**Deep Learning Lab 06**

**1. Supervised Learning, Self-Supervised Learning, and Semi-Supervised Learning**

* **Supervised Learning:** This is a type of learning where the models are trained on a labeled dataset. The input corresponds to an output label. This is aimed at learning from such labeled data to obtain a function mapping inputs to the correct outputs.
* **Self-Supervised Learning:** This resembles the approach used in supervised learning, except that it does not require explicit labels. The labels come directly from the data. For instance, it may learn to predict parts of the data from other parts and generate labels internally.
* **Semi-supervised Learning:** Training is done on both labeled and unlabeled data. The model learns on a small dataset and then takes advantage of unlabeled data for better generalization.

**2. Transductive Learning Vs Inductive Learning**

* **Transductive Learning:** In transductive learning, the model is directly trained to predict on a specific test set provided. It uses both training and test data during training but will not generalize to unseen data.
* **Inductive Learning:** In inductive learning, the model has to learn to generalize on unseen data. In training, the model has not seen the test data and tries to learn from the provided training data to come up with patterns that could be applied to new, unseen instances.

**3. Karate Club Dataset GCN Code Adaptation**

**Upload Notebook in Google Colab:** You can easily upload the Jupyter notebook by just drag-and-drop in the interface of Google Colab.

**Experiments:**

1. **Increase Epochs:** The number of epochs is increased from 50 to 500, and observe the change in validation accuracy. Record your observations in a Word file.
2. **Remove Self-Loops in GCNConv**: Apply a modification in the GCN model where there are no self-loops taken in the GCNConv() layers. See the variation in accuracy caused by doing so.
3. **Increase GCNConv Layers**: Increase the number of GCNConv() layers from 3 to 8 make some tuning in in channels and out channels, and note the effect that this causes on model performance.
4. **Residual Connections:** Introduce residual connections between some of these GCNConv() layers and examine their effect on improving performance.

**4. Message Passing GNN, GCN, GAT, and GraphSAGE**

* **MPNN - Message Passing GNN:** A generalization for many GNN models that update node representations through iterative message passing between nodes along the edges in the graph and aggregating information.
* **Graph Convolution Network (GCN):** A GNN where nodes aggregate the information from neighbors by convolutional operation typically based on graph Laplacians.
* **Graph Attention Network:** GAT extends GCN by applying attention mechanisms to scale the importance of neighboring nodes within an aggregation step.
* **GraphSAGE**: It uses the concept of neighborhood sampling and aggregation to generate embeddings. This introduces scalability by considering only a subsection of neighbors for training.