DL Lab 6 – Graph Neural Networks

# 1. NetworkX Tutorial

1.1 Upload the NetworkX Jupyter notebook file (i.e., NetworkX\_tutorial.ipynb) to Google Colab root directory. Run the code and understand it.  
1.2 Complete the code sections to get the degree matrix and Laplacian matrix of the created random graph.  
1.3 Calculate the graph density of the random graph using the following formula:  
D = 2|E| / (|V|(|V|-1))  
Where |V| is the number of nodes and |E| is the number of edges.  
1.4 Increase the N value from 20 (original value) to 200 with multiple N values in between, observe the change of graph density and degree distribution (i.e., histogram plot), and explain the observations.

## Observations:

When increasing the number of nodes from 20 to 200 in a random graph, the graph density remains relatively consistent because the probability of edge formation stays the same regardless of the number of nodes. However, the degree distribution broadens as the number of nodes increases, leading to more diverse node degrees.

# 2. Karate Club GCN Code

2.1 Explain the differences between supervised learning, self-supervised learning, and semi-supervised learning methods:  
Supervised Learning: Uses labeled data to train the model (e.g., image classification with labels).  
Self-Supervised Learning: Model creates its own labels from the data (e.g., predicting the next word in a sentence).  
Semi-Supervised Learning: Combines a small amount of labeled data with a large amount of unlabeled data.

2.2 Explain the differences between transductive learning and inductive learning:  
Transductive Learning: Makes predictions only for known test data without generalizing to new data.  
Inductive Learning: Learns a general function that can be applied to unseen data.

## Experiment Observations:

2.3 Increased the number of epochs from 50 to 500. The validation accuracy steadily improved from 64% at 50 epochs to 82% at 500 epochs, indicating consistent learning without overfitting.

2.4 Experimented without self-loops in GCNConv() layers. Removing self-loops resulted in a drop in validation accuracy, with a larger impact at earlier epochs. Self-loops allow nodes to preserve their features, improving generalization.

2.5 Increased the number of GCNConv() layers to 8. Adding more layers increases the model's depth but can lead to issues like over-smoothing if the channel count isn't adjusted properly. Adding skip connections mitigated these issues by preserving information from earlier layers.

# 3. Differences Between GNN Variants

Message Passing GNN: Focuses on propagating information between nodes through their edges. It emphasizes local information flow.  
Graph Convolution Network (GCN): A specialized form of GNN that generalizes the convolutional operation to graph structures.  
Graph Attention Network (GAT): Enhances GCNs by assigning different importance (attention scores) to different neighboring nodes.  
GraphSAGE: A scalable GNN that samples and aggregates features from a node's neighbors to make predictions.

Git Clone Link - [ThenukaAnjala/DL-lab-6 (github.com)](https://github.com/ThenukaAnjala/DL-lab-6)