Graph Convolutional Networks

Paula Gómez Duran
paulagomezduran@gmail.com

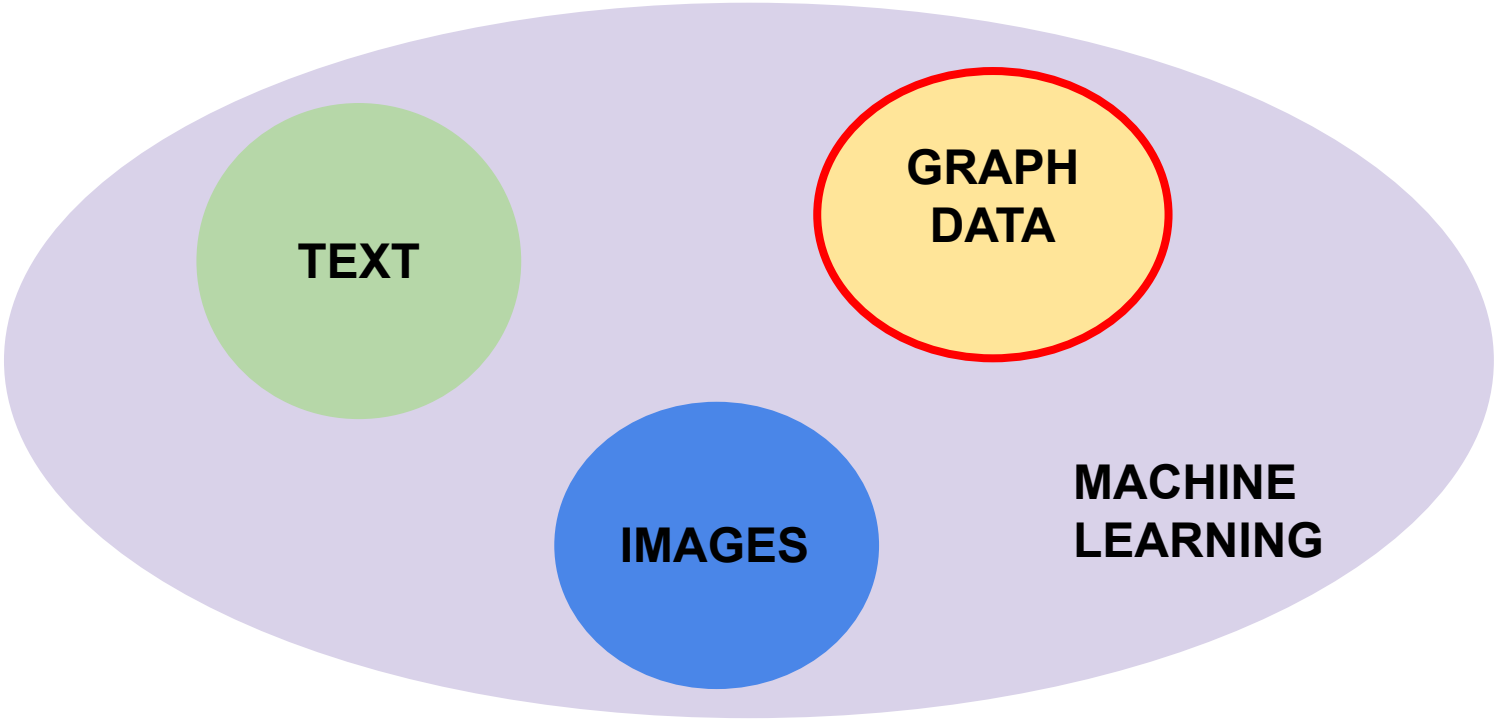
PhD student

UNIVERSITAT DE BARCELONA

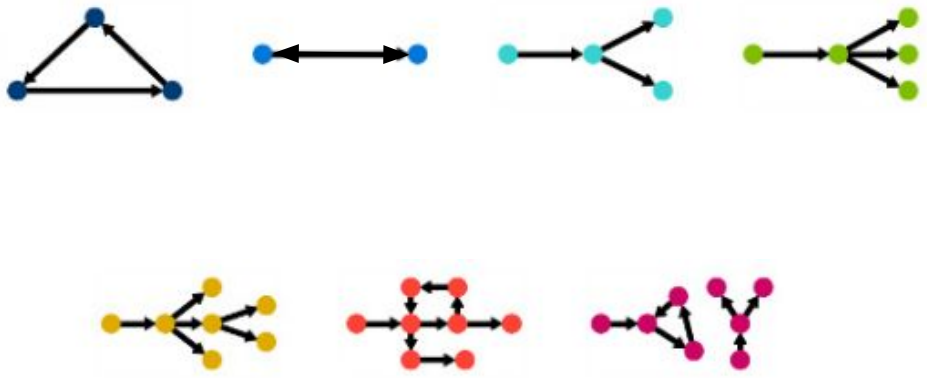
What is a GRAPH ?

