Machine Learning: Algorithms and Applications

Philip O. Ogunbona

Advanced Multimedia Research Lab
University of Wollongong

Artificial Neural Networks and Deep Learning: An Introduction
Graph Neural Networks - Graph Convolutional Networks (GCN)
Autumn

Outline

World of graphs

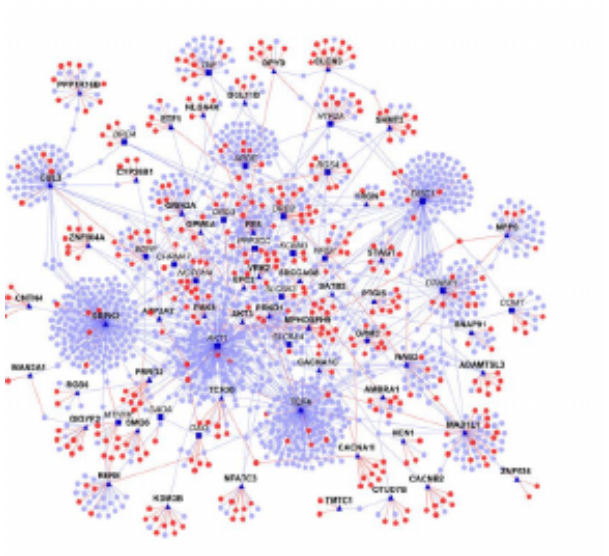
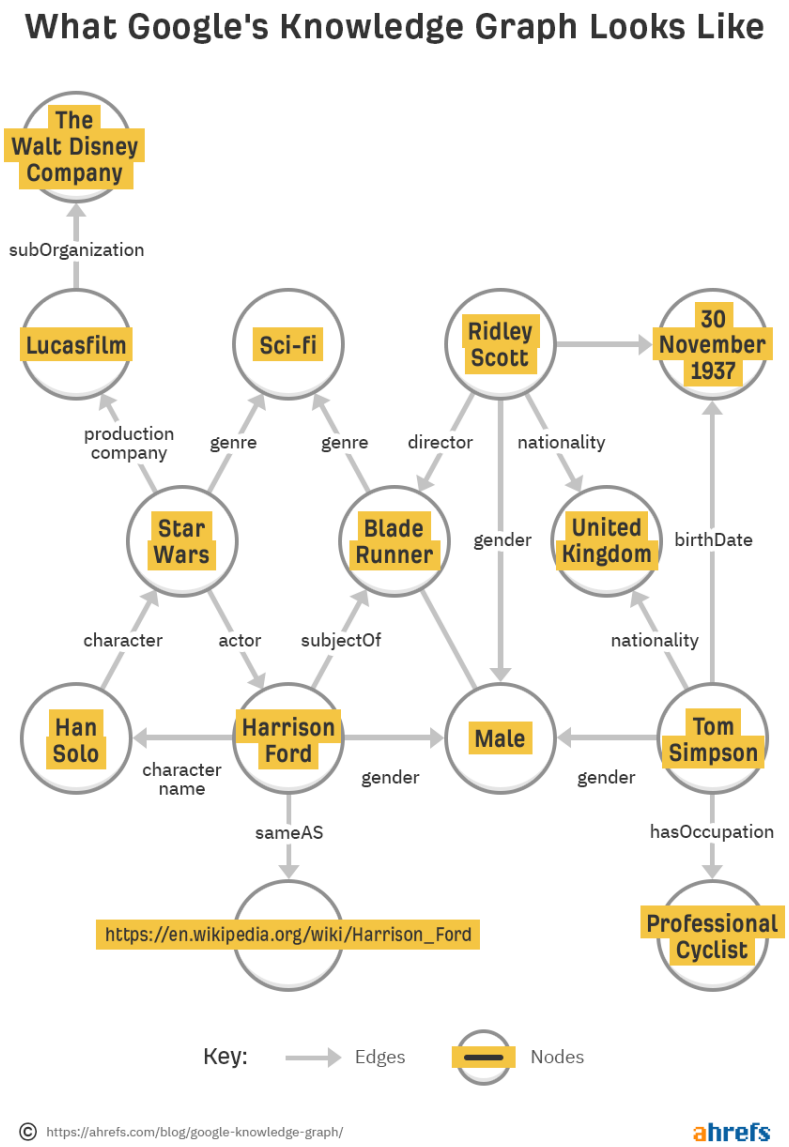


Figure 1: Protein interaction network (Credit: Thomas Kipf, University of Amsterdam)

World of graphs

Figure 1: Social network (Credit: Thomas Kipf, University of Amsterdam)

World of graphs



World of graphs

Figure 1: Road map can be modelled as graph (Credit: Thomas Kipf, University of Amsterdam)

World of graphs

Definition 1 (Graph)

A graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is defined as a set of vertices, $\mathcal{V}$, which are connected by a set of edges, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, where the symbol $\times$ denote a direct product operator.

