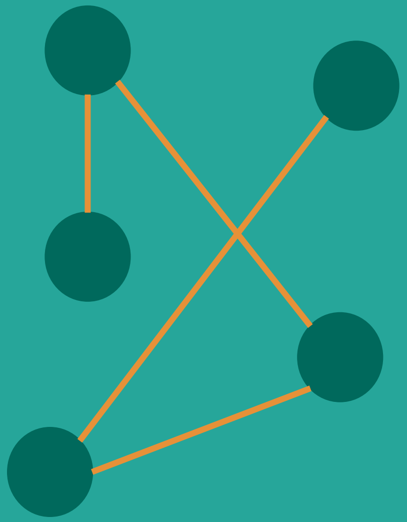
Graph Neural Networks

Inspired from my University Thesis and UPenn ESE 5140

Hetav Pandya (HP)

Watch out for the KitKat questions!

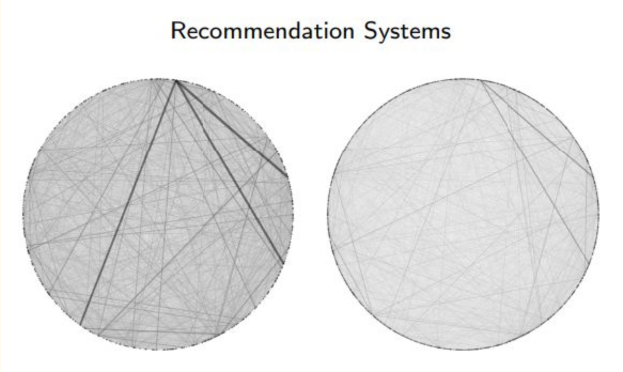
Let's talk about Graph Networks!



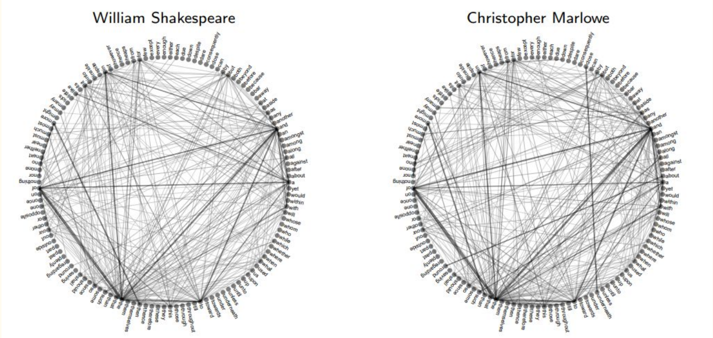
Explicit and Implicit cases

Social media networks, ASIC design problems, devices on a network…

But you don't need an explicit graphical structure to exploit graphical algorithms and techniques!



Authorship Attribution

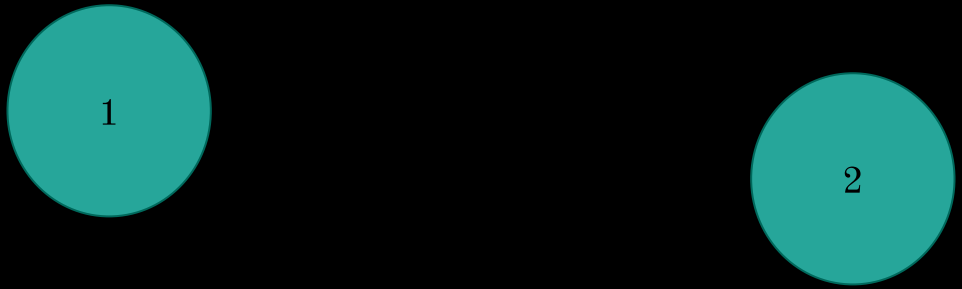


[1] J. Bump, Graph Neural Networks, https://gnn.seas.upenn.edu/ (accessed Aug. 18, 2024).

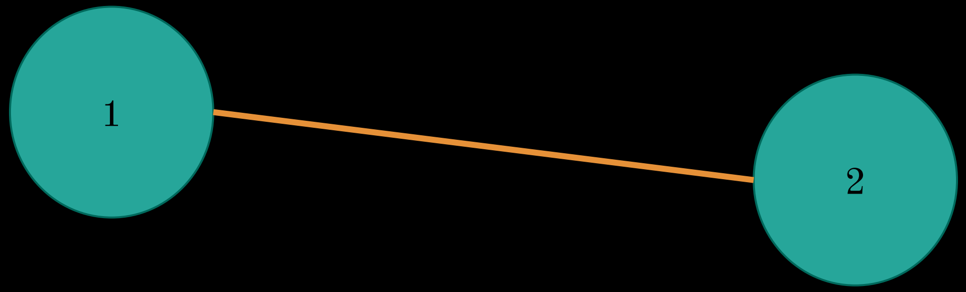
Some business use cases

- Document Classification and Information Extraction
- Personalized Content Delivery
- Telecommunications Network Management
- Mapping spread of infectious diseases
- Determining political bias in a section of population
- Detecting anomalies and potential threats in network traffic
- Molecular property prediction in drugs

