IDC410
A course on Image Processing and Machine Learning
(Lecture 20)

Shashikant Dugad,

IISER Mohali

Graphical Neural Network

Slides made using
1. **Lectures given by Petar Velickovic on YouTube:**

https://www.youtube.com/watch?v=uF53xsT7mjc&t=1350s

https://www.youtube.com/watch?v=8owQBFAHw7E&list=PPSV

GNN Lectures adapted from following References

1. Youtube Lectures given by Petar Velickovic on YouTube:

   https://www.youtube.com/watch?v=uF53xsT7mjc&t=1350s

   https://www.youtube.com/watch?v=8owQBFAHw7E&list=PPSV

   https://www.youtube.com/watch?v=uF53xsT7mjc&list=PPSV&t=728s

2. Other Youtube Videos

https://www.youtube.com/watch?v=fOctJB4kVIM&list=PPSV

https://www.youtube.com/watch?v=ABCGCf8cJOE&list=PPSV

https://www.youtube.com/watch?v=0YLZXjMHA-8&list=PPSV

https://www.youtube.com/watch?v=2KRAOZIULzw&list=PPSV

https://www.youtube.com/watch?v=wJQQFUcHO5U&list=PPSV

3. Notes:

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

https://distill.pub/2021/understanding-gnns/

Graph

- A graph represents the relations between a collection of entities called *nodes* and relations between nodes are referred as *edges*.
- Graph contains points (nodes or vertices) with associated *features* and edges (connection between points) with associated *features*
- *Heterogeneous graphs:* Given input collection of nodes and edges, it can be classified into multiple graphs. In a heterogeneous biomedical graph, there might be one type of node representing proteins, one type of node representing drugs, and one type representing diseases

Graph

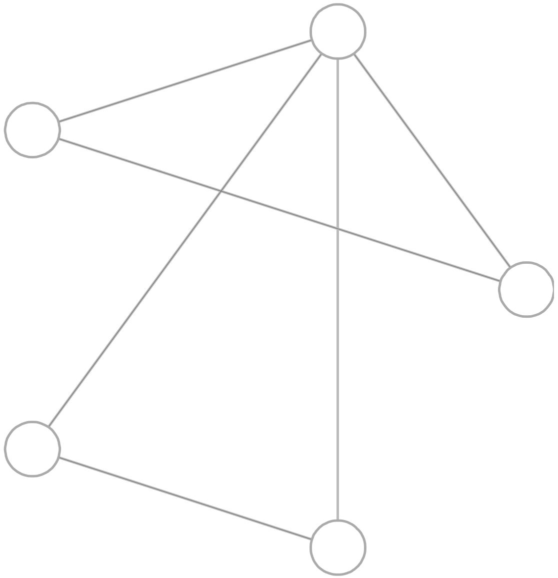
- ***Node Feature:*** Each node may be considered having several features. For example, personal data in a social network, atomic information in a molecules etc. ***Features*** of each node can be represented by a ***vector*** of size ***d***. Where ***d*** represents number of ***node features***. Note that size of node feature vector is same for all nodes. For graph of ***V nodes (vertices)*** the size of ***node feature matrix*** is ***V x d***

- ***Edge Feature:*** Edges can also have features such as energy of a bond, type of a bond etc. ***Features*** of each edge can be represented by a ***vector*** of size ***r***. Where ***r*** represents number of ***edge features***. Note that size of edge feature vector is same for all edges. Size of edge ***feature matrix*** is ***V x r***

Graph

- ***Adjacency Matrix:*** It is a *V x V* matrix indicating presence or absence of edges between nodes. *V* is total number of nodes (vertices) in the graph
- ***Node-Edge degree matrix:*** It's a *V x V* diagonal matrix (all non-diagonal elements are set 0) containing number of edges for each node (diagonal element)
- Mathematically graph can be represented as: ***G(V, E)*** where, *V* and *E* represents nodes and edges as defined above. $V \equiv V(x_i)$, x_i is a feature vector of i^{th} node.