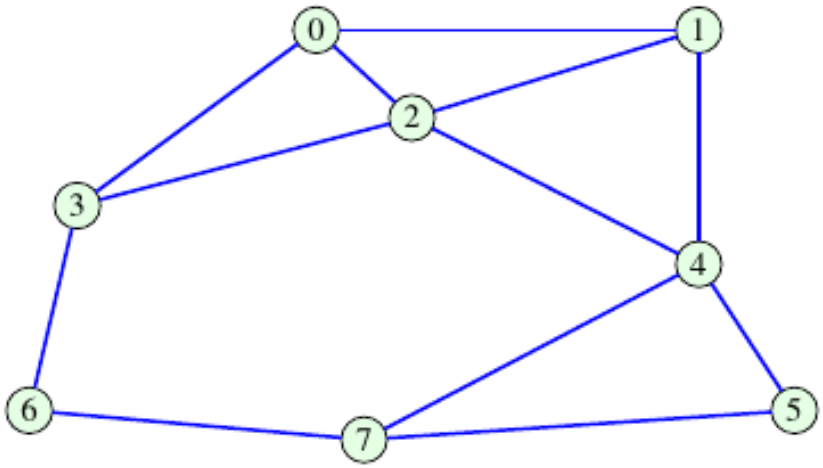


Figure 2: Undirected graph

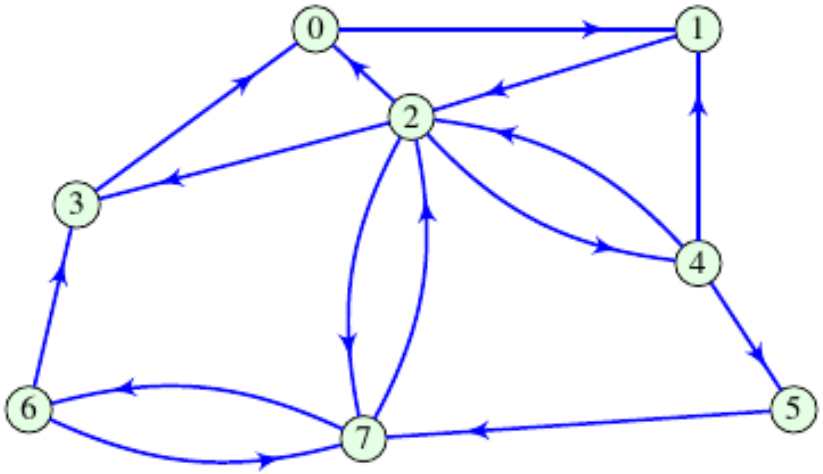


Figure 3: Directed graph

World of graphs

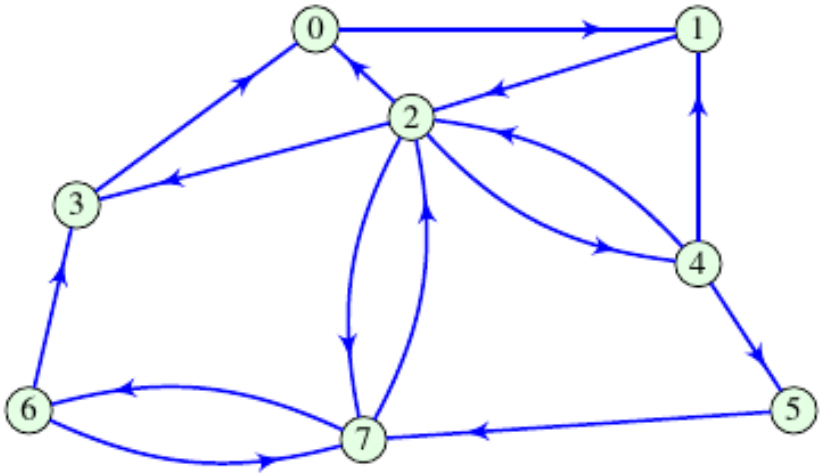


Figure 4: Directed graph (same graph as in Fig 3)

