

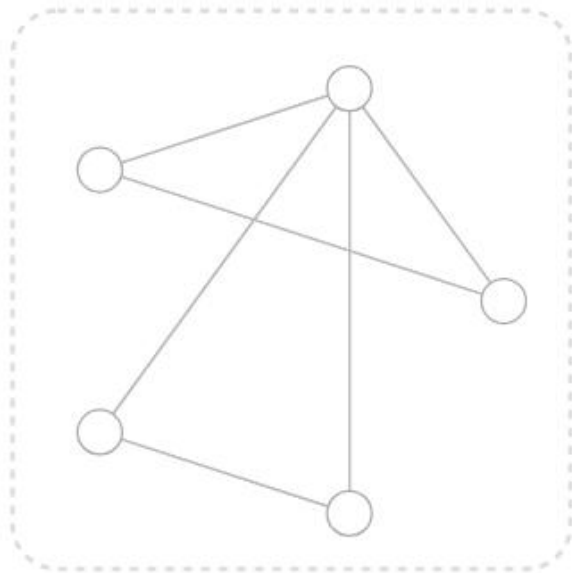
Graph neural network (GNN)

Jingnan

Step-by-step explanation

- What is graph
- What is the application of graph
- How to represent/save a graph
- What is graph neural network
- What is graph convolution network
- What is graph attention network
- What is the tips/experience to train a graph neural network
- ... (if we have time)

What is graph?



- V** Vertex (or node) attributes
e.g., node identity, number of neighbors
- E** Edge (or link) attributes and directions
e.g., edge identity, edge weight
- U** Global (or master node) attributes
e.g., number of nodes, longest path

Undirected edge



Directed edge



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

Examples of graph

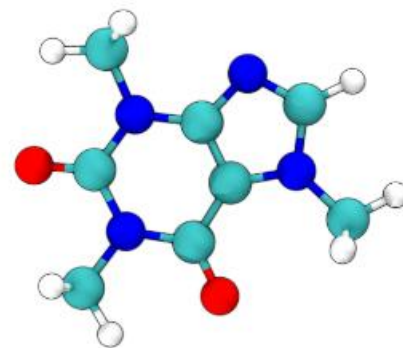
0-0	1-0	2-0	3-0	4-0
0-1	1-1	2-1	3-1	4-1
0-2	1-2	2-2	3-2	4-2
0-3	1-3	2-3	3-3	4-3
0-4	1-4	2-4	3-4	4-4