A Graph is anything with nodes connected by edges.

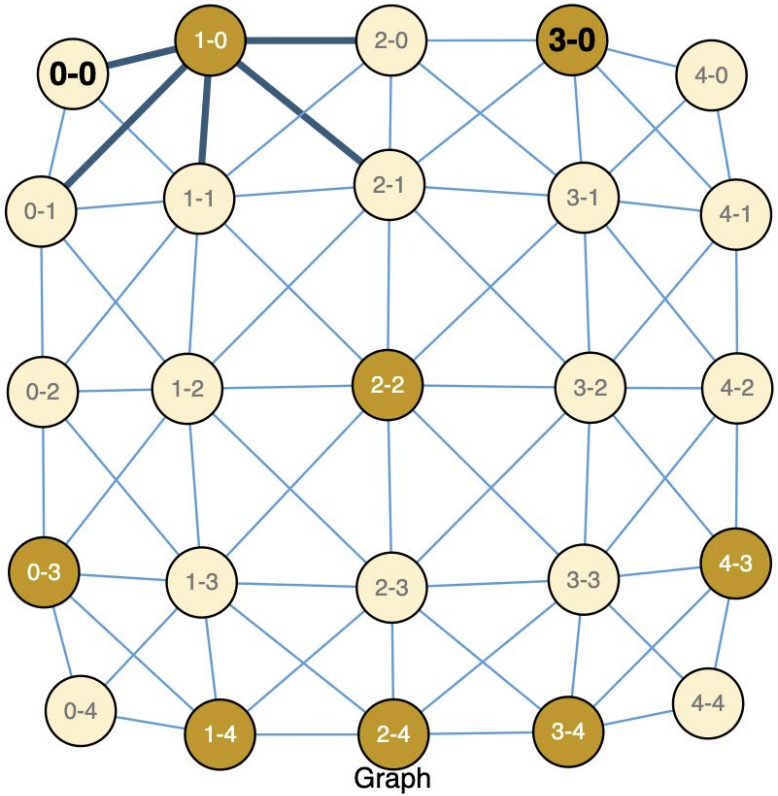
What are Graph Neural Networks?



What are Graph Neural Networks?



Graph Convolutional Networks (GCN)