Representation of the graph edge connectivity

$$\mathcal{V} = \{0, 1, 2, 3, 4, 5, 6, 7\}$$
$$\mathcal{E} \subset \{0, 1, 2, 3, 4, 5, 6, 7\} \times \{0, 1, 2, 3, 4, 5, 6, 7\}$$
$$\mathcal{E} = \{(0, 1), (1, 2), (2, 0), (2, 3), (2, 4), (2, 7), (3, 0), (4, 1), (4, 2), (4, 5), (5, 7), (6, 3), (6, 7), (7, 2), (7, 6)\}$$

World of graphs

For a given set of vertices and edges, a graph can be formally represented by its adjacency matrix, $\boldsymbol{A}$, which describes the vertex connectivity; for N vertices $\boldsymbol{A}$ is an $N \times N$ matrix.

Definition 2 (Adjacency matrix)

The elements A_{mn} of the adjacency matrix $\boldsymbol{A}$ assume values $A_{mn} \in \{0, 1\}$. The value $A_{mn} = 0$ is assigned if the vertices m and n are not connected with an edge, and $A_{mn} = 1$ if these vertices are connected, that is

$$A_{mn} \overset{def}{=} \begin{cases} 1, \text{if } (m, n) \in \mathcal{E} \\ 0, \text{if } (m, n) \notin \mathcal{E} \end{cases} \tag{1}$$

The adjacency matrix of an undirected graph is symmetric;

$$\boldsymbol{A} = \boldsymbol{A}^T.$$

World of graph

Definition 3 (Weight matrix)

A nonzero element in the weight matrix $\boldsymbol{W}$, $W_{mn} \in \mathcal{W}$, designates both an edge between the vertices m and n and the corresponding weight. The value $W_{mn} = 0$ indicates no edge connecting the vertices m and n. The elements of a weight matrix are nonnegative real numbers.

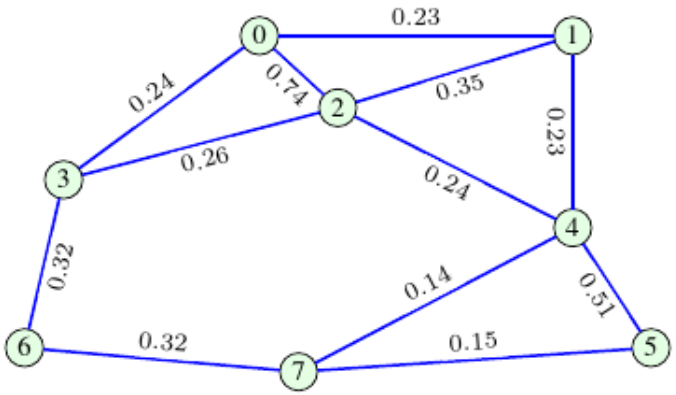


Figure 5: Weighted graph

$$\mathbf{W} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{bmatrix} 0 & 0.23 & 0.74 & 0.24 & 0 & 0 & 0 & 0 \\ 0.23 & 0 & 0.35 & 0 & 0.23 & 0 & 0 & 0 \\ 0.74 & 0.35 & 0 & 0.26 & 0.24 & 0 & 0 & 0 \\ 0.24 & 0 & 0.26 & 0 & 0 & 0 & 0.32 & 0 \\ 0 & 0.23 & 0.24 & 0 & 0 & 0.51 & 0 & 0.14 \\ 0 & 0 & 0 & 0 & 0.51 & 0 & 0 & 0.15 \\ 0 & 0 & 0 & 0.32 & 0 & 0 & 0 & 0.32 \\ 0 & 0 & 0 & 0 & 0.14 & 0.15 & 0.32 & 0 \end{bmatrix} \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}$$

Figure 6: The associated weight matrix of graph in Figure 5

World of graphs

More graphs (Broadwater & Stillman 2023)

Heterogeneous or homogeneous: Heterogeneous graphs contain nodes of different classes or categories. For example, in a recruitment network, some nodes may be employees and others may be companies. On the other hand, homogeneous graphs assume that all nodes are of the same class.

