The Graph Neural Network: An Overview

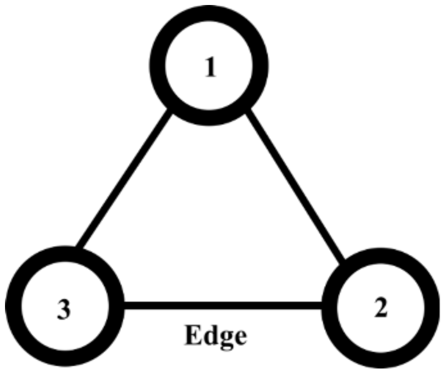
The recent advancement in neural network architecture has uplifted research on data mining and pattern recognition for many applications. The graph enabled Neural network architecture giving breakthrough in the research domains related to graphical analysis like face recognition, drug discovery, bioinformatics, social networks etc. GNN have become the standard toolkit for analyzing and learning from data on graphs.

The most fundamental part of GNN is a Graph.

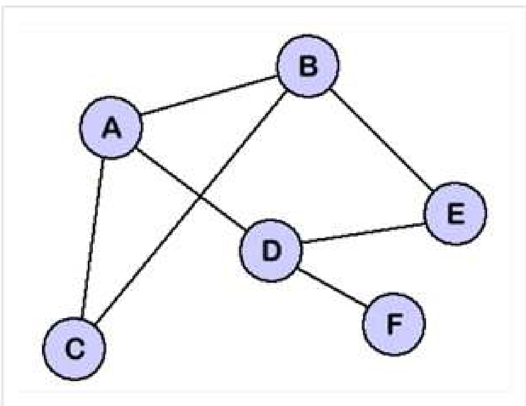
A graph consists of vertex (or node) attributes and edges (or links) attributes and directions entities) between a collection of entities (nodes) which is represented as:

$$G = (V, E)$$

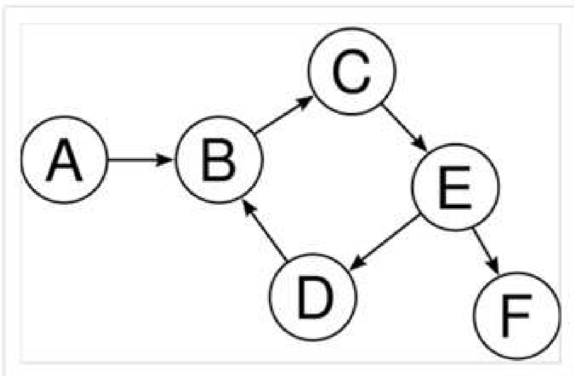
Where v is the set of nodes, and E are the edges between the nodes



If there are directional dependencies between nodes then edges are directed. If not, edges are undirected.



