Hands-on Introduction to Deep Learning

Graph Neural Network (GNN)

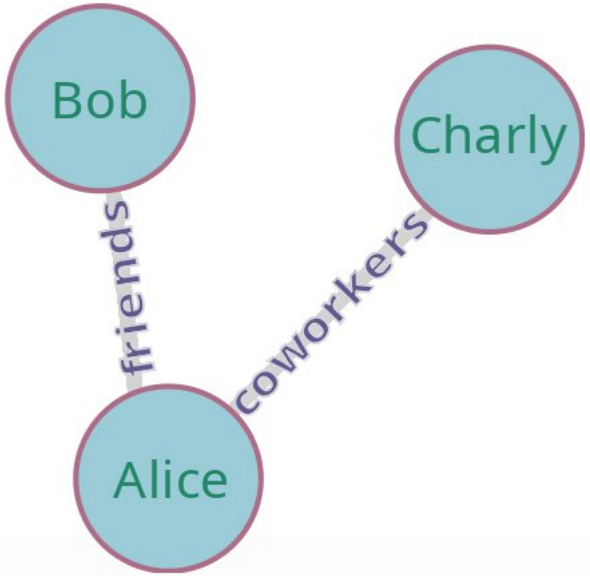
INSTITUT DU
DÉVELOPPEMENT ET DES
RESSOURCES EN
INFORMATIQUE
SCIENTIFIQUE

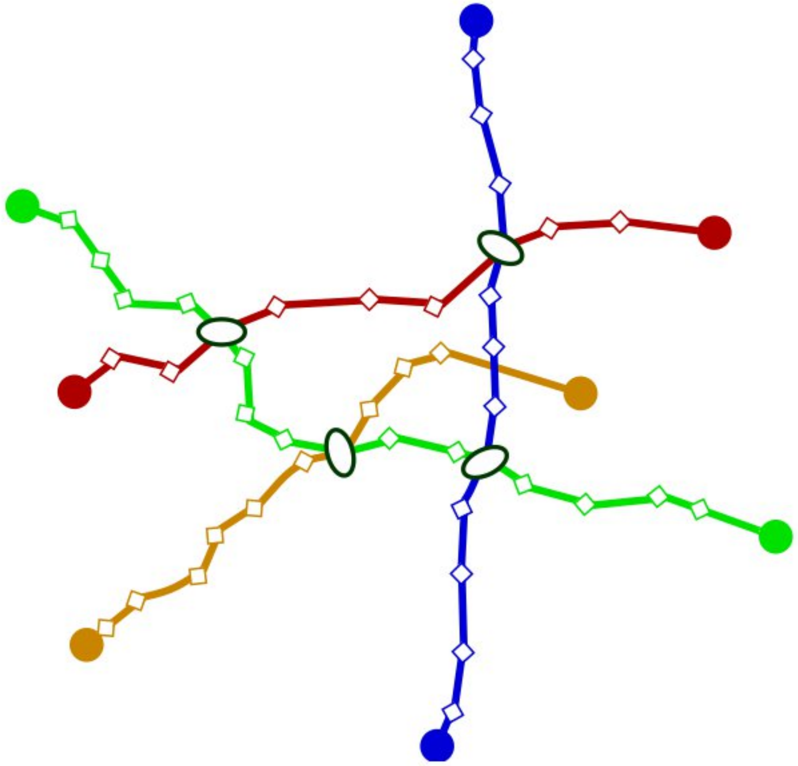
Damn you Euclid

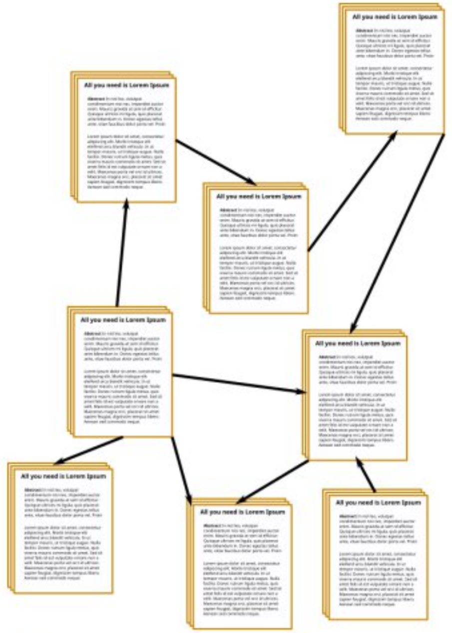
- AI success was mainly due to computer vision, speech recognition, text completion…
  - Highly structured data

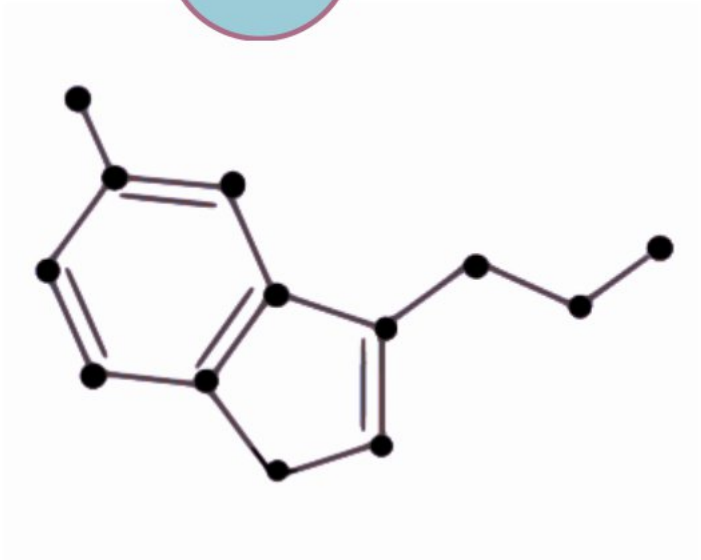
What about other problems ?