Bipartite: Similar to heterogeneous graphs, bipartite graphs also reflect graphs which can be separated or partitioned into different subsets. However, bipartite graphs have a very specific network structure such that nodes in each subset connect to nodes outside of their subset and not inside.

Cyclic/Acyclic: Cyclic and acyclic graphs describe the network connectivity. Cyclic graphs are such that if you can start from one node and move across the network you will eventually return back to the original starting node. For acyclic graphs, if you start at any node in the graph, you will not be able to return to your starting node unless you retrace your steps. Directed acyclic graphs, or DAGs, are a subset of acyclic graphs which have directed edges. DAGs are extremely important in causal analysis, as they reflect causal structure, which is widely assumed to be unidirectional, i.e. A can cause B but B cannot cause A.

World of graphs

More graphs (Broadwater & Stillman 2023)

Knowledge graphs Similar to DAGs, knowledge graphs are those that represent abstract concepts in the world. However, knowledge graphs are not restricted to being acyclic or directed. Nodes and edges in knowledge graphs are abstract entities and concepts, and relay semantic relationships. A graph is classified as a knowledge graph when it is structured to semantically represent entities and their interrelations.

Graph neural network (GNN) training

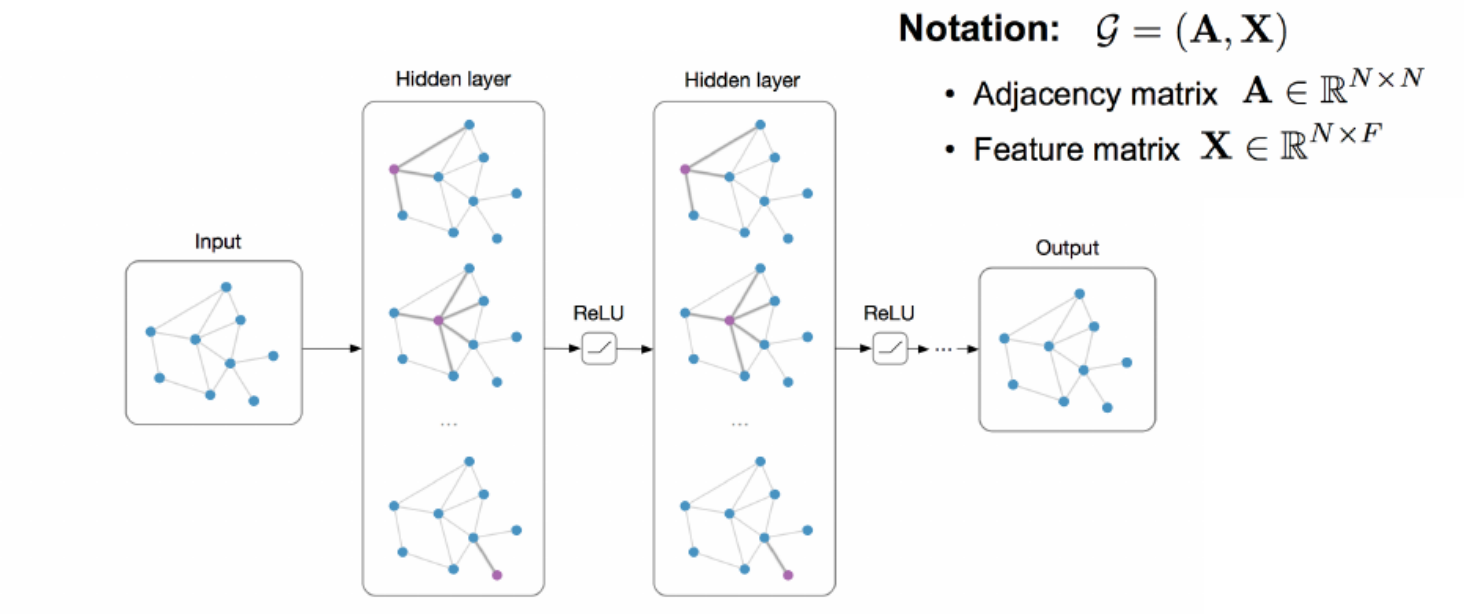


Figure 7: GNN Training (Credits: Thomas Kipf, University of Amsterdam)

Main idea: Pass message between pairs of nodes and agglomerate

In essence: Pass messages between nodes (vertices) to refine node representation. Edges can be refined too.