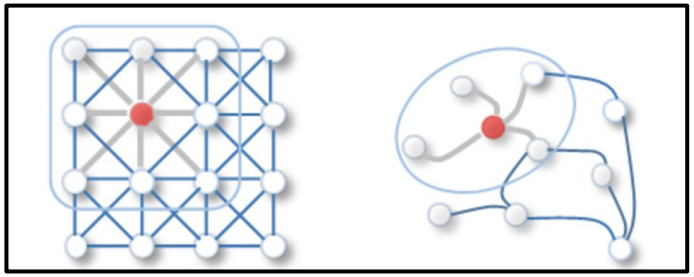


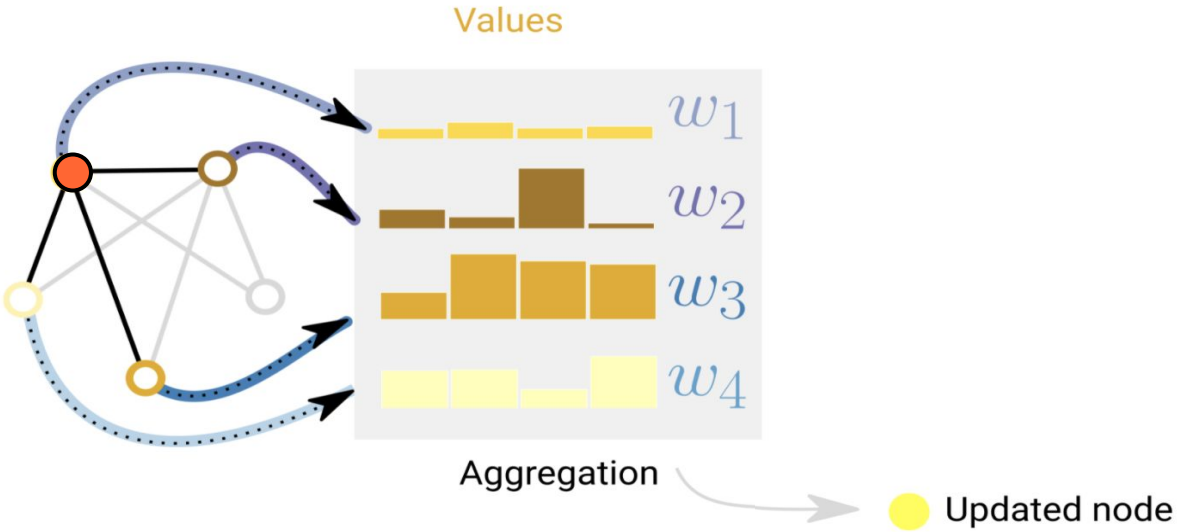
- SAME **GOAL** of the convolution

- PIXELS = RGB values
- LINKS = spatial connections

Figure from : https://distill.pub/2021/gnn-intro/

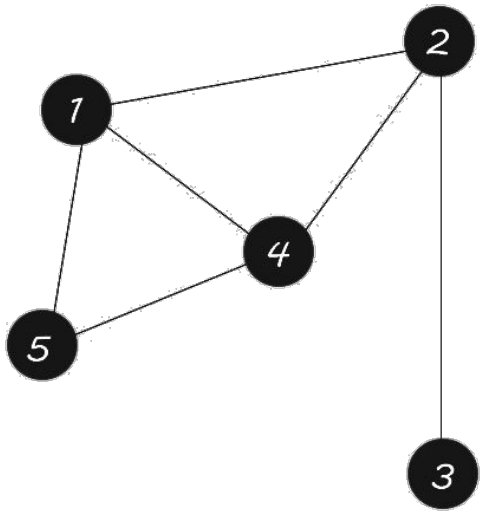
Graph Attention Networks (GAT)





How we define a Graph Convolutional Network?

$$GCN = f(\boldsymbol{X}, \boldsymbol{A})$$

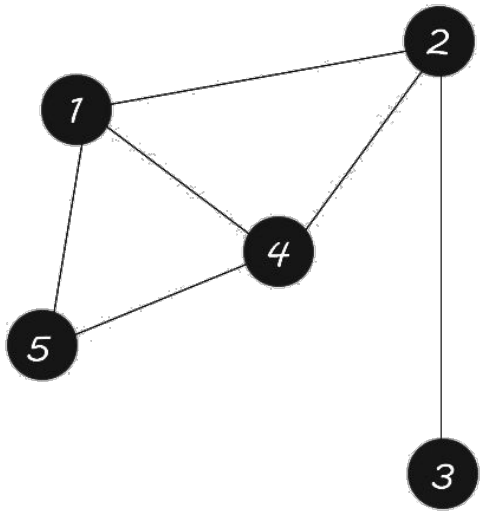


$$A = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 2 & 1 & 0 & 1 & 1 & 0 \\ 3 & 0 & 1 & 0 & 0 & 0 \\ 4 & 1 & 1 & 0 & 0 & 1 \\ 5 & 1 & 0 & 0 & 1 & 0 \end{array}$$

$\boldsymbol{X}$ = *input features*

How we define a Graph Convolutional Network?

$$GCN = f(\boldsymbol{X}, \boldsymbol{A})$$



$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

$$\boldsymbol{X} == \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{bmatrix} 0 & \cdots & 1 \\ 1 & \cdots & 0 \\ 0 & \cdots & 0 \\ 1 & \cdots & 1 \\ 1 & \cdots & 0 \end{bmatrix} \\ \underbrace{\qquad\qquad\qquad} \\ \text{K dimensions} \end{array}$$

GCN equation

Defined in: https://arxiv.org/pdf/1609.02907.pdf

New layer → H(1)

Previous layer → H(0)

$$H^{(l+1)} = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W(l))$$

Weights we want to learn

For the first layer, we do not have previous layer… so, what do you think it will be H(0)?

