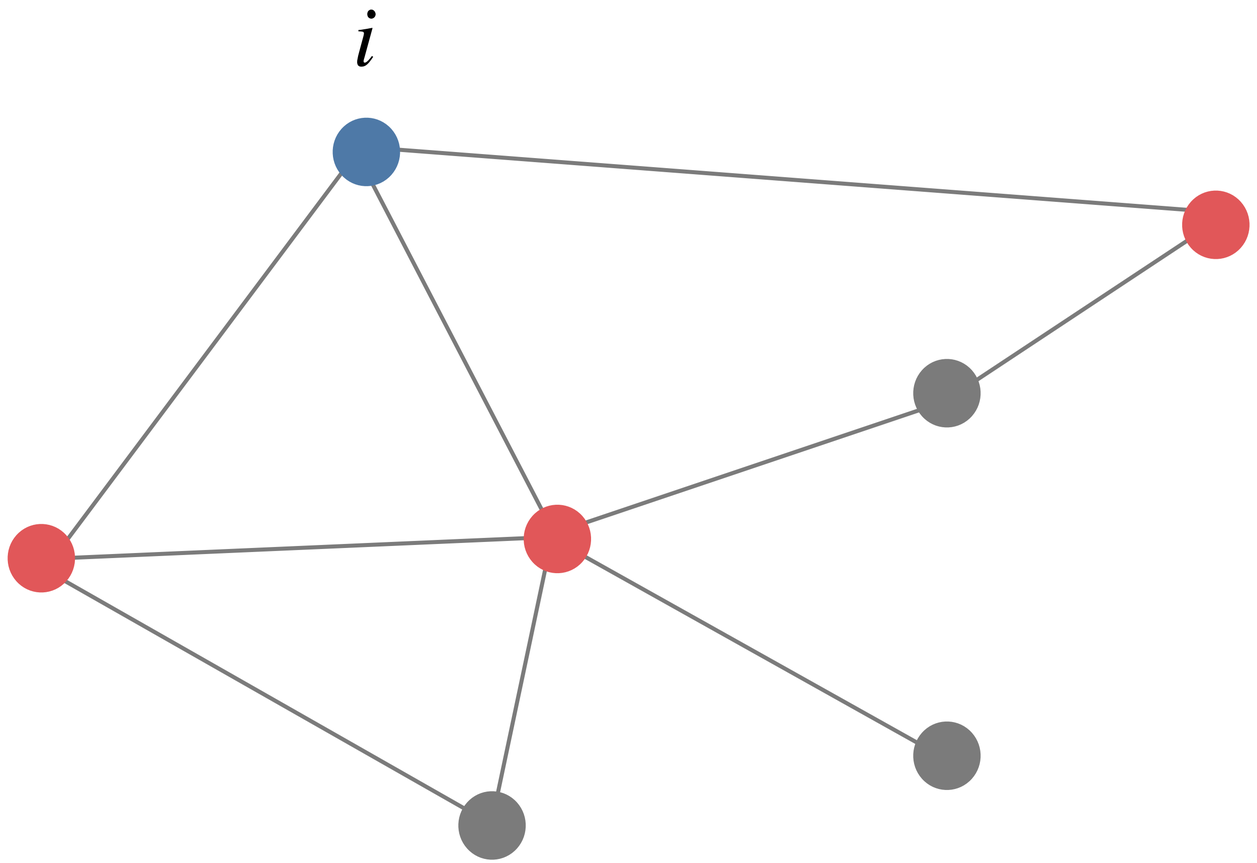
Neighbors



Neighbors



Neighbors

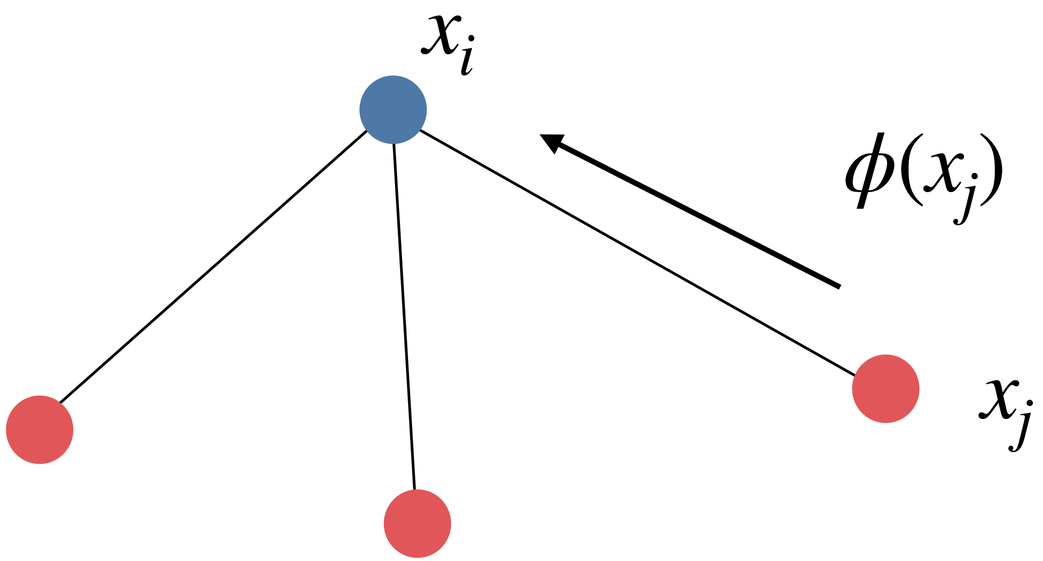


$$\mathcal{N}(i) = \{j : a_{ij} \in A\}$$

Message Passing GNN