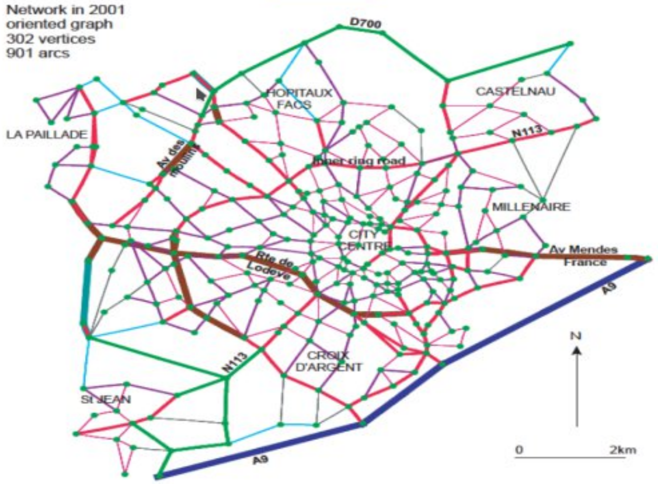
Chemistry, social science, physics, etc











Entities and relationships: nodes/vertices and edges

- Graphs can store information (features) on **nodes**, **edges**, and **globally**

	Globally	Nodes	Edges
Social Network	Group of interest, ...	Name, Age, Job, ...	Is friend, follows, family, ...
Molecule	Is a drug, Energy, ...	Atomic number, ...	Bond order, ...
Citations	Field, ...	Article, ...	Was cited, ...
Particle physics	Experiment	Particle	Decayed to, ...
Motion capture	Character	Joints	Is connected to, ...
Natural language	Paragraph, ...	Group of words, ...	Refers to, ...

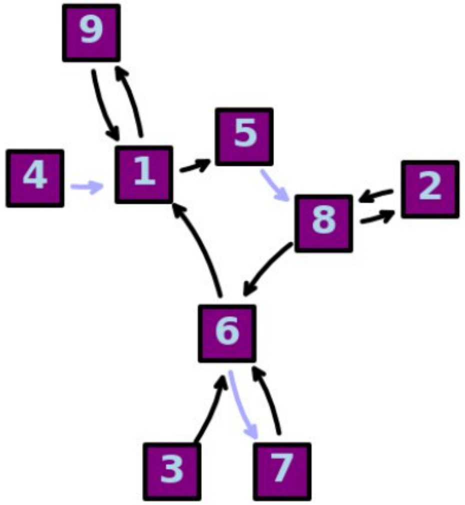
It can be a number, a concept, ...

What is a graph ?

- # Direction

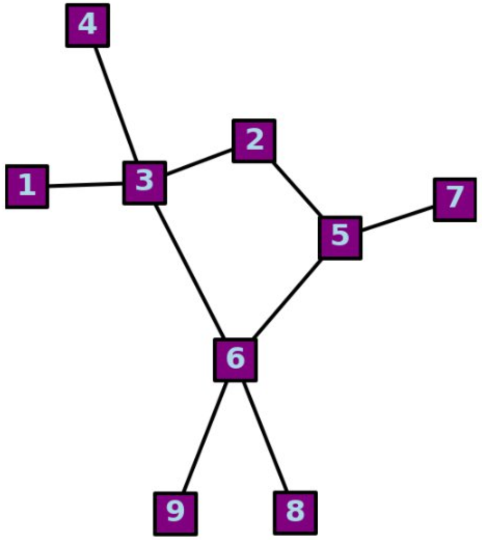
Directed : Relationships are not symmetric

- On twitter, you can follow someone but not be followed by this person
- A paper is cited in another paper



Undirected : Relationships are not symmetric

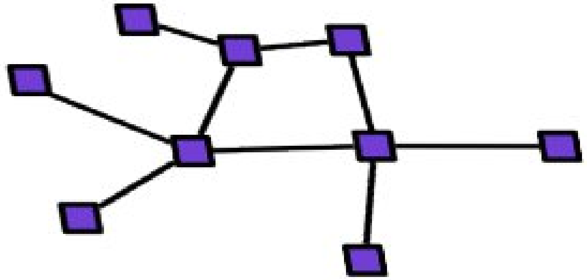
- 2 atoms share the same kind of bond



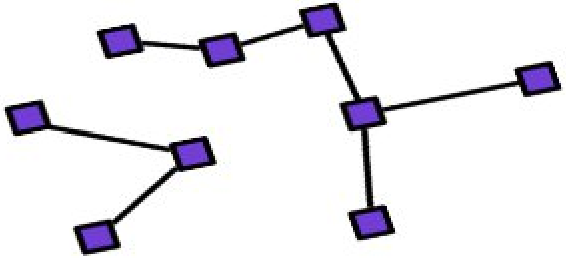
Graph properties

- Connectivity on undirected graphs

All nodes are connected via a path → **Connected Graph**



If some nodes are not connected to other via a path, they are **disconnected**

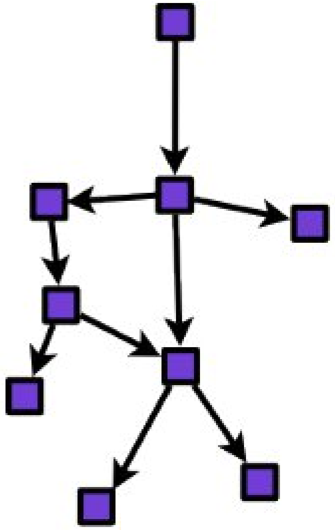


Graph properties

• Connectivity on directed graphs

Weakly Connected

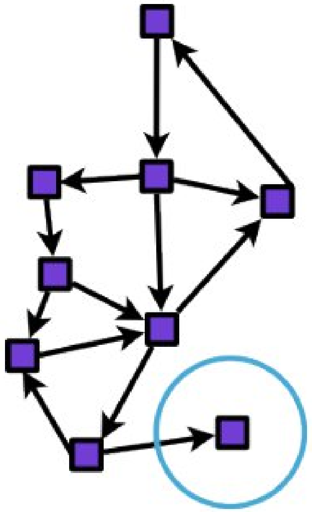
If you replace all directed edges by undirected edges, the "undirected" graph is connected