The basics

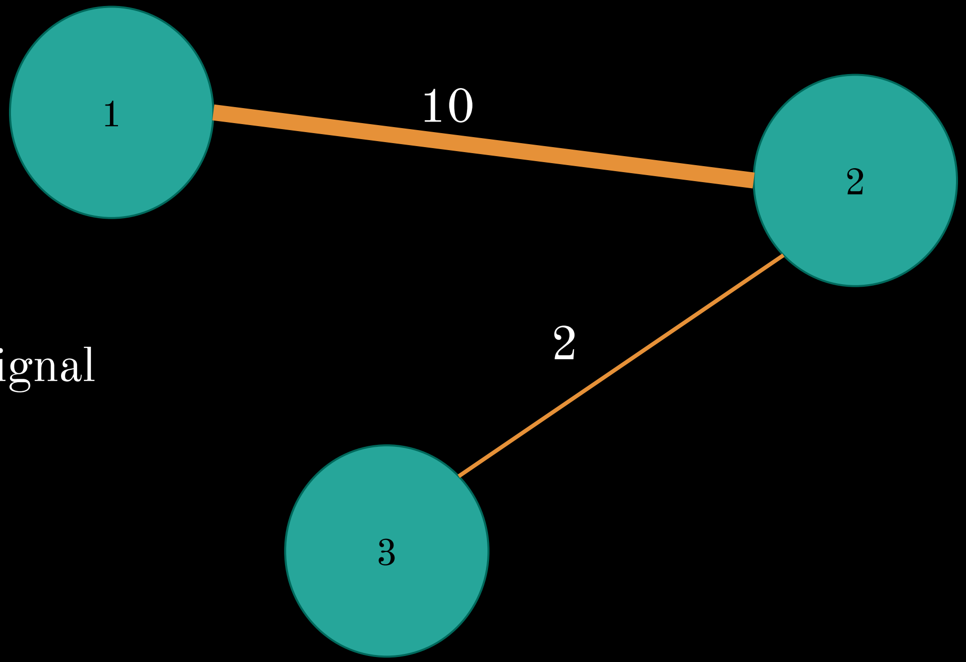


Nodes

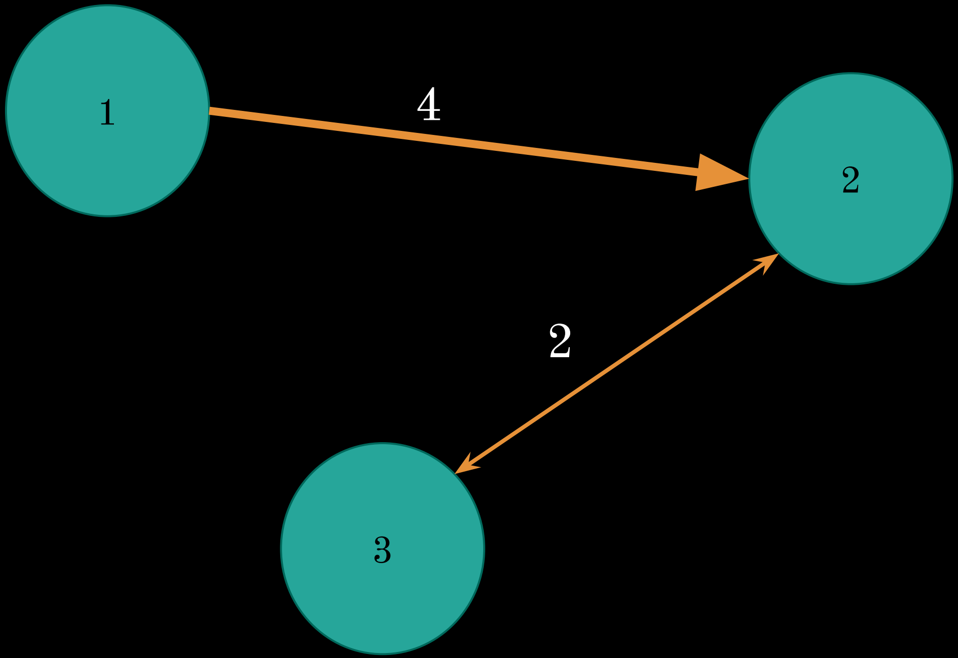


Edges

Weights / Signal Values



Direction

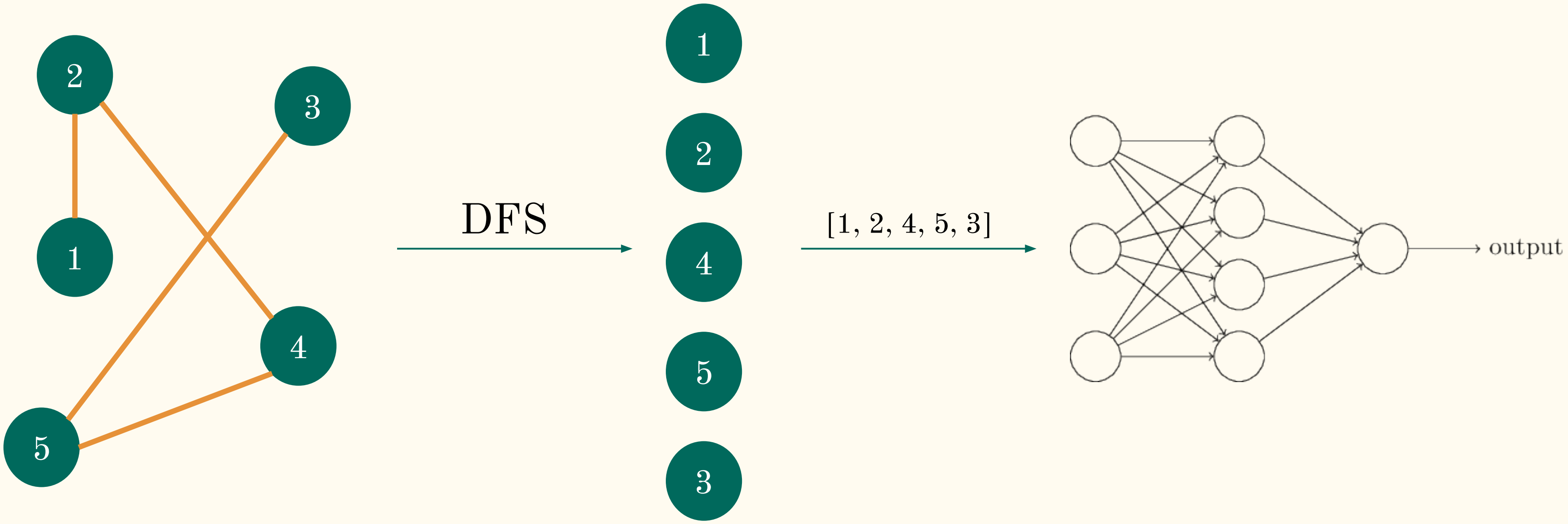


First iteration

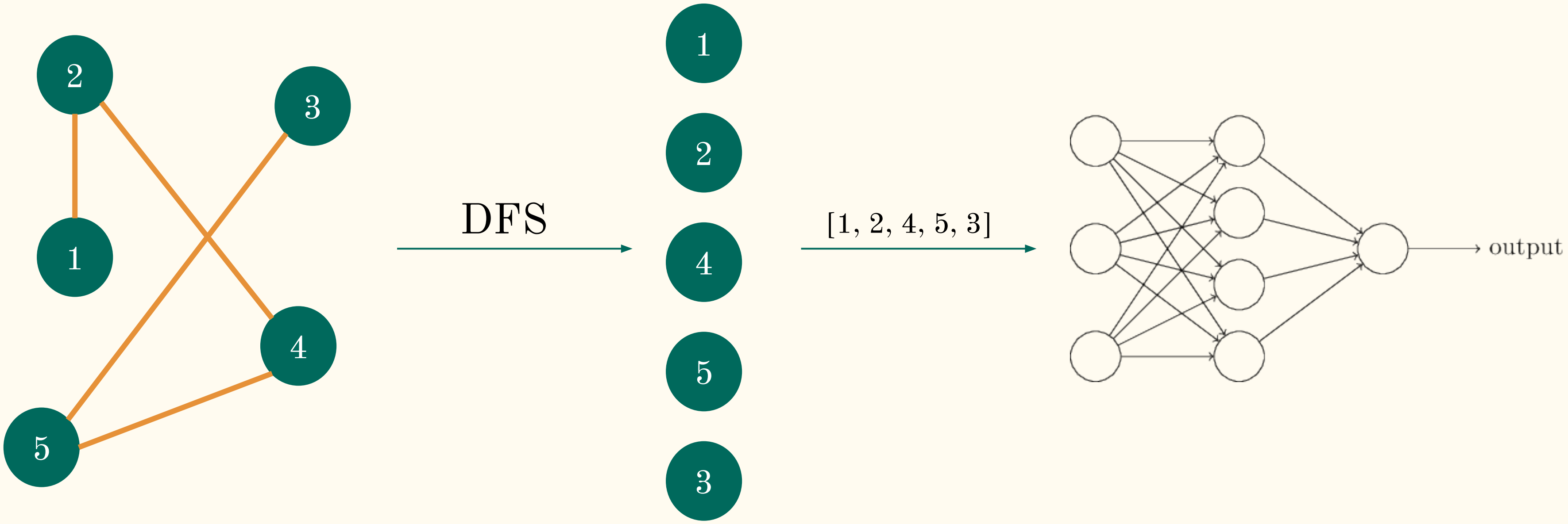