Image Pixels

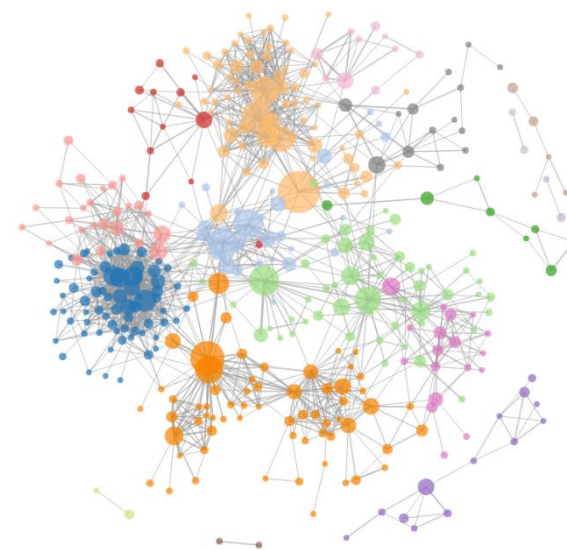
Images as graphs



Text as graphs



Molecules as graphs

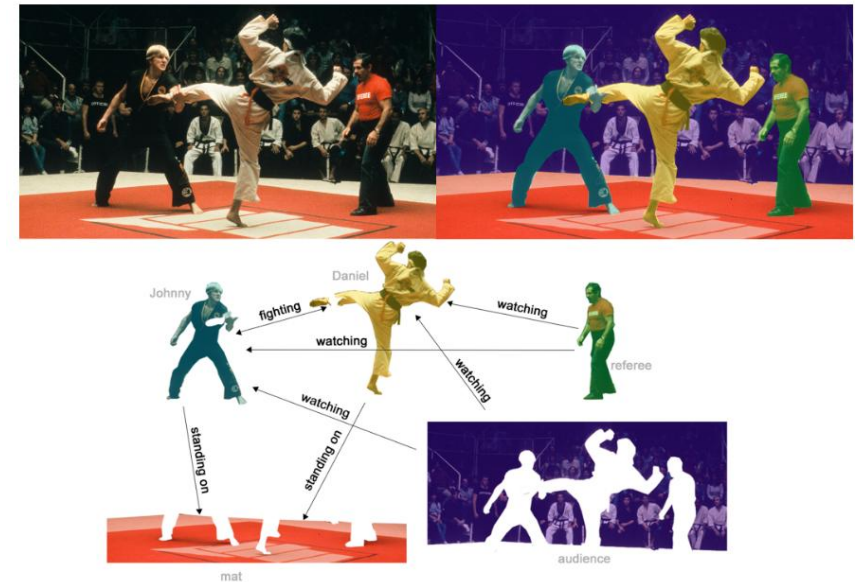


Social networks as graphs
Citation networks as graphs

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

What is the application of graph?

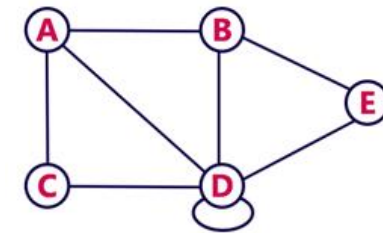
- Graph-level task [like image classification]
 - For example, for a molecule represented as a graph, we might want to predict what the molecule smells like, or whether it will bind to a receptor implicated in a disease.
- Node-level task [like image segmentation]
 - Predict the property of nodes. E.g. predict the attitude of citizens on Biden or Trump.
- Edge-level task



In (b), above, the original image (a) has been segmented into five entities: each of the fighters, the referee, the audience and the mat. (c) shows the relationships between these entities.

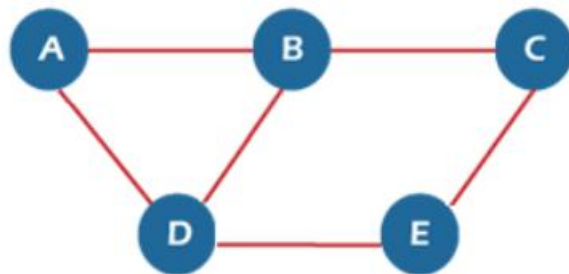
How to represent/save a graph?

- Adjacency matrix. (Cons: sparse adjacency matrices are space-inefficient.) $O(\pi \sim \pi)O$

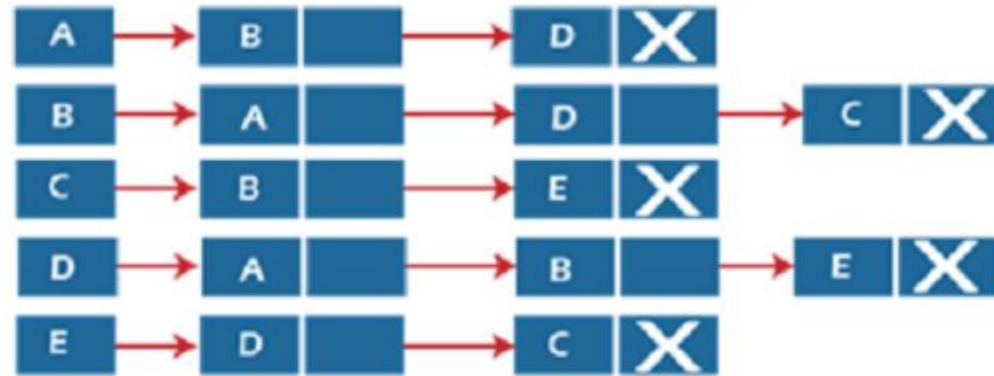


	A	B	C	D	E
A	0	1	1	1	0
B	1	0	0	1	1
C	1	0	0	1	0
D	1	1	1	1	1
E	0	1	0	1	0

- Adjacency lists.