Unilaterally Connected

For each pair of node {u, v}, there is a directed path
- u → v

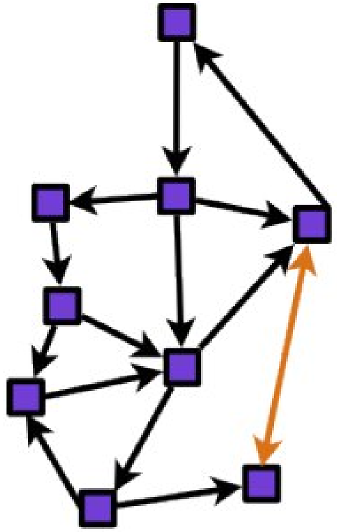
  or
- v → u



Strongly Connected

For each pair of node {u, v}, there is a directed path
- u → v

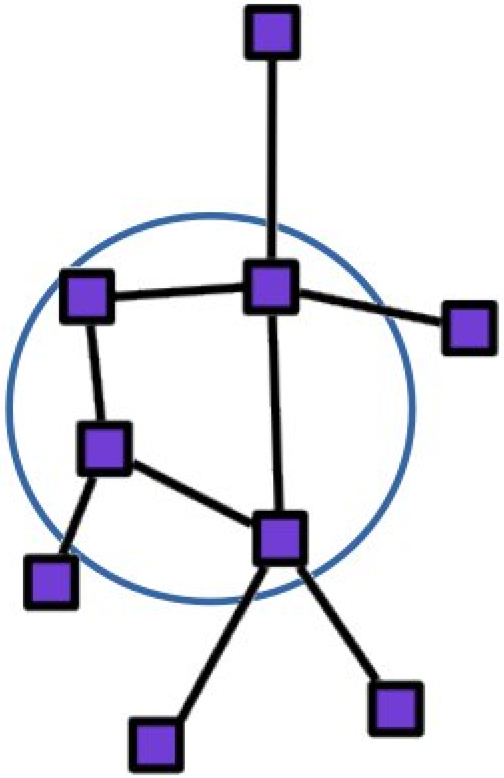
  and
- v → u



Graph properties

- Cycles

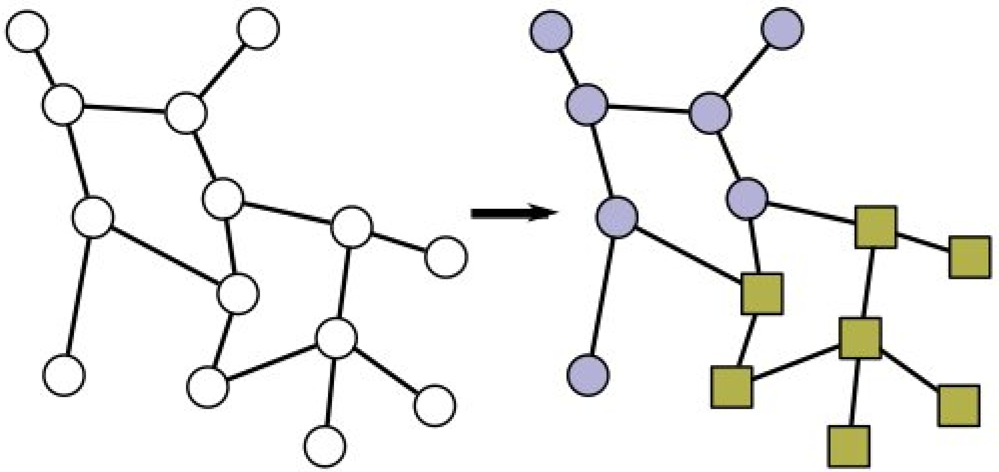
If there is a path with which you can go back to the starting node, there is a cycle



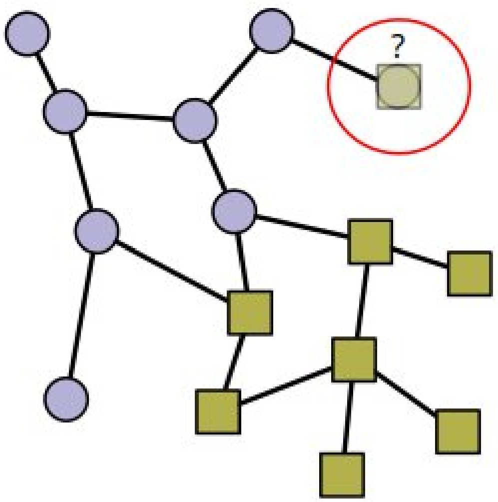
Graph properties

What tasks can we do with graphs ?

- Labeling nodes in a graph
  - Find topic of a research paper (CORA, etc)
  - Find bots in a social network
  - …