Schematic of Graph

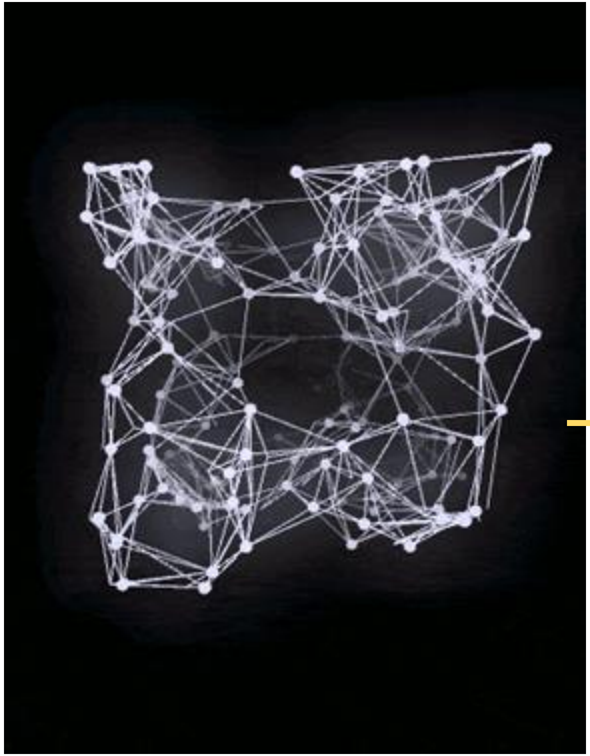


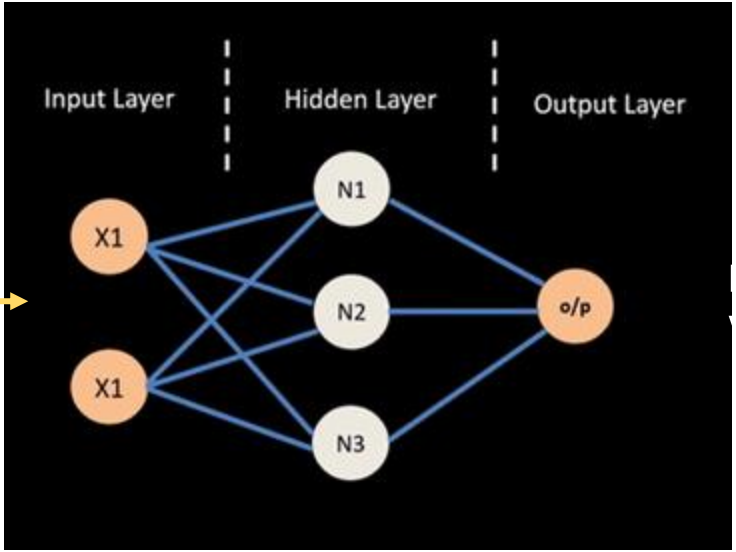
- V: Represents Node Vertex (or node) level attributes
- E: Represents Edge (or link) level attributes with directions
- G: Represents Global (or master node) level attributes

<u>For a Social Group Interaction Graph:</u>

- Attributes (features) of *V* can be Name, Age, Gender, Hobbies etc 
- Attributes (features) of *E* can be friend, colleague, boss etc. 
- Attributes (features) of *G* can be behavior of the group, such as polite, hostile etc. 

GRAPH NEURAL NETWORK



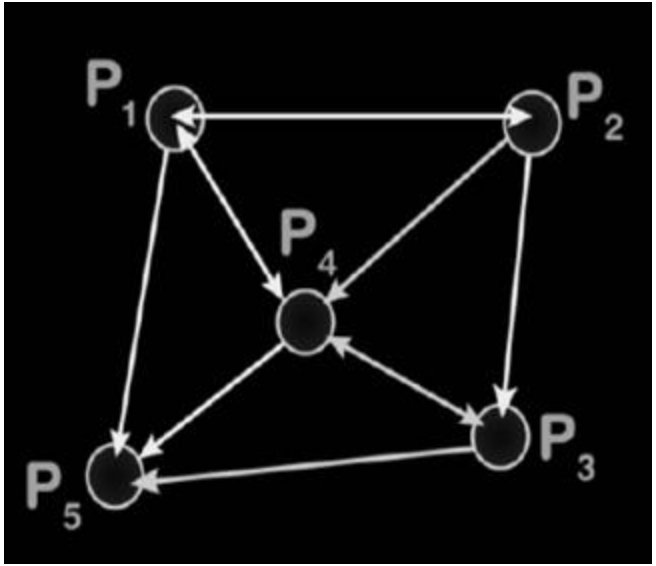


Adjacency and Degree Matrix

- The ***adjacency, degree and feature*** matrices are input to the ***Graphical Convolutional Network (GCN)***

$$\text{The adjacency matrix} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Degree\ Matrix = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$