H(0) == X

GCN equation

D = degree matrix

Left normalization	$\Rightarrow \quad \hat{D}^{-1}\hat{A}$
Symmetric normalization	

$$H^{(l+1)} = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W(l)\right)$$

GCN equation

Symmetric normalization

D = degree matrix

$$H^{(l+1)} = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W(l)\right)$$

$$A = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 2 & 1 & 0 & 1 & 1 & 0 \\ 3 & 0 & 1 & 0 & 0 & 0 \\ 4 & 1 & 1 & 0 & 0 & 1 \\ 5 & 1 & 0 & 0 & 1 & 0 \end{array} + \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \hat{A}$$

GCN equation

Symmetric normalization

D = degree matrix

$$H^{(l+1)} = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W(l)\right)$$

X *= input features*

$\hat{A} =$

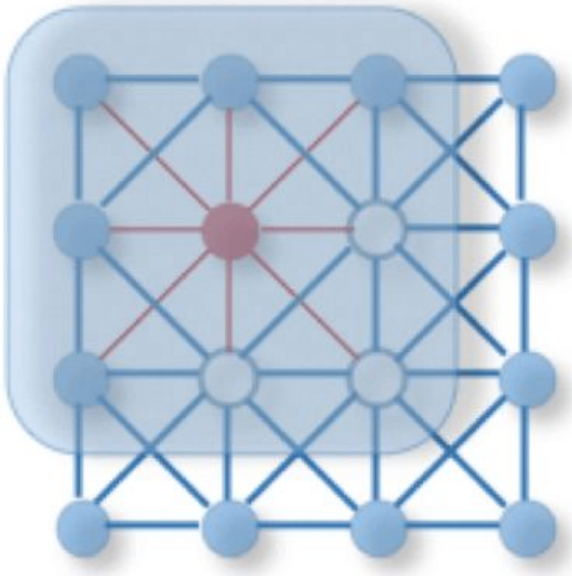
	1	2	3	4	5
1	1	1	0	1	1
2	1	1	1	1	0
3	0	1	1	0	0
4	1	1	0	1	1
5	1	0	0	1	1