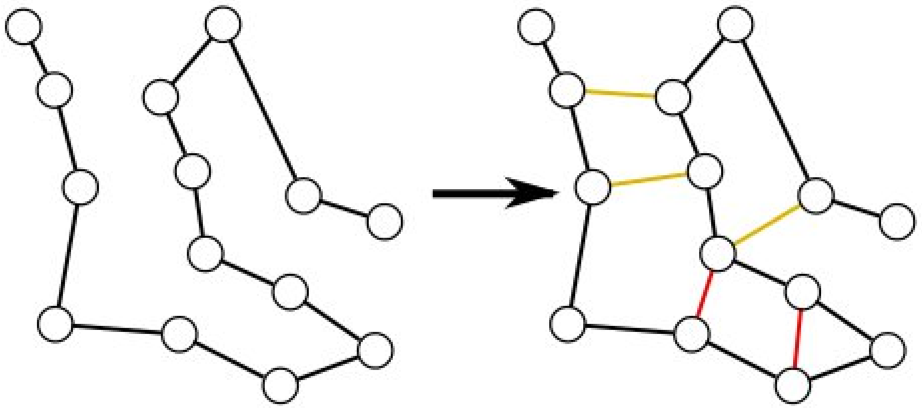
- Give a label to a new node

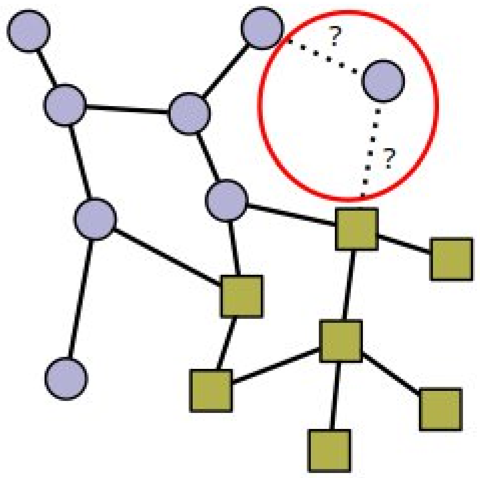


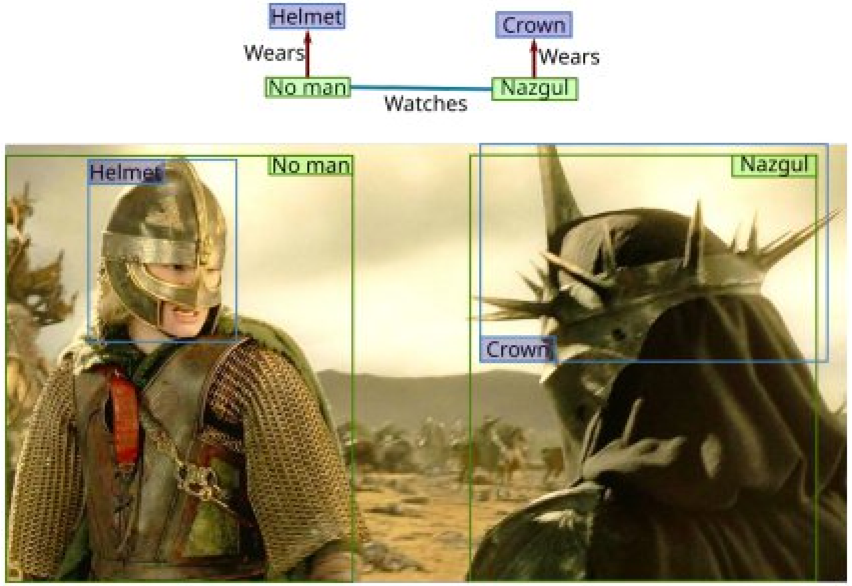
- Regression

Tasks on graphs : Node prediction

- Find relationships
  - Contact map of aminoacids (Alphafold)
  - Contact suggestion (social network)
  - ETA for directions (regression)
  - Relation between segment in pictures
  - ...

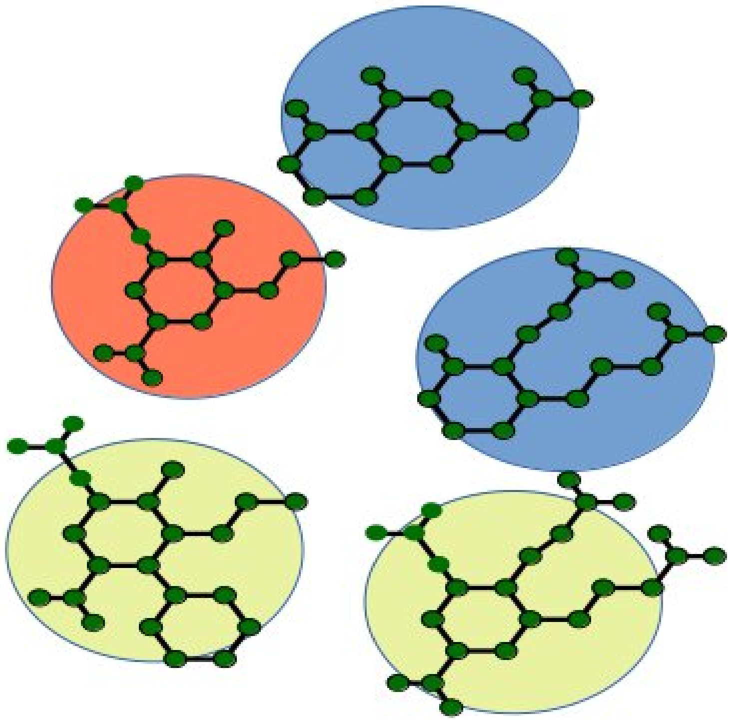






Tasks on graphs : Edge prediction

- Predict properties of a graph
  - Chemical properties (solubility, carcinogenic, possible drug)
  - Classification of the research field in an ego network
  - ...