Feature Matrix

- The feature matrix contains all the feature data *(columns)* of each node *(rows)*. Table below contains 4 features and 5 nodes

X	Y	Layer	Eff_ADC
73.0367	20.1063	1	152.1
54.8556	57.2024	2	174.07
73.0367	19.3055	3	179.14
82.7457	11.2976	4	518.83
74.4237	19.3055	5	136.89

Properties of Graph: Permutation Matrix

- Permutation and and permutation of matrices
- It is desirable to have same behavior of graph irrespective of sequence in nodes are ordered in the feature matrix. We have ***V!*** permutations for a given graph having ***V*** nodes
  - Permutation (3,1,4,2) ➔ $y_1 \rightarrow x_3$, $y_2 \rightarrow x_1$, $y_3 \rightarrow x_4$, $y_1 \rightarrow x_2$
  - The permutation of nodes in the feature matrix can be accomplished by applying appropriate permutation matrix ***(PM)*** on the feature matrix as shown below
  - The PM has ***ONLY*** one element to be ***1*** in each row and others are ***0***
  - Summation of each column is ***ONE***

$$\mathbf{P}_{(2,4,1,3)}\mathbf{X} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} — & \mathbf{x}_1 & — \\ — & \mathbf{x}_2 & — \\ — & \mathbf{x}_3 & — \\ — & \mathbf{x}_4 & — \end{bmatrix} = \begin{bmatrix} — & \mathbf{x}_2 & — \\ — & \mathbf{x}_4 & — \\ — & \mathbf{x}_1 & — \\ — & \mathbf{x}_3 & — \end{bmatrix}$$

Properties of Graph: Permutation Invariance

- When a function ***f*** is applied on a graph, its result should not depend on the sequence in which nodes are ordered in the feature matrix
- It means the function ***f*** is applied on the feature matrix ***X***, should be invariant under permutation. This is referred as permutational invariance as shown below

$$f(PX) = f(X)$$

- One of the example of such funct

$$f(\mathbf{X}) = \phi\left(\sum_{i \in \mathcal{V}} \psi\left(\mathbf{x}_i\right)\right)$$

- where, Ψ and φ are learnable functions. Aggregator could be sum or mean or max etc.

Properties of Graph: Permutation Equivariance

- Invariance of permutation is required for graph behavior (graph classification)
- However, what if, we are looking for node level or edge level predictions then,
  - Permutation invariant aggregator would destroy such possibilities of node or edge level predictions
- We need to choose the function that will not change the node order i.e. result should be same if we apply the permutation before $(f(PX))$ or after application of the function $(Pf(X))$. Such property of the function is referred as, permutation equ-variance i.e.,

$$f(PX) = Pf(X)$$

Permutation Invariance and Equivariance

- Applying function simultaneously on ***node*** and ***edge*** (established in ***adjacency matrix A***) is a general purpose approach
- Applying permutation on ***adjacency matrix A*** requires permutation of both ***rows and columns***
  - This results into a mathematical operation PAP^T
- Therefore, operation of invariance and equivariance on nodes and edges can be represented as:

Invariance: $f(PX, PAP^T) = f(X, A)$

Equivariance: $f(PX, PAP^T) = Pf(X, A)$

Learning on graphs

- Now we augment the set of nodes with **edges** between them.
  - That is, we consider general $E \subseteq V \times V$.

- We can represent these edges with an **adjacency matrix**, **A**, such that:

$$a_{ij} = \begin{cases} 1 & (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

- Further additions (e.g. *edge features*) are possible but **ignored** for simplicity.

- Our main desiderata (*permutation {in,equi}variance*) still hold!