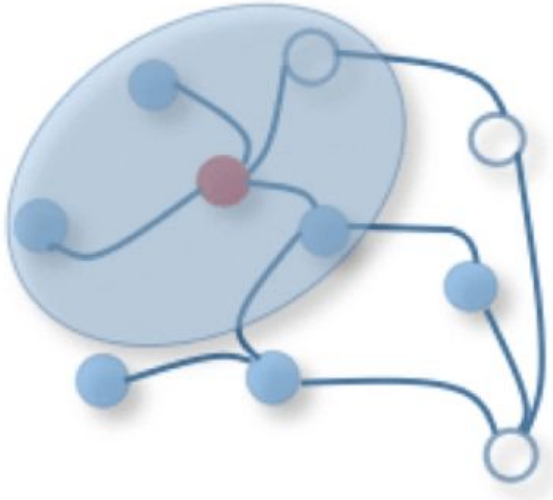
$\hat{D} =$

	1	2	3	4	5
1	4	0	0	0	0
2	0	4	0	0	0
3	0	0	2	0	0
4	0	0	0	4	0
5	0	0	0	0	3

Analogy with Image Convolution

$$H^{(l+1)} = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

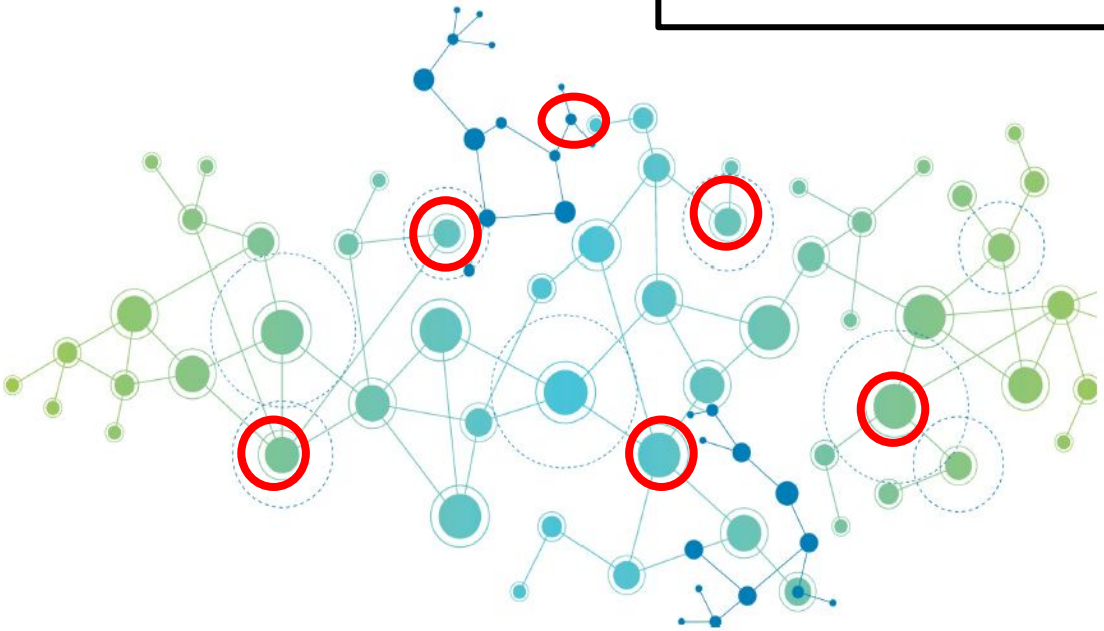




GCN equation defined in: https://arxiv.org/pdf/1609.02907.pdf

GCN: generating predictions

All graph should be fit into the network.



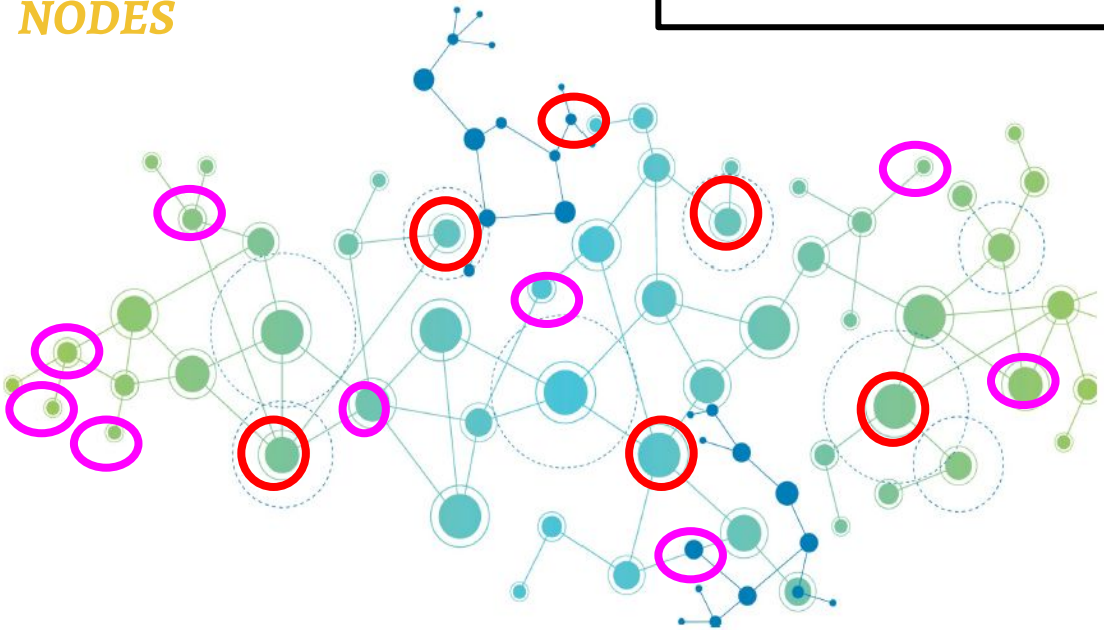
When splitting in train, val and test we are just SELECTING nodes.

TRAINING NODES

GCN: generating predictions

ALL OTHER NODES

All graph should be fit into the network.



When splitting in train, val and test we are just SELECTING nodes.

TRAINING NODES

VALIDATION NODES

TEST NODES

IMPORTANT

We just have ONE GRAPH,
even we have 3 different sets.

Kick off the lab

1. Launch a web browser (Chrome recommended).
2. Login to a Google account. Create a new one if preferred.
3. Two options:
   a. Create a new EMPTY notebook in Google Colab pressing this link.
   b. You can copy the live code from this Jupyter notebook.
4. Change runtime type to work with **GPU**.