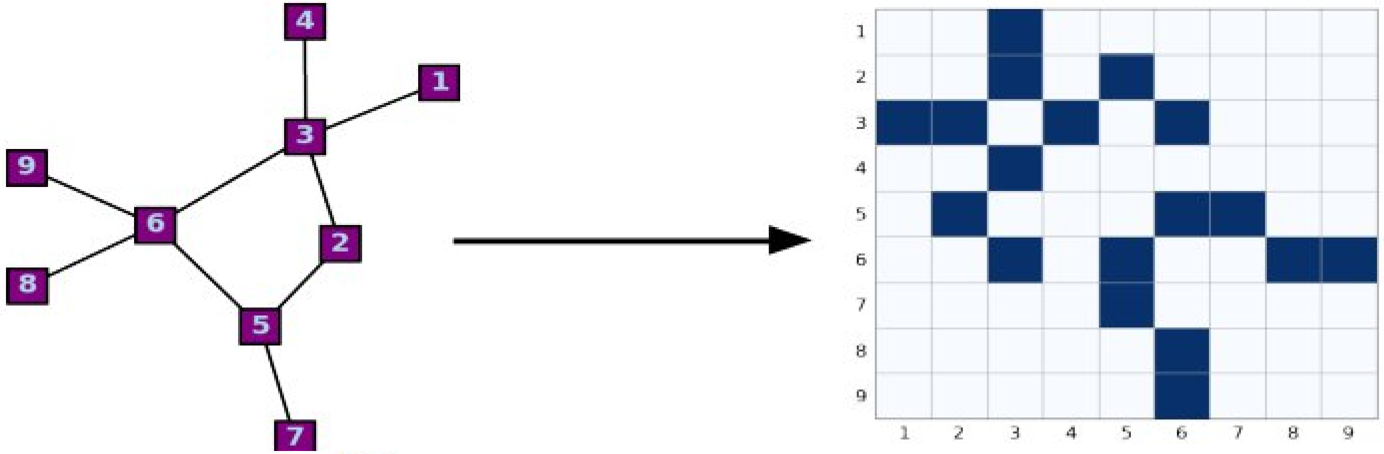


Tasks on graphs : Graph prediction

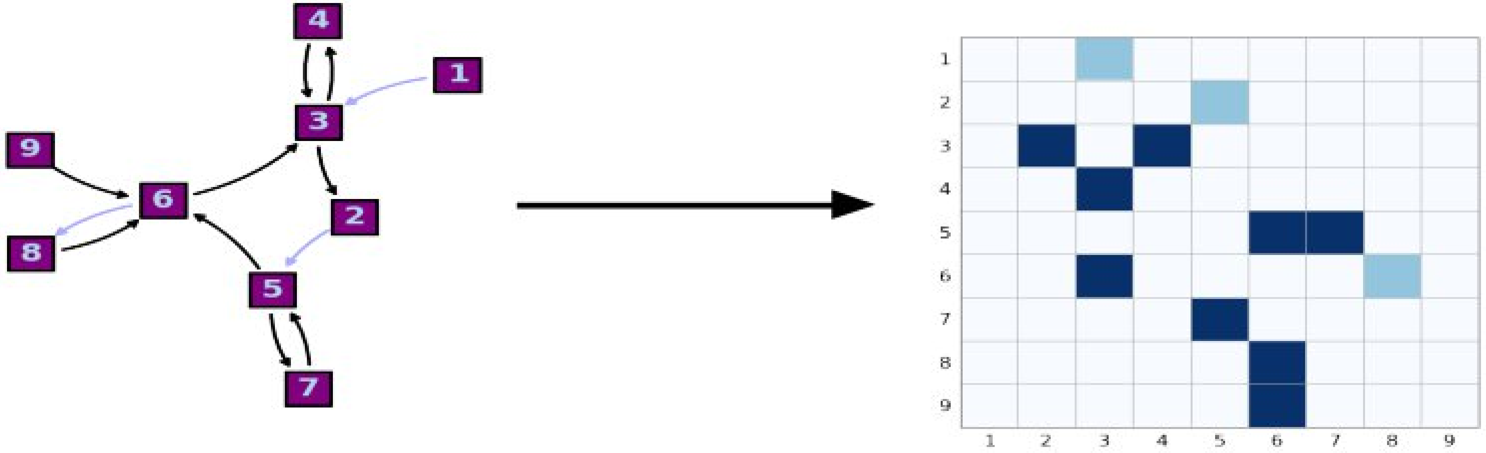
How to represent graphs ?

How to represent a graph

- Adjacency matrix
  - N × N matrix with value ≠ 0 when an edge exists



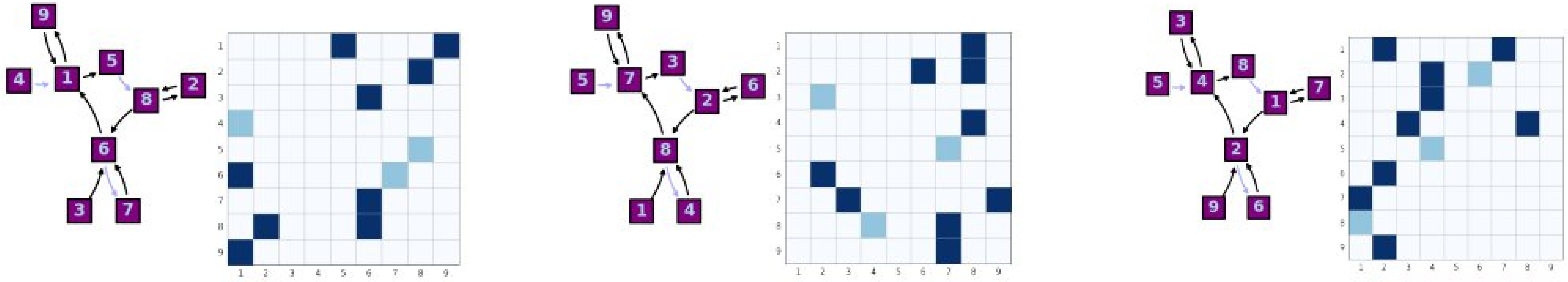
For undirected graphs, the adjacency matrix is symmetric



For undirected graphs, the adjacency matrix is NOT symmetric

- Problems with adjacency matrices

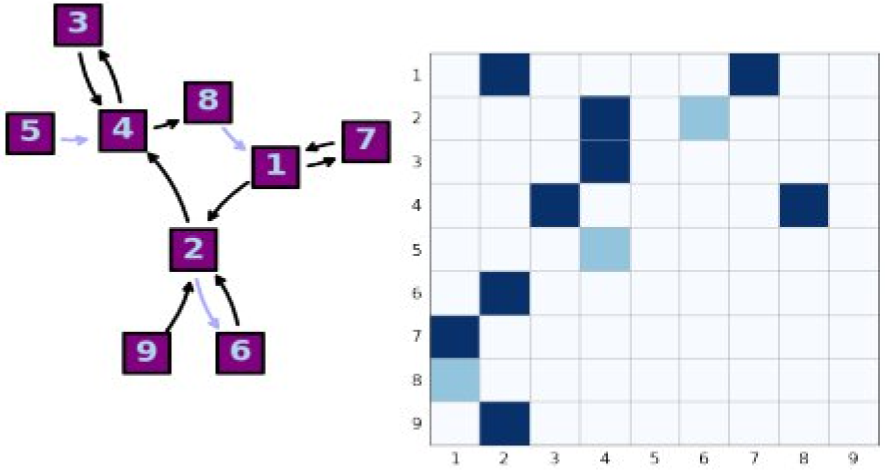
  – The size grows as **N × N** → problems with storage
  – The matrix is likely to be **sparse**
  – **N!** permutations represent the same graph



Difficult and inefficient to store and different representations are not guaranteed to give the same results

How to represent a graph

• Adjacency list



Nodes: [1.0, 1.0, 1.0, 1.0, 1.0,
1.0, 1.0, 1.0, 1.0]