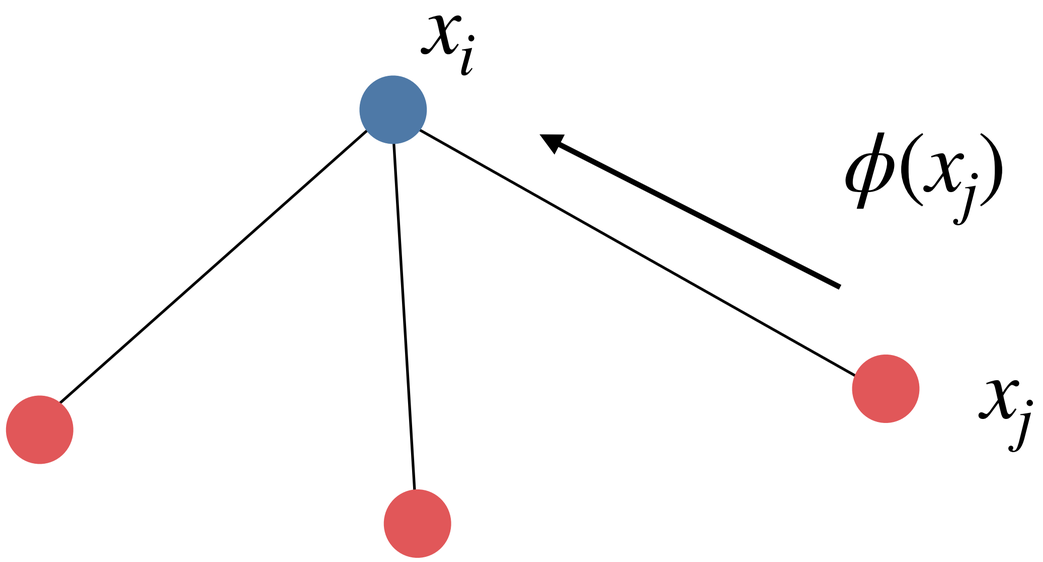
1) Each node constructs a message for its neighbors
2) Messages are sent and each node aggregates messages from all its neighbors
3) Each node updates its attributes