GNN training

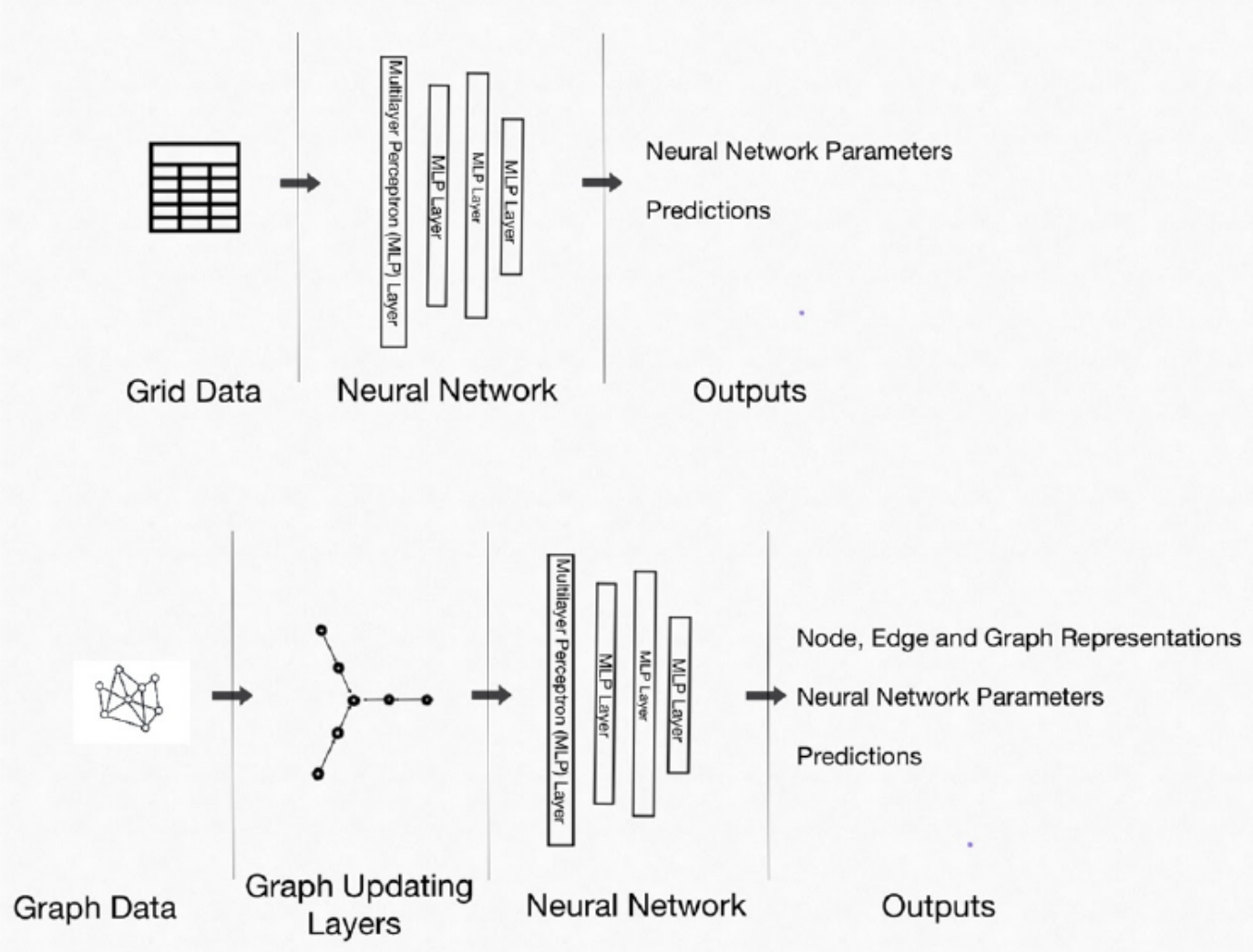


Figure 8: Basic Comparison of CNN and GCN (Broadwater & Stillman 2023)

