Graph Neural Networks

Graph Convolutional Layers

Dr. Alireza Aghamohammadi

GCN Layer Variations

- **Basic GCN Layer:** The layer aggregates information from neighbors and combines it with the current node representation using a simple summation:

$$H_{k+1} = a\left[\beta_k \mathbf{1}^T + \Omega_k H_k (A + I)\right]$$

  where:

  - H_k: Node embeddings at layer k
  - A: Adjacency matrix
  - I: Identity matrix
  - β_k, Ω_k: Learnable parameters
  - $a(\cdot)$: Activation function

- **Diagonal Enhancement:** The current node embedding is weighted by a factor $(1 + \epsilon_k)$, where ϵ_k is a learned scalar:

$$H_{k+1} = a\left[\beta_k \mathbf{1}^T + \Omega_k H_k (A + (1 + \epsilon_k) I)\right]$$

- **Linear Transformation of Current Node:** Applies a distinct linear transform Γ_k to the current node representation:

$$H_{k+1} = a\left[\beta_k \mathbf{1}^T + \Omega_k H_k A + \Gamma_k H_k\right]$$

Alternative Aggregation Methods

- **Residual Connections:** The aggregated neighbor representation is transformed and combined with the current node embedding, often using concatenation:

$$H_{k+1} = \begin{bmatrix} a\left[\beta_k \mathbf{1}^T + \Omega_k H_k A\right] \\ H_k \end{bmatrix}$$

- **Average Aggregation:** Instead of summing neighbor embeddings, take their average:

$$H_{k+1} = a\left[\beta_k \mathbf{1}^T + \Omega_k H_k (AD^{-1} + I)\right]$$

  where:
  - D: Diagonal degree matrix (D_{ii} is the number of neighbors for node i)

- **Kipf Normalization:** Normalizes the sum of neighbor embeddings to down-weight nodes with many neighbors:

$$\mathsf{agg}\,[n] = \sum_{m \in \mathsf{ne}[n]} \frac{h_m}{\sqrt{|\mathsf{ne}[n]|\,|\mathsf{ne}[m]|}}$$

  In matrix form:

$$H_{k+1} = a\left[\beta_k \mathbf{1}^T + \Omega_k H_k (D^{-1/2} A D^{-1/2} + I)\right]$$

Max Pooling Aggregation

❖ **Max Pooling:** Aggregates the embeddings of neighbors by taking the element-wise maximum:

$$\mathsf{agg}\left[n\right] = \max_{m \in \mathsf{ne}[n]} \left[h_m\right]$$

where $\max[\bullet]$ denotes the element-wise maximum of neighbor embeddings h_m.