undirected graph



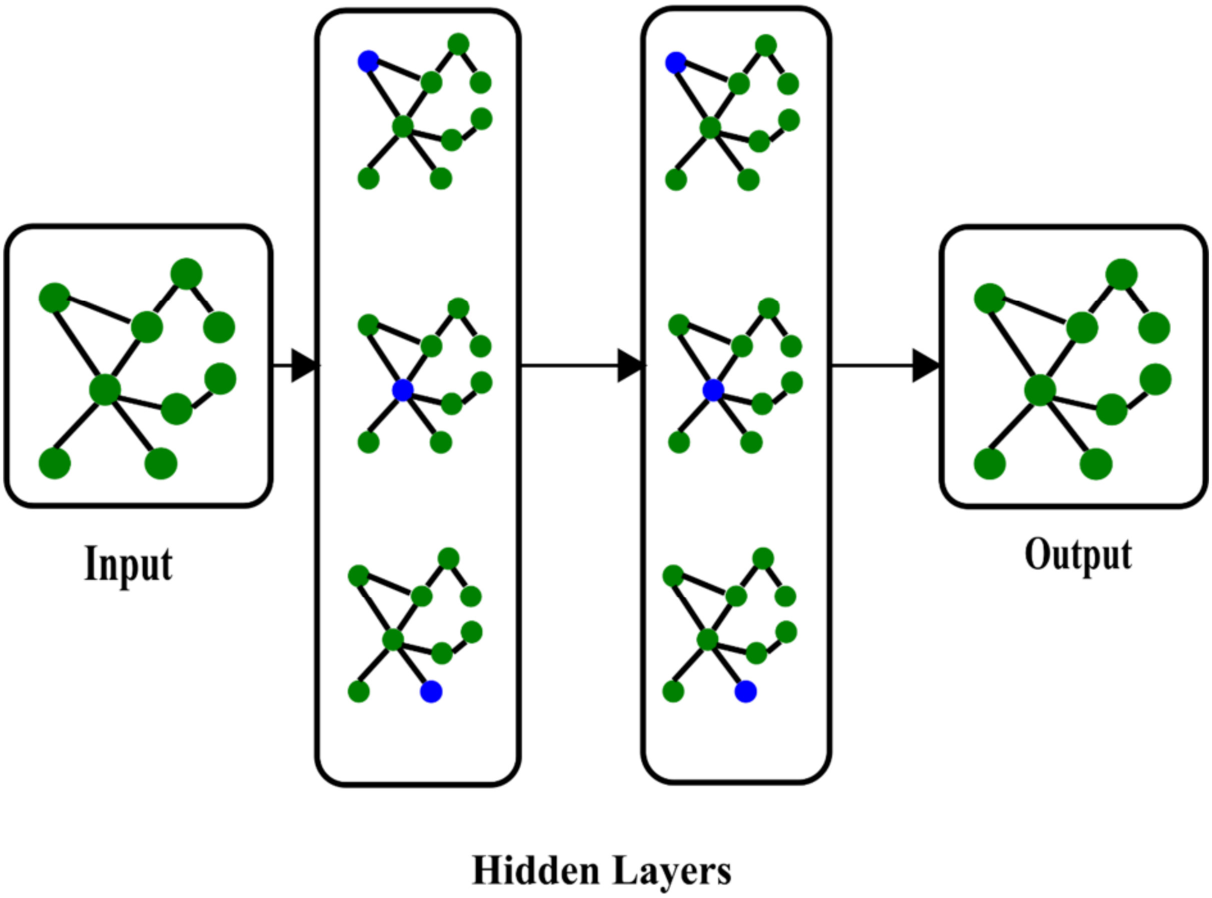
directed graph

Table I: Adjacency matrix representation of (i) undirected graph (ii) directed graph

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

	A	B	C	D	E	F
A	0	1	0	0	0	0
B	0	0	1	-1	0	0
C	0	-1	0	0	1	0
D	0	1	0	0	-1	0
E	0	0	-1	1	1	0
F	0	0	0	0	-1	0

Graph Neural Network



Graph Neural Network is a special type of Neural Network which operates on the structure Graph. The objectives of GNN is to classify the labelled nodes without knowing actual truth label of the nodes. and predict the links associated with it.

In the node classification problem,

Each node v has a feature x_v as well as a ground truth label t_v. In the context of a partially labelled graph G, the goal is to predict the labels of the unlabelled nodes based on these labelled nodes. The node is represented by a d-dimensional vector (state) h_v., which contains information about its neighbourhood.

$$h_v = f(x_v, x_{co[v]}, h_{ne[v]}, x_{ne[v]})$$

Where

$x_{co[v]}$ represents the features of the edge connected with node *v*

$h_{ne[v]}$ represents the embedding of its neighbourhood node *v*

$x_{ne[v]}$ represents the feature of the neighbourhood node *v*

f denotes the transition function which projects input into d- dimension vector space

to get the unique solution h_v we apply Banach fixed point theorem in iterative process. This continuous process is called as message passing or neighbourhood aggregation which is represented as

$$H^{t+1} = F(H^t, X)$$

here H and X represents the combinations of h and x respectively.

The out put of GNN is calculated by passing the state vector h_v and feature vector x_v through the output function *g*

$$o_v = g(h_v, x_v)$$

f and g are fully connected feed forward Neural networks. The loss function can be formulated as following:

$$Loss\ function, L = \sum_{i=1}^{n} (t_i - o_i)$$

GNN Applications

GNN have been explored in a wide range of applications across various domains like Chip Design, Combinatorial Optimization, Scene Reasoning, Recommendation system, Fake News Detection, Natural

Language Processing, Knowledge Graphs, Transportation, Autonomous Driving, Protein Folding & Drug Design, Energy Physics & Simulations, Traffic network etc.