Edges: [0.4, 0.4, 1.0, 1.0, 1.0, 1.0,
1.0, 1.0, 0.4, 1.0, 1.0, 1.0]

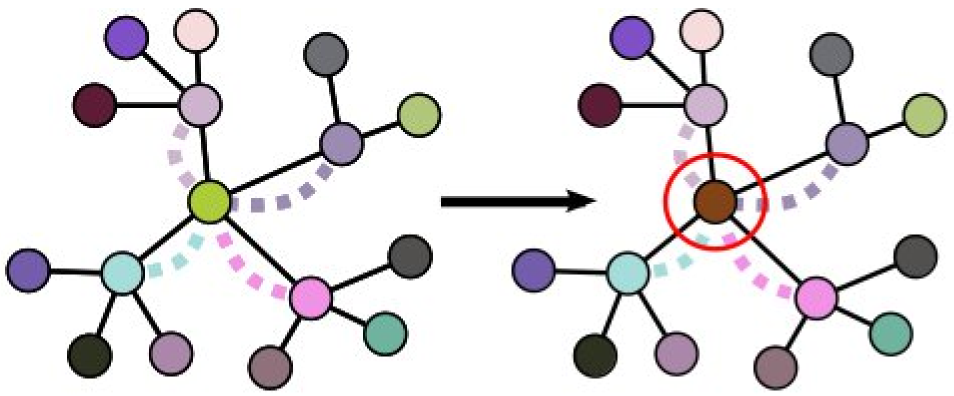
Adjacency list: [[5, 4],
[8, 1],
[4, 8], [4,3],
[3, 4],
[1, 7], [1, 2],
[2, 4], [2, 6],
[7,1],
[6, 2],
[9, 2]]

Global: [1.0, 1.0]

How to represent a graph

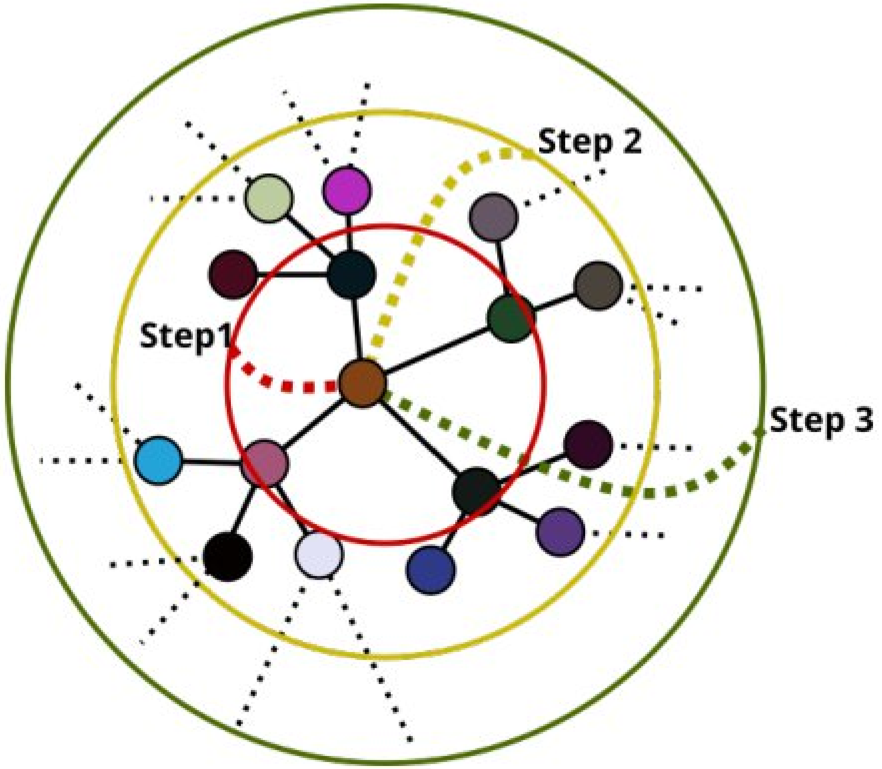
Learn on graphs

- Just like for pictures we can learn from neighborhood with a convolution operator



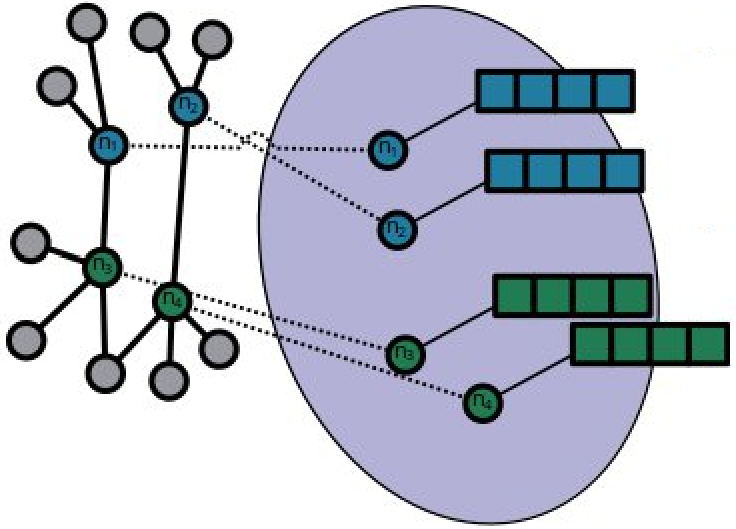
- A bit more complex since the number of neighbors is unlikely to be constant
- We want the operator to be permutation invariant