GNN training

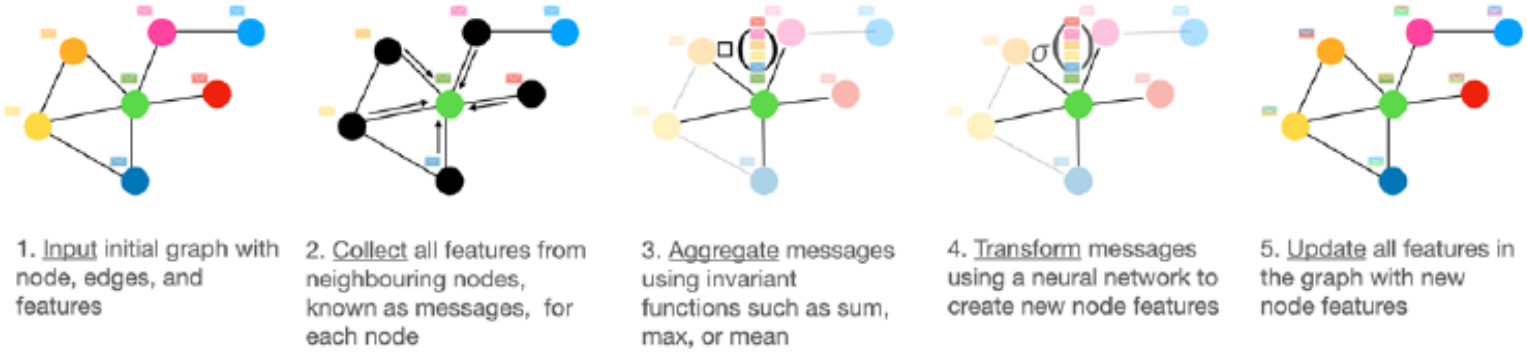


Figure 9: Elements of message passing (Broadwater & Stillman 2023)

1. Input initial graph with node, edges and features
2. Collect all features from neighbouring nodes; known as messages, for each node
3. agglomerate (aggregate) messages using invariant function such as sum , max, or mean
4. Transform messages using neural network to create new node feature
5. Update all features in the graph with new node features

GNN Training

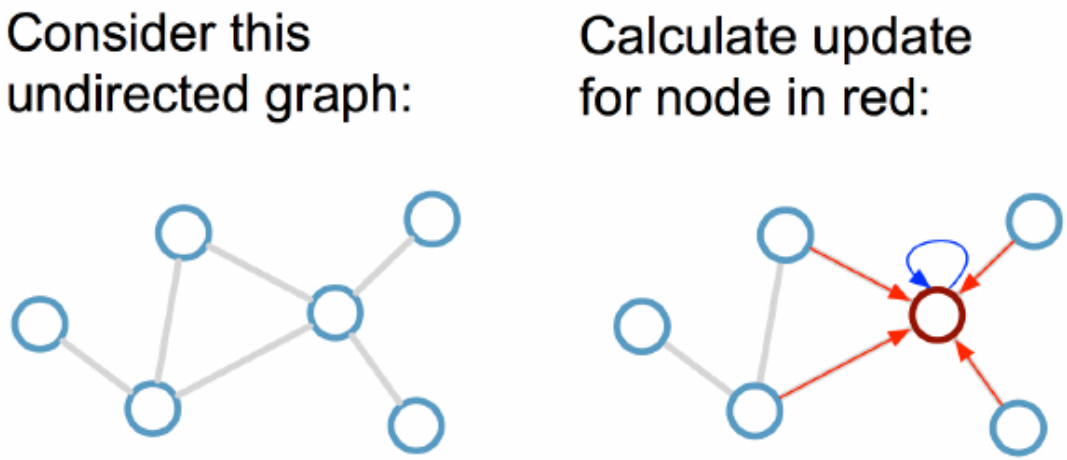


Figure 10: Updating a node (Credits: Thomas Kipf, University of Amsterdam)

Update rule:

$$h_i^{(l+1)} = \sigma\left(h_i^{(l)} W_0^{(l)} + \sum_{j \in N_i} \frac{1}{c_{ij}} h_j^{(l)} W_1^{(l)}\right) \tag{2}$$

N_i: neighbour indices; c_{ij}: norm constant; fixed or trainable.

Good properties: weight sharing over all locations (W); invariance to permutations; linear complexity; applicable in both transductive and inductive settings.

