DL lab 8 – Graph Neural Networks

1. 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.



* **Graph Density**: Graph density decreases as N increases due to a fixed number of edges.
* **Degree Distribution**: Degree distribution becomes more heterogeneous, with a broader range of node degrees.
* **Histogram Plot**: The histogram plot of the degree distribution widens, showing a long-tailed distribution.
* **Sparsity**: The graph becomes sparser, with fewer connections relative to possible connections.

1. In the KarateClub dataset based GCN code, we use semi-supervised training approach along with the transudative leaning method.
2. Explain the differences between supervised learning, self-supervised learning and semi-supervised learning methods

* **Supervised Learning:** This method involves training a model on labelled data so that it may learn to classify or predict things based on the features that are input. For every case, a precisely defined target variable is needed.
* **Self-Supervised Learning:** Self-supervised learning is an unsupervised learning method in which the model infers tasks or labels from the data itself. It gains knowledge by making predictions about certain portions of the data based on other portions, like filling in the blanks in a phrase.
* **Semi-Supervised Learning:** Labelled and unlabeled data are combined for training in semi-supervised learning. It makes use of a greater quantity of unlabeled data in conjunction with a smaller amount of labelled data, which might enhance model performance.

1. Explain the differences between transductive learning and inductive learning.

* **Transductive Learning:** Predicting based on certain, known data points in the dataset is known as transductive learning. It is centred on determining the labels or values of data points that belong to the same training dataset.
* **Inductive Learning:** Making predictions on fresh, unknown data by extrapolating from the training set is known as inductive learning. It seeks to develop a model capable of handling examples never seen before and extrapolating its knowledge to new situations.

1. 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.

* **Monitoring for potential overfitting is also important when the number of training epochs is increased from 50 to 500. During the extended training period, you can expect the validation accuracy to generally increase, but after a certain point, the accuracy may stabilize or exhibit fluctuations.**

1. Experiment without self-loops added to GCNConv() layers in the GCN() model and detail the model accuracy increase/decrease in the word file.

* **When testing without self-loops in GCNConv() layers, a loss of self-connections that could provide useful information could result in a little decline in the model's accuracy. Self-loops can help nodes keep their own properties. The accuracy decline could not be as significant, though, and other training enhancements might more than make up for it.**

1. 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.

* **By raising the GCNConv() layers to 8, the GCN() model's capacity can be significantly increased. There is a potential that it will overfit as well due to the added complexity. Carefully choosing the in\_channels and out\_channels for GCNConv() will maximize performance. Skip links between layers may be added to lessen overfitting and boost performance.**

1. Explain the differences between Message Passing GNN, graph convolution network (GCN), graph attention network (GAT) and GraphSAGE. Write the answers in the word file.

* **Message-Passing Graph Neural Network:** GNNs are generally framed within neural networks. They use message-passing to transfer and compile data among nodes in a graph. They support a wide range of neighbourhood definitions and aggregation functions and are quite adaptable.
* **Graph Convolution Network (GCN):** A subset of GNNs, GCNs combine node features by adding the weighted sum of their neighbouring nodes' features. It has demonstrated good performance in a variety of workloads and is based on a reduced convolution technique.
* **Graph Attention Network (GAT):** GAT is a different kind of GNN that weighs the properties of neighbour nodes according to their importance using attention processes. This is particularly helpful for capturing more intricate links in the graph since it enables nodes to selectively attend to their neighbours.
* **GraphSAGE:** A GNN model called GraphSAGE (Graph Sample and Aggregated) creates embeddings for nodes by gathering and combining characteristics from nearby nodes. It can handle inductive learning tasks, where nodes not observed during training require embeddings, and is especially helpful for big graphs.