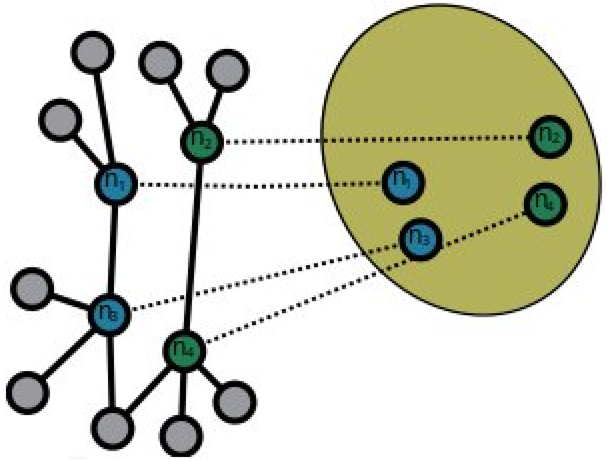
Graph convolution



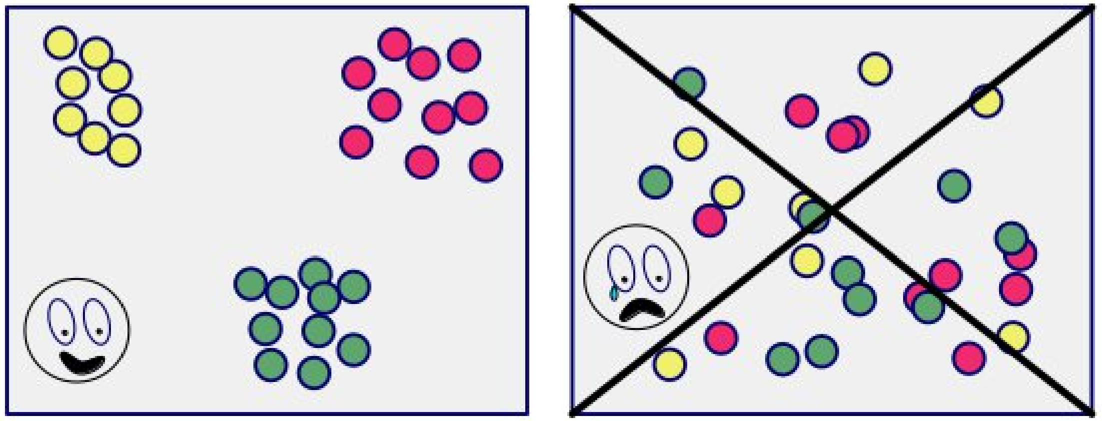
- Several steps are needed to retrieve information for distant nodes
- For large graphs → **cutoff**
- It is possible to use a **virtual node** connected to all other nodes. But in practice it becomes intractable quickly

Graph convolution





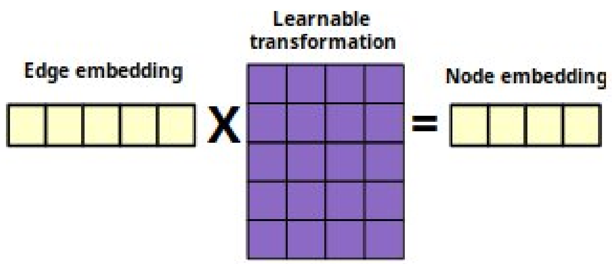
- Features stored in nodes/edges/graphs are not easily processable
- We transform the features into a vector in the latent space (**dimension is a hyperparameter**)
- The embedding have to be suited for the task → **Learnable**

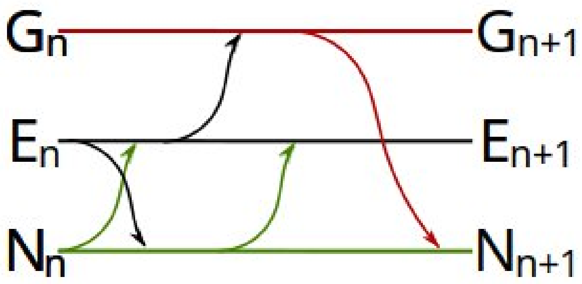


Compress information : Embeddings

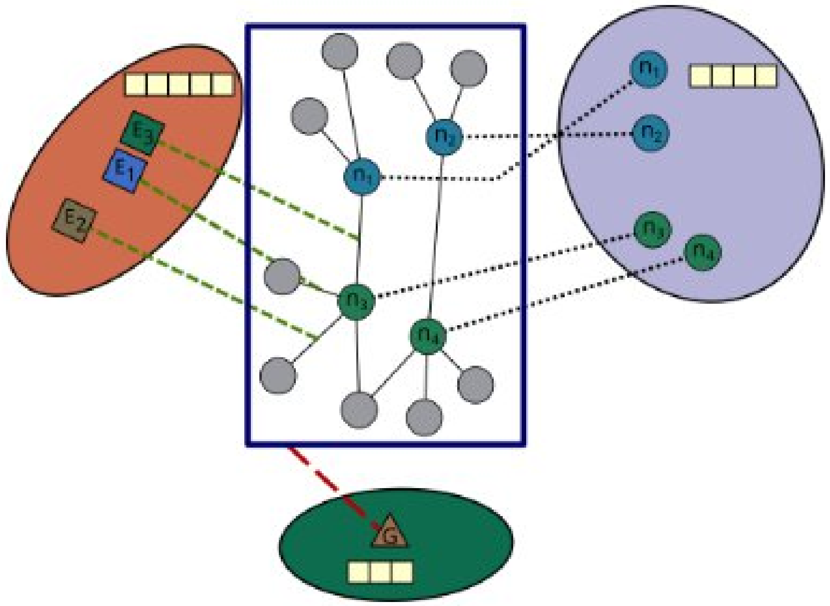
Message passing : Share information

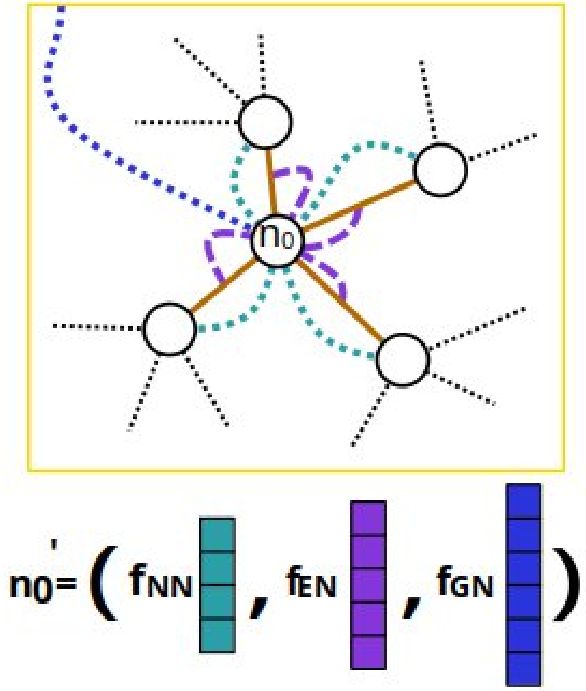
- We have embeddings for each part of the graph (possibly different vector sizes)
- Each part can learn from the others with a transformation