Create a message



message passing function

Create a message



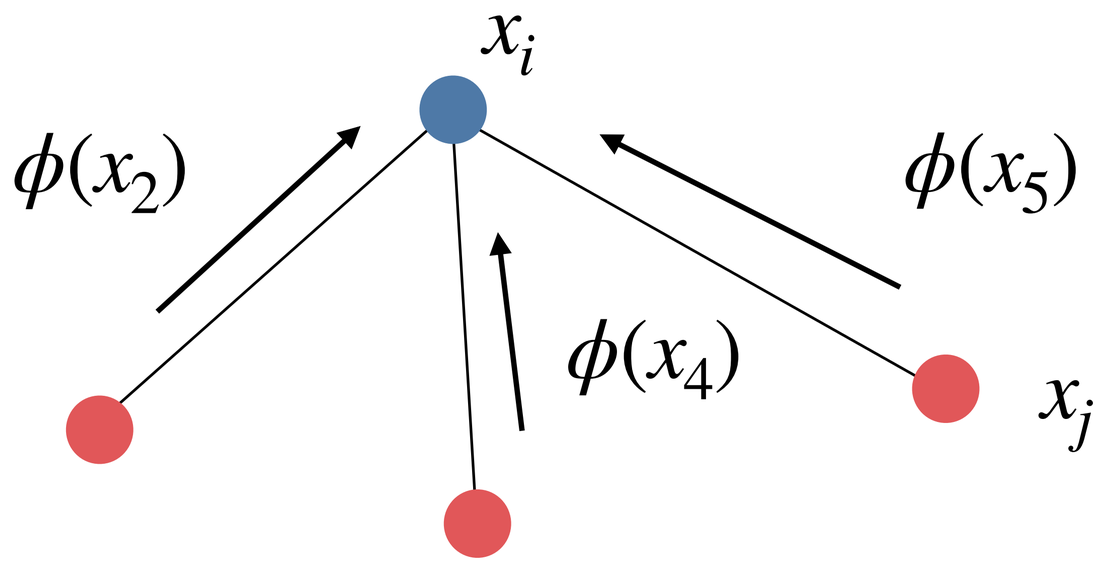
message passing function

simple: Wx_j

more general form:

$\phi(x_j, x_i)$

Aggregate



Aggregate

$$x_{agg} = \sum_{j \in \mathcal{N}(i)} \phi(x_j, x_i) \qquad \text{(sum)}$$

Aggregate

$$x_{agg} = \sum_{j \in \mathcal{N}(i)} \phi(x_j, x_i) \qquad \text{(sum)}$$

$$x_{agg} = \frac{\sum_{j \in \mathcal{N}(i)} \phi(x_j, x_i)}{|\mathcal{N}(i)|} \qquad \text{(mean)}$$

Aggregate

$$x_{agg} = \sum_{j \in \mathcal{N}(i)} \phi(x_j, x_i) \qquad \text{(sum)}$$

$$x_{agg} = \frac{\sum_{j \in \mathcal{N}(i)} \phi(x_j, x_i)}{|\mathcal{N}(i)|} \qquad \text{(mean)}$$

$$x_{agg} = \max_{j \in \mathcal{N}(i)} \phi(x_j, x_i) \qquad \text{(max or min)}$$

Aggregate

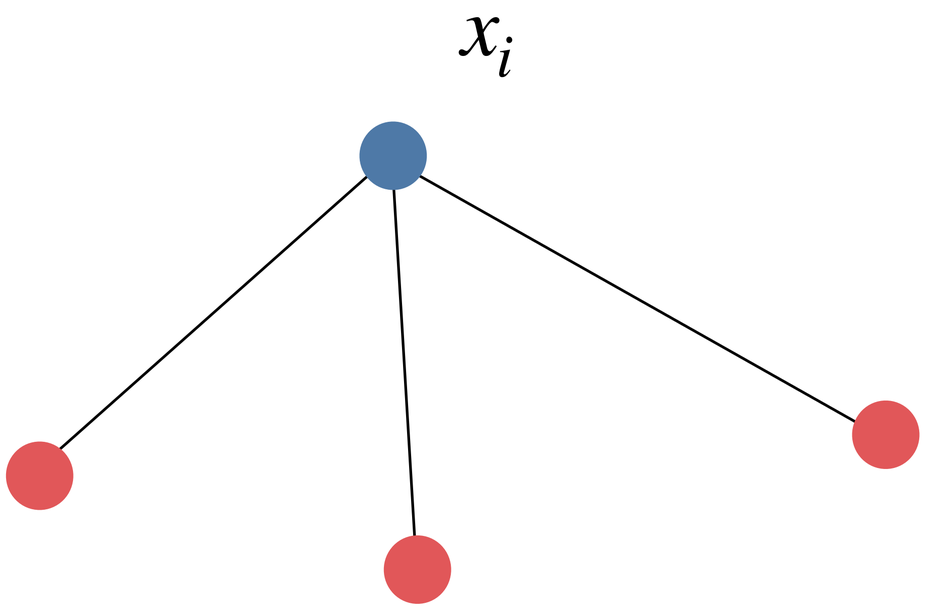
$$x_{agg} = \sum_{j \in \mathcal{N}(i)} \phi(x_j, x_i)$$ (sum)