Undirected Graph



Adjacency List

How to represent/save/show a graph?

```
import networkx as nx
```

```
graph = nx.Graph() # empty graph
```

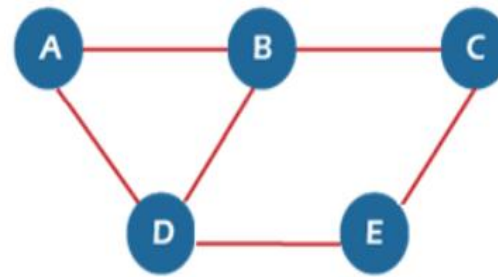
```
node0 = (1.1, -1.5, 1.2) # property of a node  
graph.add_node(node0) # add a node
```

```
node1 = (5.2, 2.5, -0.4)  
graph.add_node(node1)
```

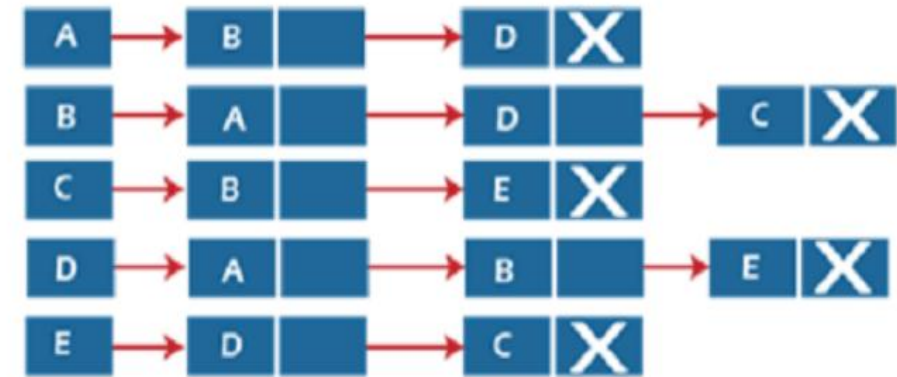
```
graph.add_edge(0,1) # add an edge
```

```
pickle.dump(graph, 'path_to_save', 'wb') # save the graph
```