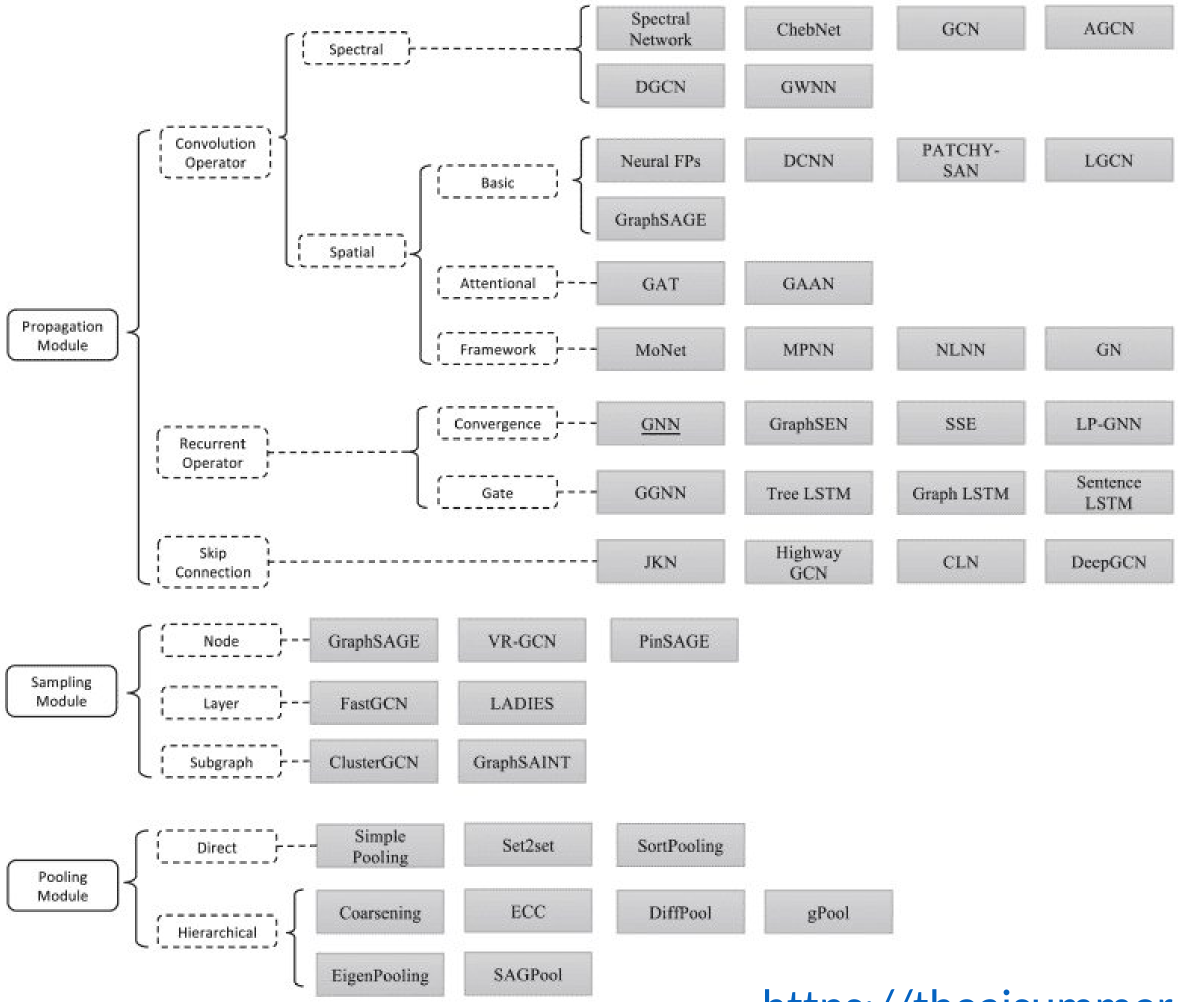
- The information is aggregated to form a message that the node/edge will send to others





Conclusions

Several model architectures are available



https://theaisummer.com/gnn-architectures/

Models

- It is possible to use Deep Learning on non-Euclidean data structures. The field is called **Geometric Deep Learning** https://geometricdeeplearning.com/
- Graph structures appear easily on many scientific problems
- GNN can be seen as a generalization of convolution
- We can aggregate features to form a message to be passed
- There are several models already available
- A large part of the problem is to find a good way to transform the original data to fit NN architectures → Representation learning

Conclusion

- Pytorch Geometric
- Deep Graph Library
- Graph Nets
- Spektral
- …

Available libraries

References

- Books
  - Deep Learning on Graphs (Jiliang Tang and Yao Ma)
  - Introduction to Graph Neural Networks (Zhiyuan Liu and Jie Zhou)

- Websites
  - https://distill.pub/2021/gnn-intro/
  - https://neptune.ai/blog/graph-neural-network-and-some-of-gnn-applications
  - https://venturebeat.com/ai/what-are-graph-neural-networks-gnn/
  - https://theaisummer.com/graph-convolutional-networks/
  - https://towardsdatascience.com/node-embeddings-for-beginners-554ab1625d98

- Articles

  - **Zhou, Jie, et al. "Graph neural networks : A review of methods and applications." AI Open 1 (2020) : 57-81.**
  - **Scarselli, Franco, et al. "The graph neural network model." IEEE transactions on neural networks 20.1 (2008) : 61-80**
  - **Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." arXiv preprint arXiv:1609.02907 (2016).**
  - **Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations." Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014.**
  - **Shlomi, Jonathan, Peter Battaglia, and Jean-Roch Vlimant. "Graph neural networks in particle physics." Machine Learning : Science and Technology 2.2 (2020) : 021001.**
  - **Duong, Chi Thang, et al. "On node features for graph neural networks." arXiv preprint arXiv:1911.08795 (2019).**