$$x_{agg} = \frac{\sum_{j \in \mathcal{N}(i)} \phi(x_j, x_i)}{|\mathcal{N}(i)|}$$ (mean)

$$x_{agg} = \max_{j \in \mathcal{N}(i)} \phi(x_j, x_i)$$ (max or min)

$$x_{agg} = \bigoplus_{j \in \mathcal{N}(i)} \phi(x_j, x_i)$$ some other function that is *order invariant*

Update



hidden state
(dimension size may change)

$$x_i^{(k+1)} = \gamma(x_{agg}, x_i^{(k)})$$

some function

All Together

update

$$x_i^{(k+1)} = \gamma(x_i^{(k)}, \bigoplus_{j \in \mathcal{N}(i)} \phi(x_j^{(k)}, x_i^{(k)}))$$

message passing

aggregation

ϕ, γ are learned functions (e.g., Neural Networks)

Simple version

nonlinear activation function

$$x_i^{(k+1)} = \sigma\left(W_{self}^{(k)} x_i^{(k)} + W_{neighbor}^{(k)} \sum_{j \in \mathcal{N}(i)} x_j^{(k)} + b^{(k)} \right)$$

Like a CNN kernel, *for a given layer k*, W and b are same for every node across graph

Graph Convolutional Network (GCN)

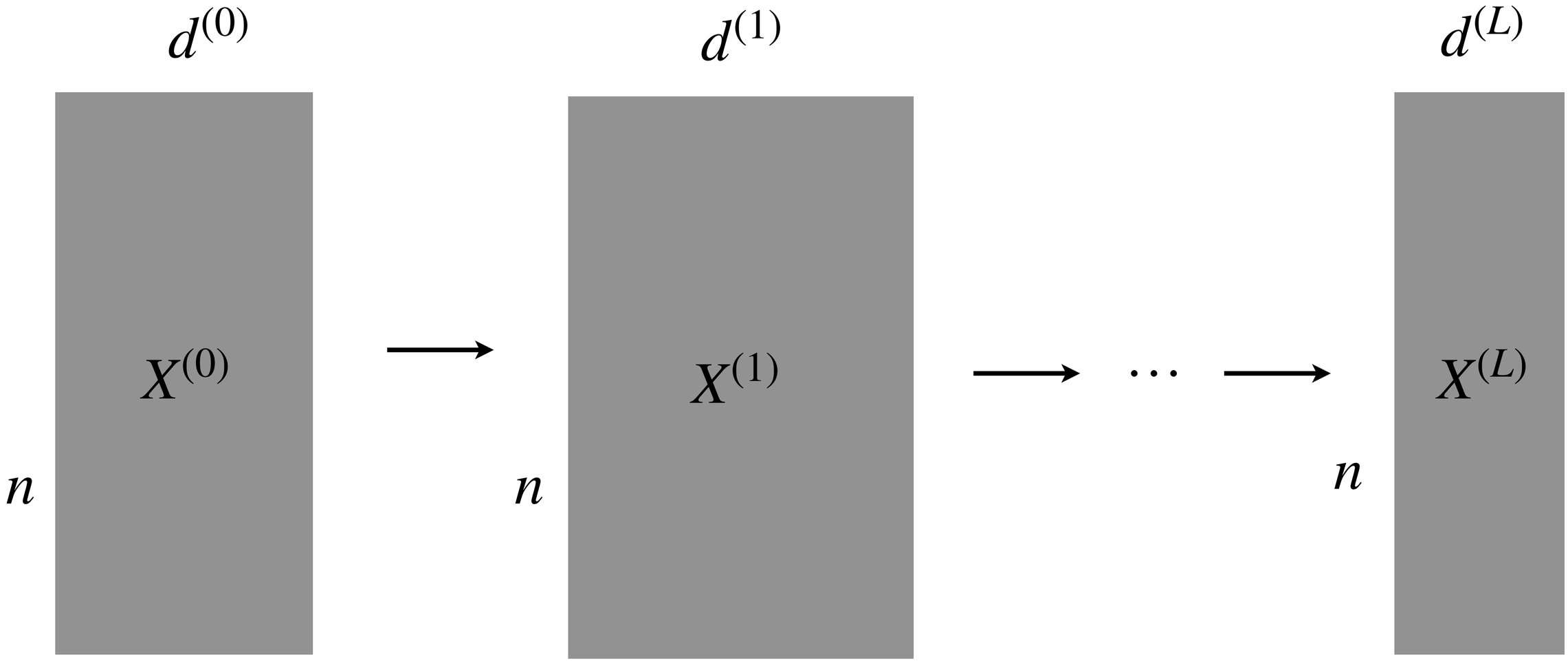
$$x_i^{(k+1)} = \sigma\left(W^{(k)} \sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{x_j^{(k)}}{\sqrt{|\mathcal{N}(i)||\mathcal{N}(j)|}} + b^{(k)} \right)$$