Let's logically build up a GNN

- What do we have?
- A Graph
- Concept of neural network

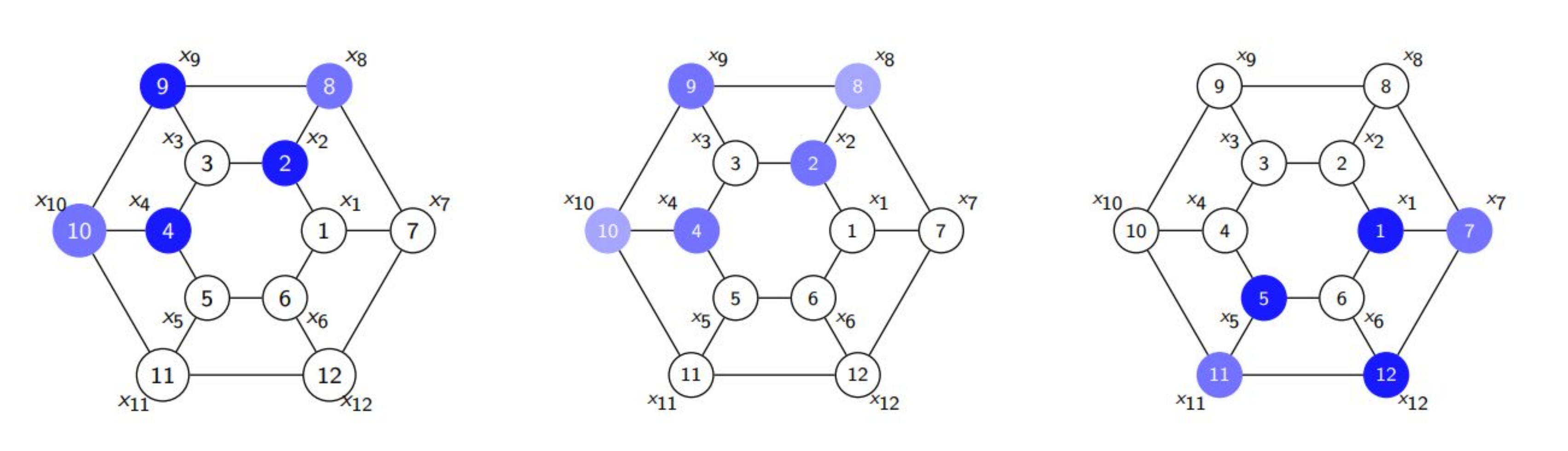
Iteration 1:



Iteration 1: What are we missing?



Iteration 1: What are we missing?



Training Data

Test Data #1

Test Data #2

[1] J. Bump, Graph Neural Networks, https://gnn.seas.upenn.edu/ (accessed Aug. 18, 2024).

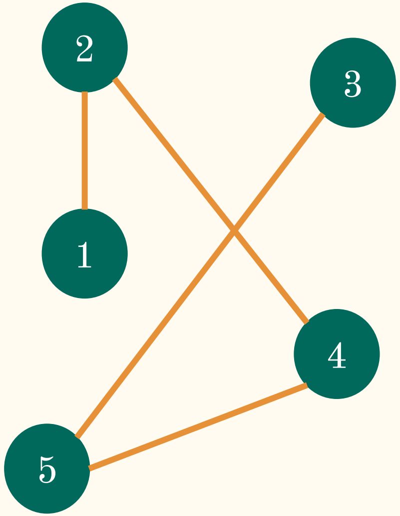
Second iteration

Let’s logically build up a GNN

- What do we have?
- A Graph
- Concept of neural network
- Need to encode the graph relationships

