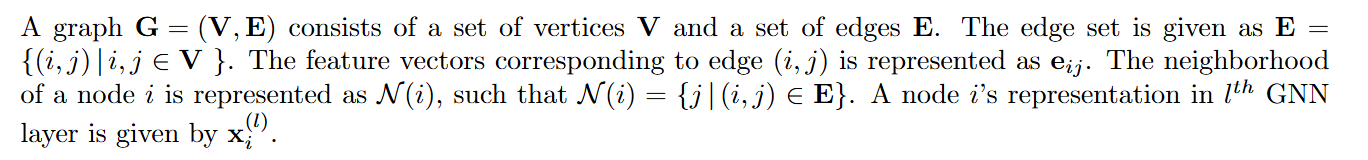
**DNN**

文本, 信件

描述已自动生成

1. **Input:** i(0) : initial data（normalization and preprocessing）, 特征数据feature data。（RSSI）
2. **Output:** o: result. Normally include two neurons x and y（The x and y coordinate values ​​of the output device）
3. **Hidden layers:** H : 每个隐藏层从上一层提取和转换数据特征。Extracting and transforming data features。
4. **Number of neurons:** Nh: 每层神经元个数代表计算能力，神经元是最基本的计算单位basic unit of calculation。每个神经元接受上一层的输入，利用激活函数activation function和权重矩阵进行处理、产生输出。
5. **Activation Function**激活函数：gh：线性输出转非线性Convert linear output to nonlinear output。（ReLU（Rectified Linear Unit），Sigmoid）
6. **Weight matrix**权重矩阵 : Wh: 第h层与前一层之间的连接权重，决定了输入信号如何影响当前层的神经元输出。
7. **Bias vector**偏置向量：bh：每一层都有对应的偏置向量，表示每个神经元的固定偏移量，用于调整神经元的激活值。偏置帮助网络在输入为零时仍能产生非零输出，从而增加模型的灵活性和学习能力。

# GNN



图表, 雷达图

描述已自动生成

https://seon.io/resources/dictionary/graph-neural-network-gnn/

GNNs use deep learning to reach conclusions.

1. **Input:** Graph structure, including nodes (each has their own characteristics: RSSI, location, device type…) and edges (connection relationship: signal strength and communication distance).
2. **Message Passing & Aggregation**: 传递nodes exchange information with their neighbor nodes through edges&Linear transformation via weight matrix, 聚合Update the feature representation of the node(求和，平均，最大)
3. **Graph Layers**：类似于DNN的隐藏层，each layer updates the representation of each node based on the features of its neighboring nodes.

图卷积（Graph Convolution）：GNN中的“卷积”操作将邻居节点的信息聚合到当前节点，并通过权重矩阵和激活函数更新节点的特征。

1. **Node Embedding Update**节点表示更新：The updating process is similar to feature extraction in DNN。但在GNN中，节点不仅依赖自身特征，还依赖于其邻居节点的信息。
2. **Output:** x,y
3. **Loss Function and Optimization:** GNN的训练也通过定义损失函数（如均方误差MSE），结合反向传播进行权重更新。通过迭代训练，模型的预测精度逐步提升。

# comparison

图示

描述已自动生成

|  |  |  |
| --- | --- | --- |
|  | DNN | GNN |
| INPUT | Feature data | Graph structured data |
| OUTPUT | Device location (x, y) | Node location(x,y) |
| EACH LAYER INFO | Transferred layer by layer, processed independently | Transferred between nodes, affected by neighboring nodes |
| HIDDEN LAYER | Neuron,  Activation function | Aggregate and update based on nodes and other nodes |
| NEURONS/NODES UPDATE | Weights and Biases | Neighbor node feature aggregation |
| WEIGHT MATRIX | The connection weights between layers | The connection weight between nodes (edge) |
| BIAS VECTOR | Each layer is independent, used to adjust the activation value | Normally no |
| ACTIVATION FUNCTION | ReLU, Sigmoid… |  |
| LOSS FUNCTION | MSE(Mean square error) | MSE, Node classification error |
| INFO UPDATE | Layer independent | in the graph |