sum over neighbors and self

Matrix form (for some GNNs)

$$Y = \tilde{A}XW$$

Network of Layers

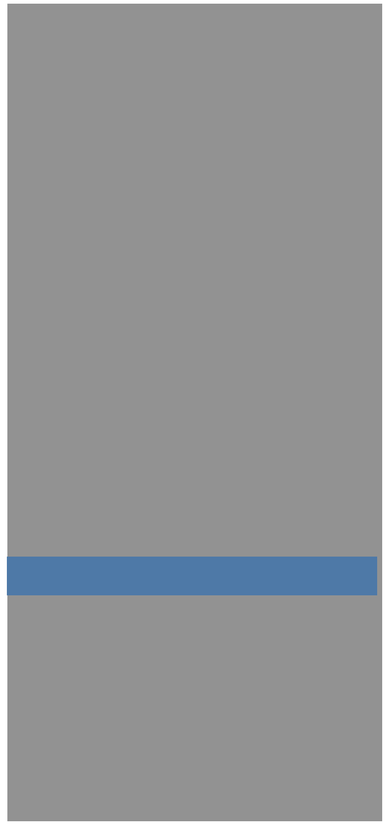


Node, link, and graph classification/regression

$$X^{(L)} = \quad \overset{d^{(L)}}{\underset{n}{\square}}$$

Node, link, and graph classification/regression

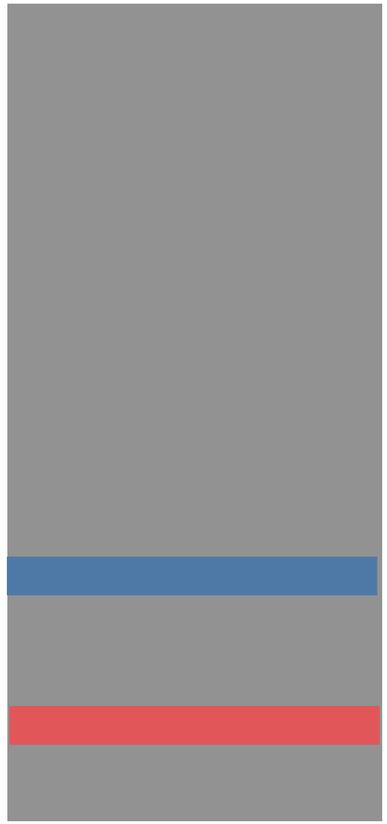
$$X^{(L)} =$$



node

Node, link, and graph classification/regression

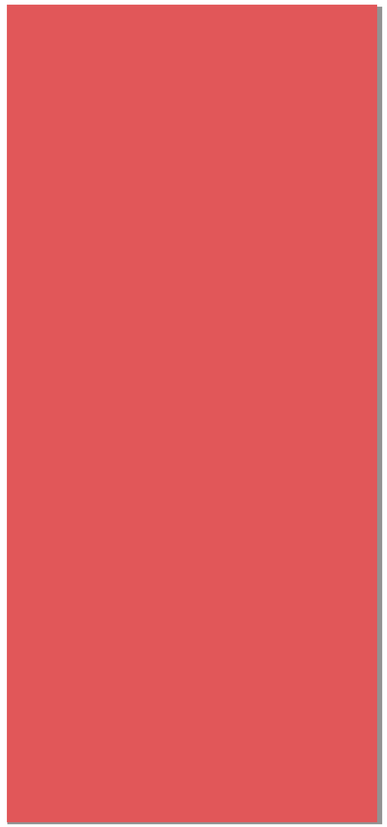
$X^{(L)} =$



edge

Node, link, and graph classification/regression

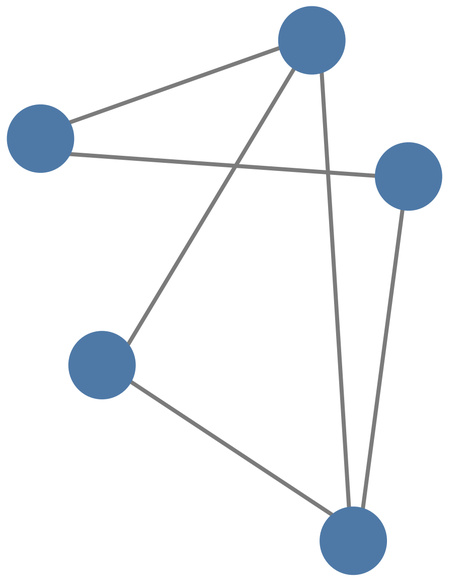