Let’s introduce the concept of Graph Shift Operators

Adjacency Matrix: S



0	1	0	0	0
1	0	0	1	0
0	0	0	0	1
0	1	0	0	1
0	0	1	1	0

Types of Graph Shift Operators

Adjacency Matrix

Does not have degree info

Normalized Adjacency Matrix

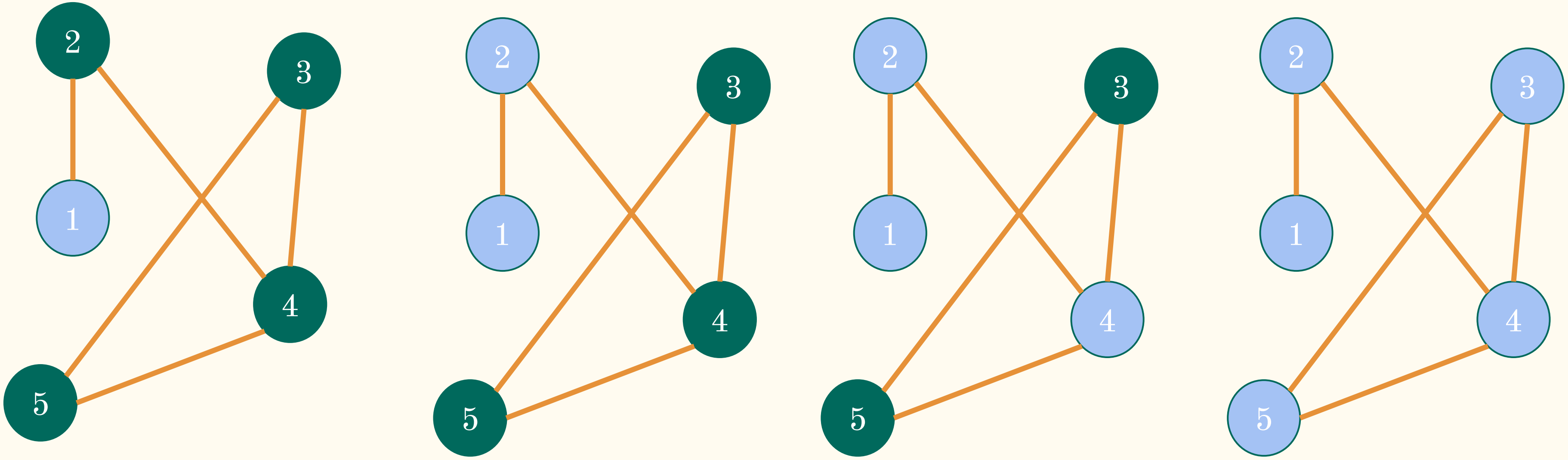
Laplacian Matrix

Has degree info

Normalized Laplacian Matrix

Convolution with Graph Shift Operator

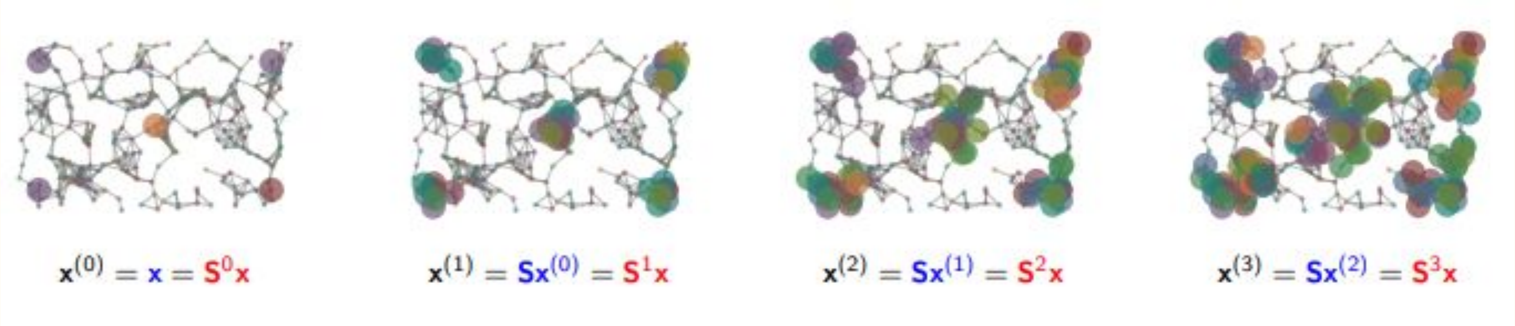
$$\sum_{k=0}^{\infty} h_k \mathbf{S}^k \mathbf{x}$$



$h_0 \mathbf{S}^0 \mathbf{x}$ $h_1 \mathbf{S}^1 \mathbf{x}$ $h_2 \mathbf{S}^2 \mathbf{x}$ $h_3 \mathbf{S}^3 \mathbf{x}$

Convolution with Graph Shift Operator

Real-life example



Notice that we started with 5 nodes and the shift operator gives us information about graph locality and connections

[1] J. Bump, Graph Neural Networks, https://gnn.seas.upenn.edu/ (accessed Aug. 18, 2024).