More examples: <https://networkx.org/documentation/stable/tutorial.html>



Undirected Graph

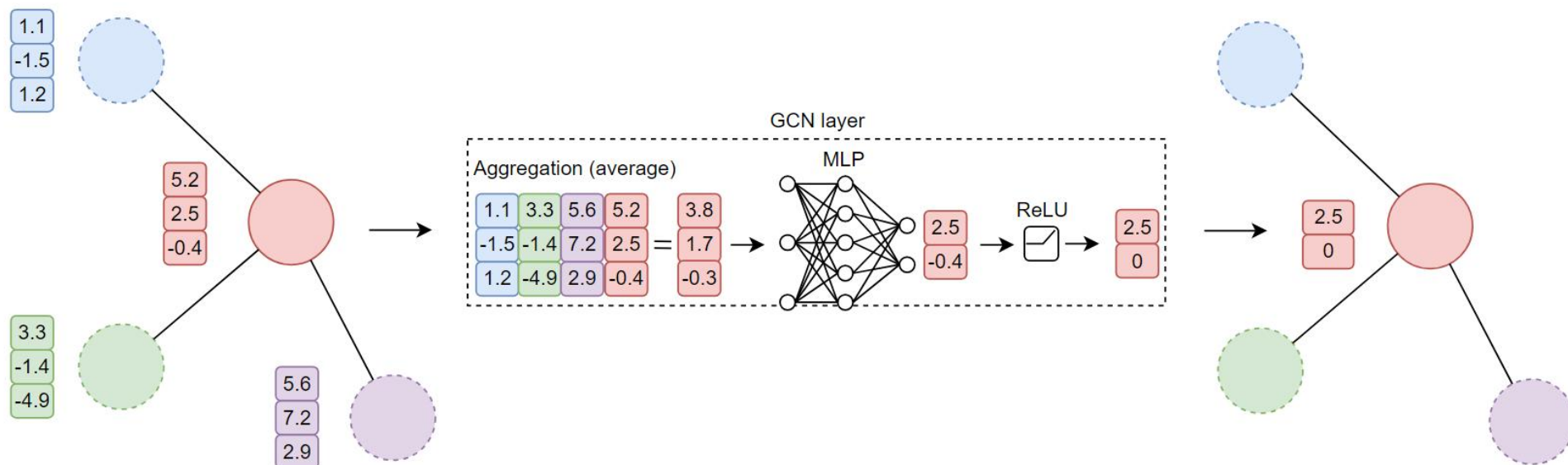


Adjacency List

What is graph neural network?

- Neural networks which receive graph as input, aggregate graph information, compute hidden features, and output prediction values.

What is graph convolution network?



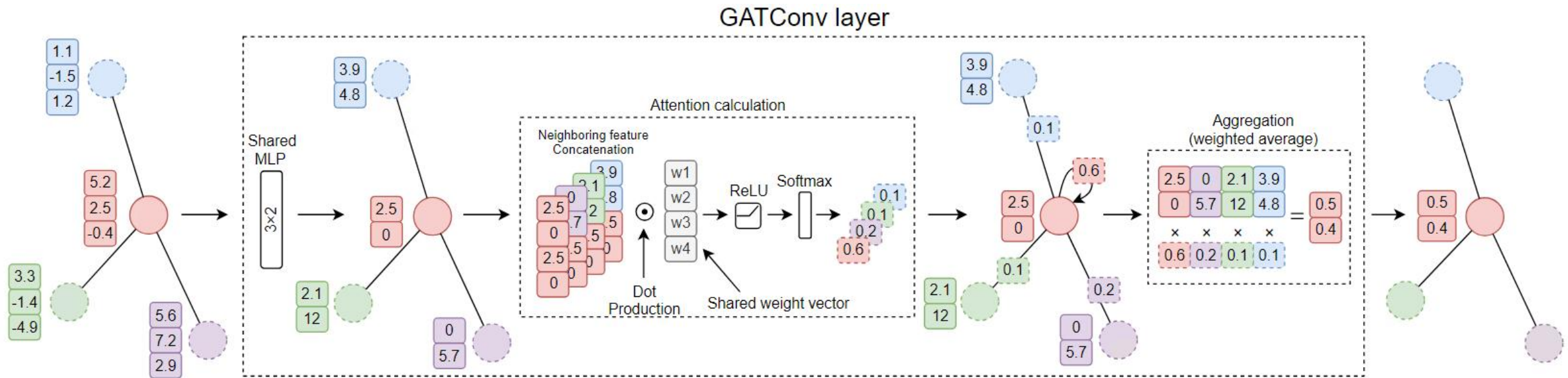
```
import networkx as nx
from torch_geometric.loader import DataLoader
from torch_geometric.nn import GCNConv
```

```
graph_ls = [g1, g2, g3, ...] # build a list of graphs
graph_data_loader = DataLoader(graph_ls)
```

```
net = GCNConv(in_chn=3, out_chn=2)
```

```
for data_batch in graph_data_loader:
    net(data_batch.x, data_batch.edge_index, data_batch.batch)
```

What is graph attention network?