Convolution Function Construct for Graphs

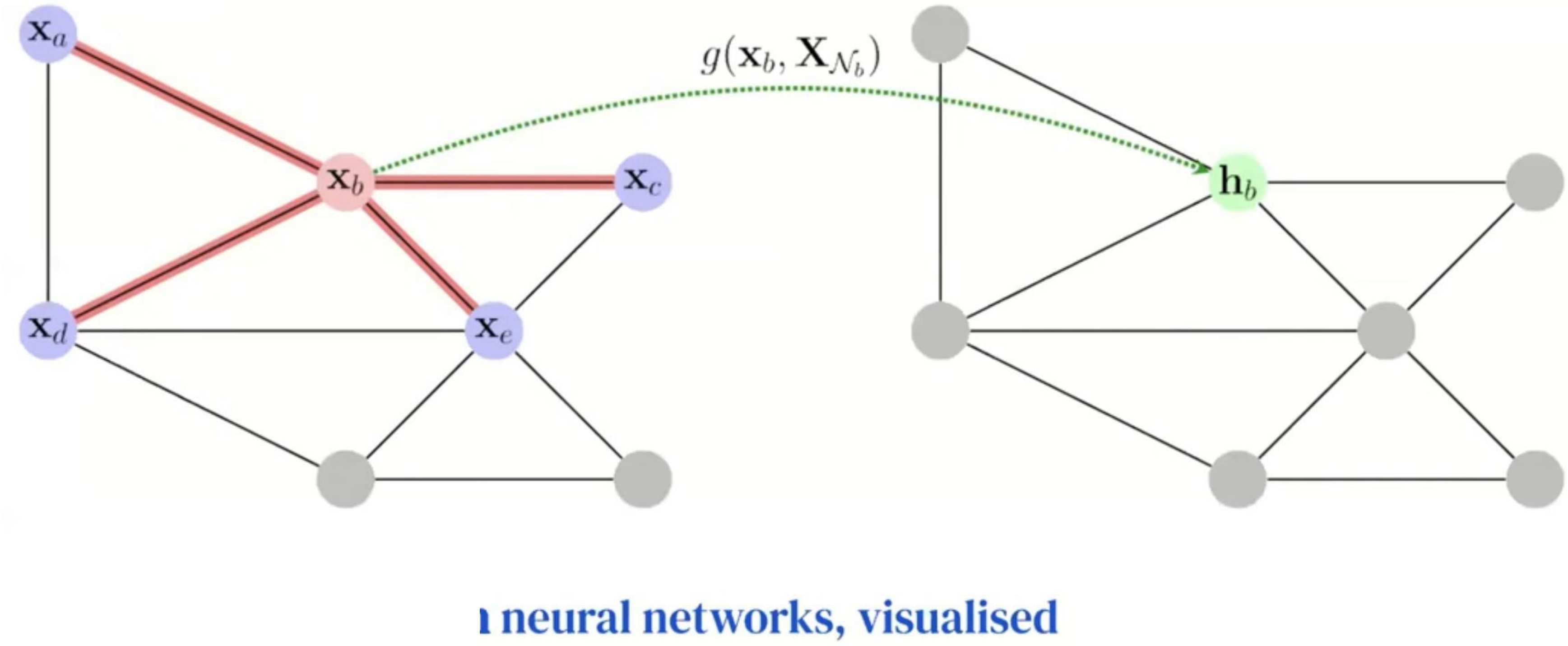
- Permutation equivariant function ***f(X,A)*** can be constructed by applying ***local function g*** over all neighborhoods as shown:

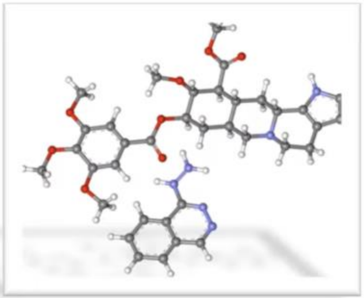
ph neural networks

:ruct permutation equivariant functions, f(**X**, **A**), by app
g, over *all* neighbourhoods:

- To ensure equivariance of ***f(X,A)***, the function ***g*** should not depend on the order nodes in X_{Ni}
  - Hence ***g*** should be ***permutation invariant***

Schematic of Graph Convolution





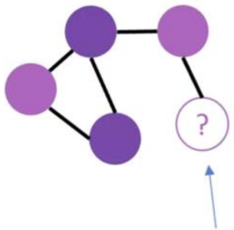
Medicine / Pharmacy

Recommender Systems

Social Networks

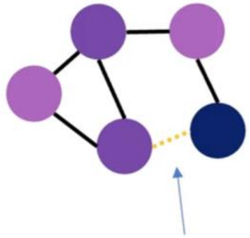
3D Games / Meshes

Node-level predictions



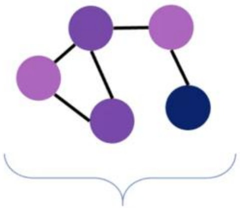
Does this person smoke?
(unlabeled node)

Edge-level predictions
(Link prediction)



Next Netflix video?

Graph-level predictions



Is this molecule a suitable drug?