But we are still missing a very important piece!

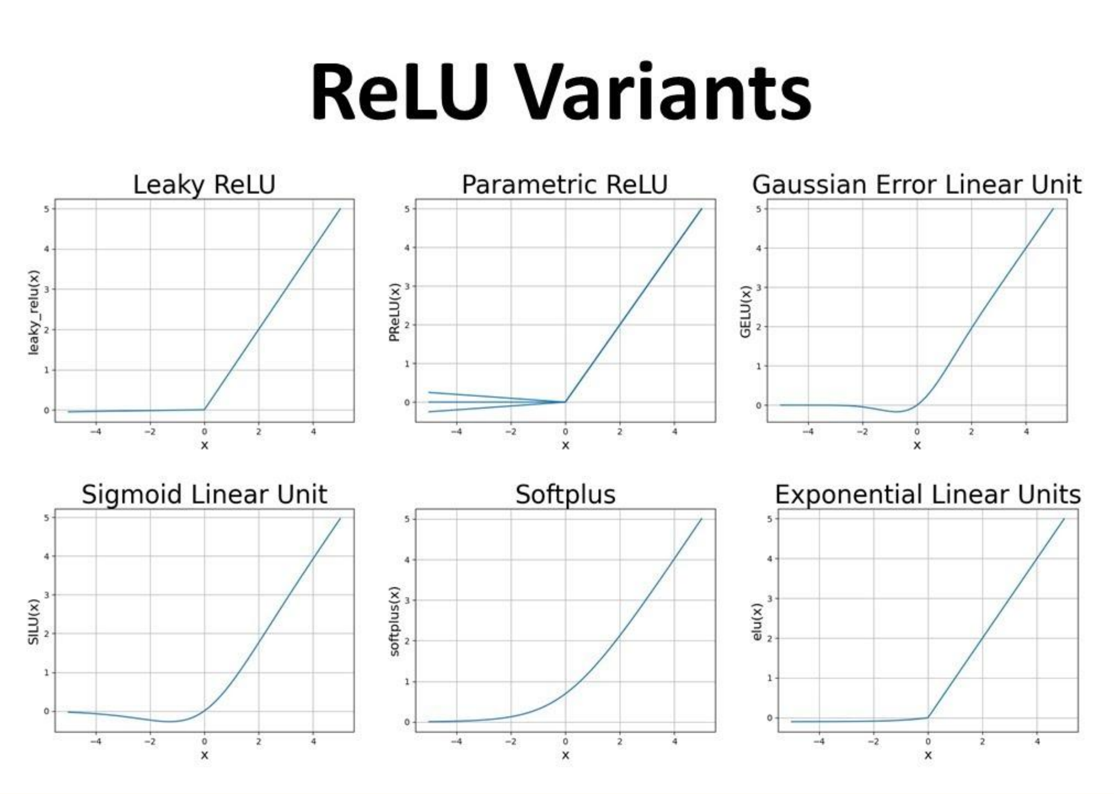
Hint: We are relying on a “linear” combination of matrix S multiplications.

Third iteration

Let's logically build up a GNN

- What do we have?
- A Graph
- Concept of neural network
- Graph Shift Operator
- Need for non-linearity

What are non-linearities?