$d^{(L)}$

$X^{(L)} =$  n

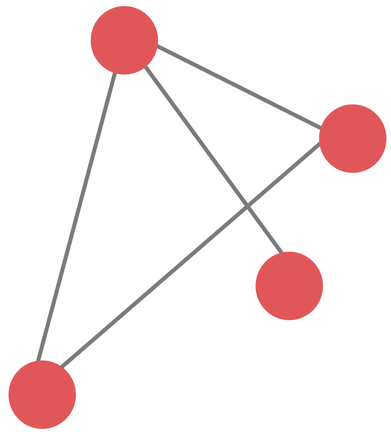
 graph

Testing

inductive: test on graphs not seen in training (basically like what we've done)



train

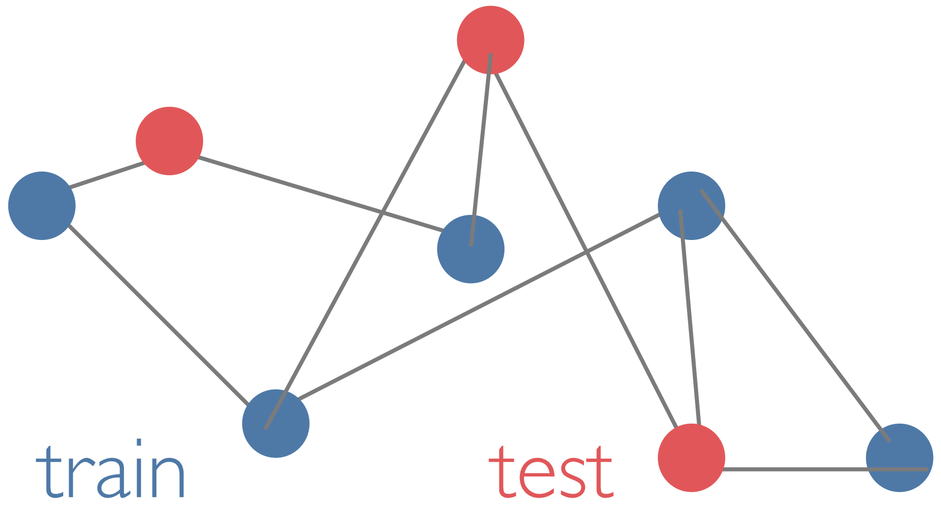


test

Testing

inductive: test on graphs not seen in training (basically like what we've done)

transductive: train and test on same graphs. features of all nodes are visible, but during training labels/targets for some testing nodes are hidden



Graph Attention Networks

$$x_{agg} = \sum_{j \in \mathcal{N}(i)} \alpha_{i,j} W x_j$$