GNN usage

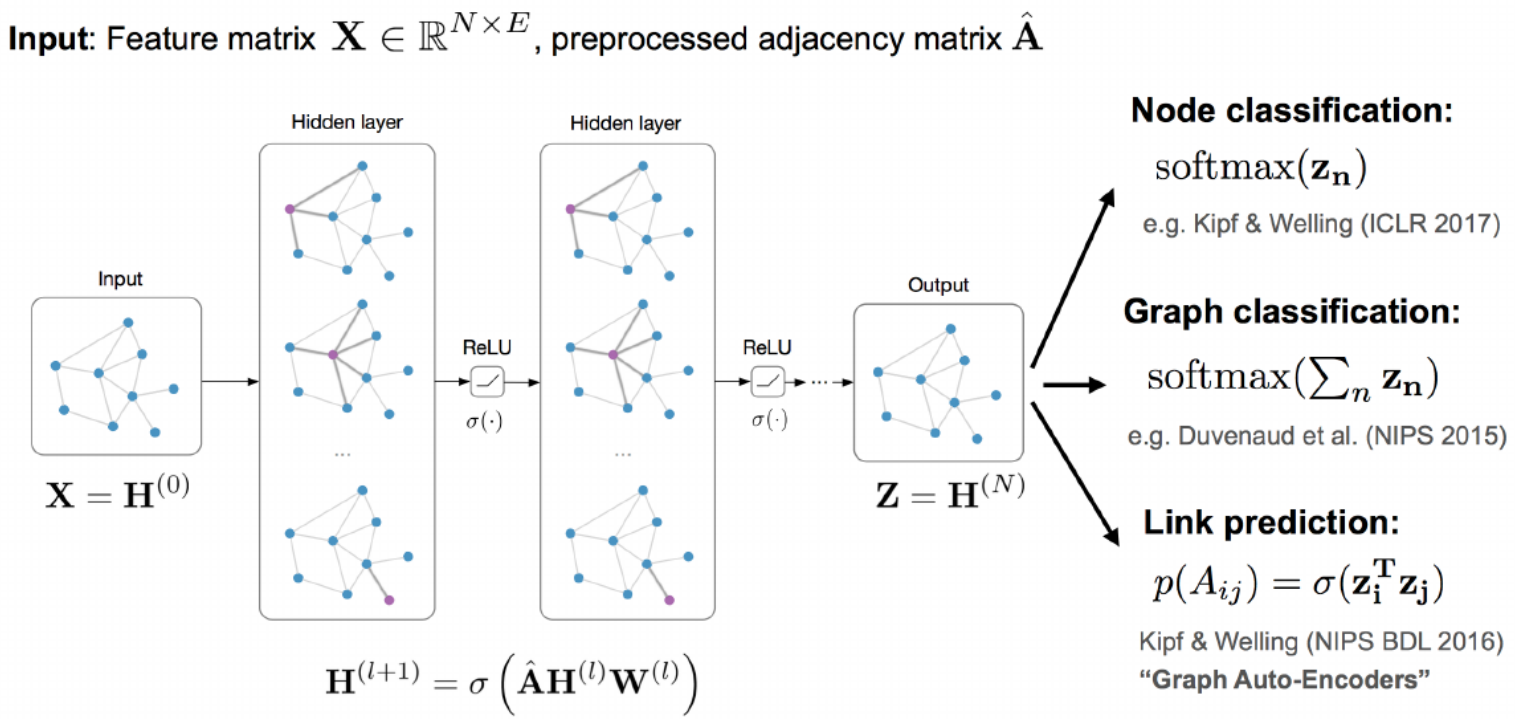


Figure 11: GCN usage (Credits: Thomas Kipf, University of Amsterdam)

Bibliography

Broadwater, K. & Stillman, N. (2023), *Graph Neural Networks in Action*, meap: version 8 edn, Manning Publishing Co. Ltd., Shelter Island, NY, USA.

Chollet, F. (2021), *Deep Learning with Python*, 2nd edn, Manning Publishing Co. Ltd., Shelter Island, NY, USA.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B. & Bharath, A. A. (2018), 'Generative adversarial networks: An overview', *IEEE Signal Processing Magazine* **35**(1), 53 – 65.

Géron, A. (2019), *Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O'Reilly Media, Inc., Boston, USA.

Goodfellow, I., Bengio, Y. & Courville, A. (2016), *Deep Learning*, MIT Press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014), Generative adversarial nets, *in* 'Proc. Advances Neural Information Processing Systems Conf', Montreal, Quebec, Canada, p. 2672–2680.