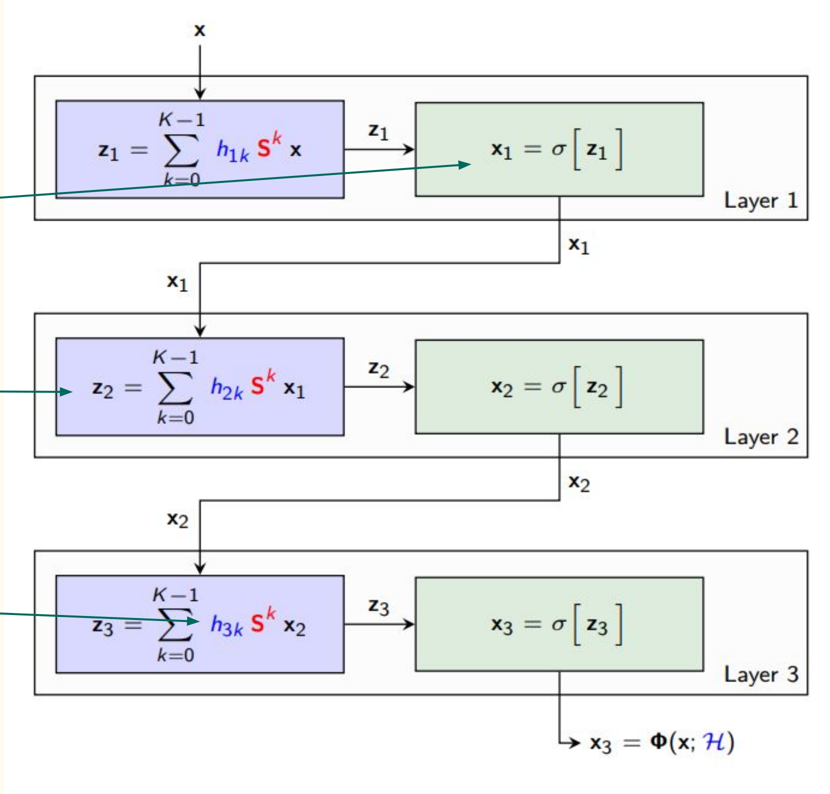
Nonlinearity implies that the relationship between input and output is more complex and cannot be expressed as a simple linear function.

How does it all come together!

Non-linearity

Convolutional filters

Trainable parameters (h)



[1] J. Bump, Graph Neural Networks, https://gnn.seas.upenn.edu/ (accessed Aug. 18, 2024).

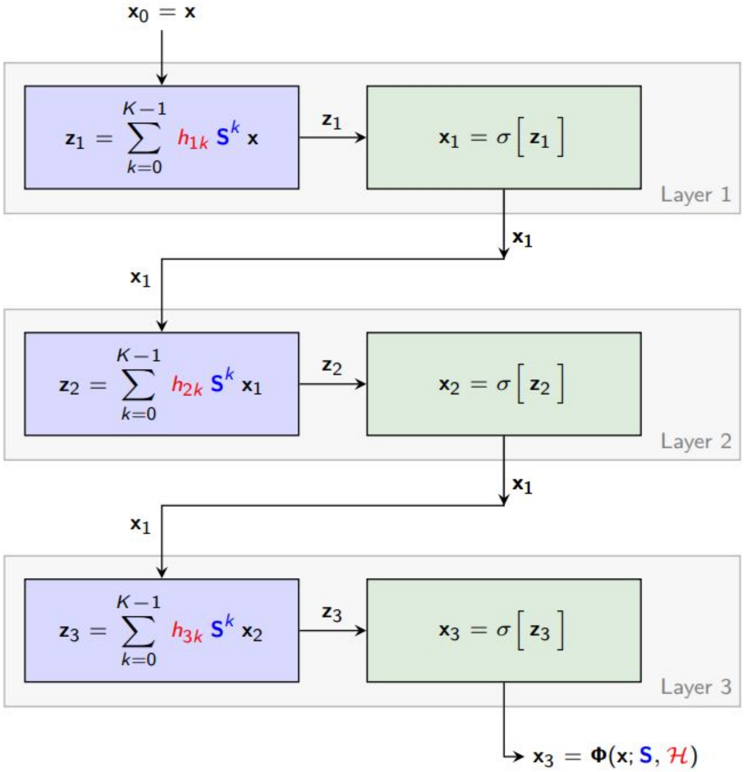
The process of training...

- Learn Optimal GNN tensor $\mathcal{H}^* = (\mathbf{h}_1^*, \mathbf{h}_2^*, \mathbf{h}_3^*)$ as

$$\mathcal{H}^* = \underset{\mathcal{H}}{\text{argmin}} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{T}} \ell\Big(\Phi(\mathbf{x};\mathbf{S},\mathcal{H}),\mathbf{y}\Big)$$

- Optimization is over tensor only. Graph $\mathbf{S}$ is given

$\Rightarrow$ Prior information given to the GNN



[1] J. Bump, Graph Neural Networks, https://gnn.seas.upenn.edu/ (accessed Aug. 18, 2024).

Demo Time!

