DL lab 6 – Graph Neural Networks

1. Upload the NetworkX jupyter notebook file (i.e.,NetworkX\_tutorial.ipynb) to google colab root directory.
   * Run the above code and understand it.
   * Complete the code sections to get the degree matrix and Laplacian matrix of the created random graph.
   * Calculate the graph density of the random graph in the code. Use the below equation (D = graph density, |V| = number of nodes and |E| = number of edges).
   * Increase the N value from 20 (original value) to 200 with multiple N values in between and observe the change of graph density and degree distribution (i.e., histogram plot). Explain what you observe and write the answer in a word file.



1. In the KarateClub dataset based GCN code, we use semi-supervised training approach along with the transductive leaning method.
   * Explain the differences between supervised learning, self-supervised learning and semi-supervised learning methods
   * Explain the differences between transductive learning and inductive learning.
2. Upload the KarateClub dataset based GCN jupyter notebook file (i.e., KarateClub\_GCN\_introduction.ipynb ) to google colab root directory.
   * In this code, we use Zachary’s karate club network dataset.
   * Run the above code and understand it.
   * Increase the number of epochs from 50 to 500 and observe the change in validation accuracy and write what you observe in the word file.
   * Experiment without self-loops added to GCNConv() layers in the GCN() model and detail the model accuracy increase/decrease in the word file.
   * Increase the number of GCNConv() layers in the GCN() model upto 8 layers from original 3 layers. Detail the accuracy increase/decrease in the word file.
     1. In\_channels and out\_channels in GCNConv() can be considered as hyper-parameters and you can use the best performing values you find.
     2. Add skip connections between some of the GCNConv() layers and try to see if that can improve the model performance.
     3. Detail what you observe in the word file.
3. Explain the differences between Message Passing GNN, graph convolution network (GCN), graph attention network (GAT) and GraphSAGE. Write the answers in the word file.

**Answer:**

**Differences Between Message Passing GNN, GCN, GAT, and GraphSAGE:**

1. **Message Passing GNN:**
   * **Concept:** The general framework for most GNN architectures. In Message Passing GNNs, nodes exchange information (messages) with their neighbors, aggregate the messages, and update their representations.
   * **Steps:**
     1. **Message generation** between neighboring nodes.
     2. **Aggregation** of received messages.
     3. **Update** of node embeddings based on aggregated information.
2. **Graph Convolution Network (GCN):**
   * **Concept:** A type of Message Passing GNN that extends the idea of convolution (from CNNs) to graphs. It aggregates information from neighbors and applies weights to the node’s features.
   * **Advantages:** Efficient for semi-supervised learning, especially when using large graphs with few labeled nodes.
   * **Limitation:** GCNs often face the over-smoothing problem when stacking many layers.
3. **Graph Attention Network (GAT):**
   * **Concept:** GATs introduce attention mechanisms to GCNs, where the importance of neighbors is learned rather than fixed.
   * **Mechanism:** Instead of treating all neighbors equally, GAT assigns a learnable weight (attention score) to each neighboring node.
   * **Advantages:** More flexible than GCNs as it learns how much focus each neighbor deserves, improving performance in heterogeneous or noisy graphs.
4. **GraphSAGE:**
   * **Concept:** GraphSAGE generates node embeddings by sampling and aggregating information from a fixed number of neighboring nodes. Unlike GCN, which processes the entire graph at once, GraphSAGE is inductive and works with unseen nodes.
   * **Advantages:** Scales better to large graphs and supports inductive learning (works with new, unseen nodes).
   * **Limitation:** Might not fully capture all graph information, as it only samples a subset of neighbors.

**Submission.**

Download the final modified notebook files (all 2 jupyter notebooks). Add these notebooks and the word file to a new zip file. Upload this zip file to the courseweb submission link. The file name should be your registration number.