More graph neural networks

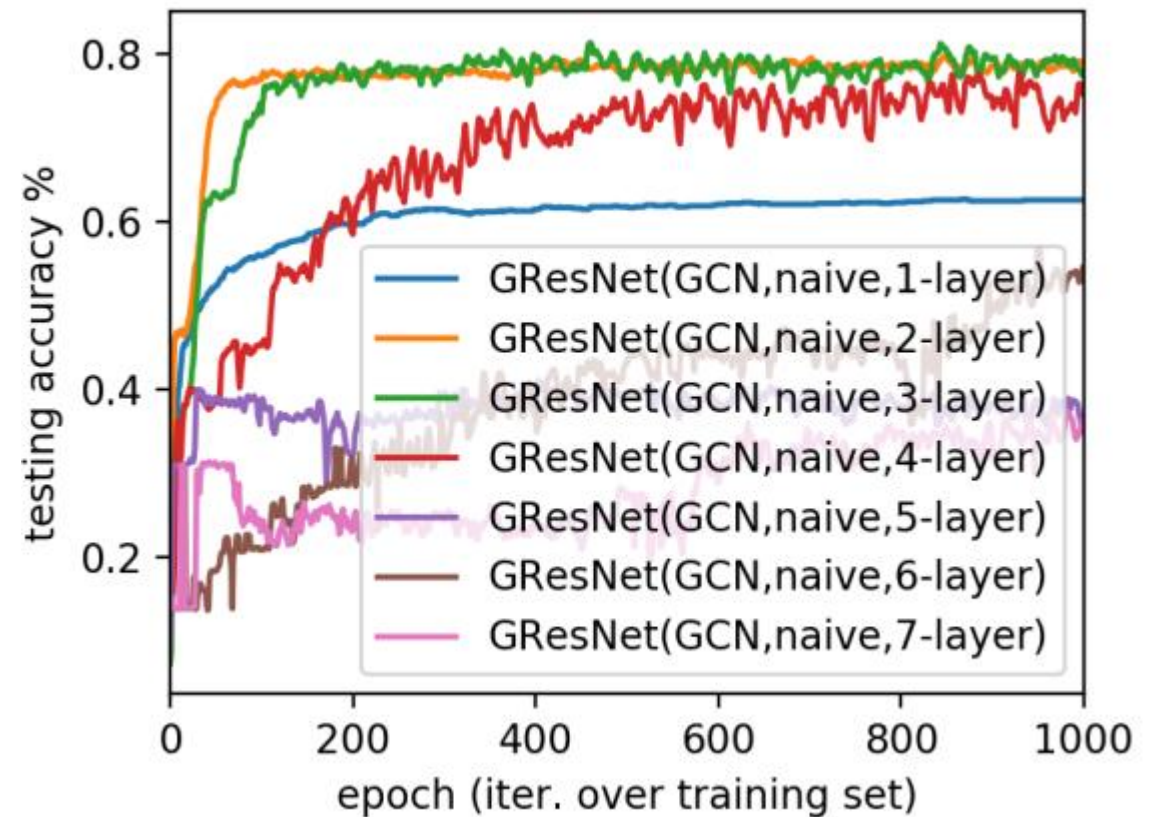
- <https://pytorch-geometric.readthedocs.io/en/latest/modules/nn.html#convolutional-layers>

What is the tips/experience to train a graph neural network?

- GNN is very fast!

	Train- ing time [hour]	#par	#FLOPs [GB]
CNN_CT	3.7	30.2	177.62
CNN_Vessel	3.7	30.2	177.62
PNN_Vessel	1.1	2.1	13.8
GCN_Vessel	0.25	1.2	2.4

- GNN needs only 2-4 layers



Others

- Optuna (<https://optuna.org>): an easy tool to automatically explore the hyper-parameters

Hyper-parameters	Search space (candidate values)
Learning rate	1e-3 , 1e-4
Batch size	8, 16, 32 , 64
Hidden channels	32, 64, 128 , 256
Convolution layers	1, 2, 3, 4
Convolution kernel	ChebConv [22], GINConv [29], GCNConv [27], GATConv [32], SGConv [26], GraphConv [28]
Normalization	BatchNorm [37], InstanceNorm [38], LayerNorm [39], GraphNorm [40], DiffGroupNorm [41]

- How to decide which topic to research?

Thank you