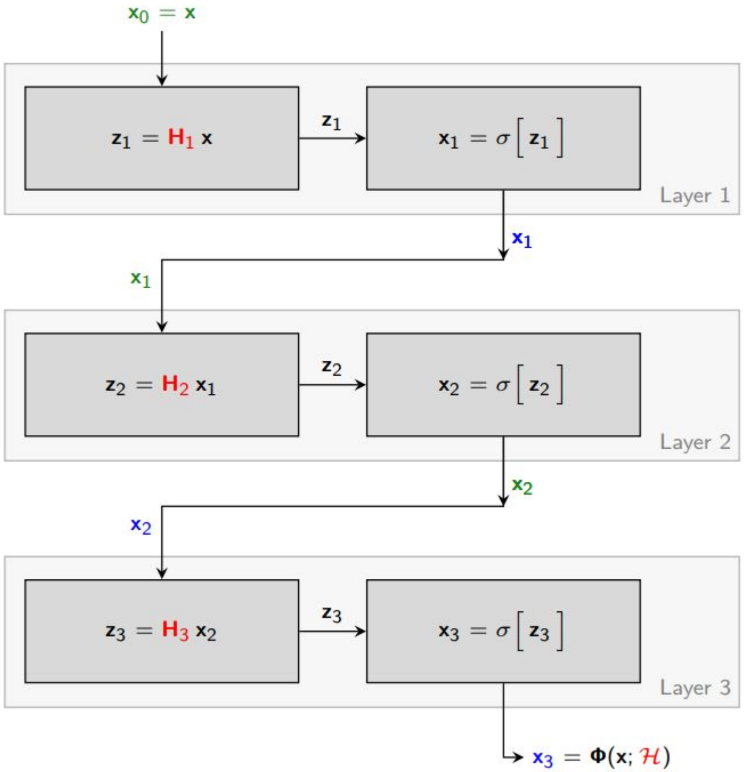
If time permits...

A brief discussion about FCNN

- Illustrate definition with an FCNN with 3 layers
- Feed input signal $\mathbf{x} = \mathbf{x}_0$ into Layer 1

$$\mathbf{x}_1 = \sigma\left[\mathbf{z}_1\right] = \sigma\left[\mathbf{H}_{1k}\,\mathbf{x}_0\right]$$

- Output $\Phi(\mathbf{x}; \mathcal{H})$ Parametrized by $\mathcal{H} = [\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3]$



[1] J. Bump, Graph Neural Networks, https://gnn.seas.upenn.edu/ (accessed Aug. 18, 2024).

FCNN vs GNN

Which one would theoretically perform better?

- Since the GNN is a particular case of a fully connected NN, the latter attains a smaller cost

$$\min_{\mathcal{H}} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{T}} \ell\Big(\mathbf{\Phi}(\mathbf{x};\mathcal{H}),\mathbf{y}\Big) \leq \min_{\mathcal{H}} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{T}} \ell\Big(\mathbf{\Phi}(\mathbf{x};\mathbf{S},\mathcal{H}),\mathbf{y}\Big)$$

- The fully connected NN does better. But this holds for the training set

- In practice, the GNN does better because it generalizes better to unseen signals

  $\Rightarrow$ Because it exploits internal symmetries of graph signals codified in the graph shift operator

[1] J. Bump, Graph Neural Networks, https://gnn.seas.upenn.edu/ (accessed Aug. 18, 2024).

Laplacian Shift Operators

Laplacian Matrix

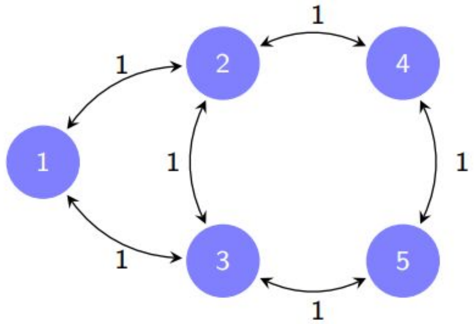
- The Laplacian matrix of a graph with adjacency matrix $\mathbf{A}$ is $\Rightarrow \mathbf{L} = \mathbf{D} - \mathbf{A} = \text{diag}(\mathbf{A1}) - \mathbf{A}$

- Can also be written explicitly in terms of graph weights $A_{ij} = w_{ij}$

  $\Rightarrow$ Off diagonal entries $\Rightarrow L_{ij} = -A_{ij} = -w_{ij}$

  $\Rightarrow$ Diagonal entries $\Rightarrow L_{ii} = d_i = \sum_{j \in n(i)} w_{ij}$

$$\mathbf{L} = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 \\ -1 & -1 & 3 & 0 & -1 \\ 0 & -1 & 0 & 2 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$



[1] J. Bump, Graph Neural Networks, https://gnn.seas.upenn.edu/ (accessed Aug. 18, 2024).

Thank you!

Let’s keep connected!

Hetav Pandya

Software Engineer @ Arista Networks | University of Toronto Alumnus | Compute...

Feedback!