Ben KABONGO B.
ben.kabongo_buzangu@ens-paris-saclay.fr

Lab session # 6
ALTEGRAD 2023

11/22/23

1 Question 1

The law of degree of a node of an Erdos-Renyi random graphs $G(n,p)$ is given by the binomial $\mathbf{B}(n-1,p)$. This is because each node can be connected with probability p to the other n-1 in the graph independently. The number of nodes is therefore the expectation of the binomial $\mathbf{B}(n-1,p)$, which is $(n-1)\times p$.
Let $n = 25$

- **p = 0.2** : The expected degree is 24 x 0.2 = 4.8 (= 5).
- **p = 0.4** : The expected degree is 24 x 0.4 = 9.6 (= 10).

2 Question 2

Let's take a quick look at the dimensions of the matrices manipulated in the network.
Let n be the sum of the number of nodes in all the graphs in a batch and d the number of features.
Let n_{graph} be the number of graphs and n_{class} the number of classes.
We have that :

$$A \in \mathbb{R}^{n\times n} \qquad X \in \mathbb{R}^{n\times d}$$
$$W^0 \in \mathbb{R}^{d\times h_1} \qquad W^1 \in \mathbb{R}^{h_1\times h_2}$$
$$W^2 \in \mathbb{R}^{h_2\times h_3} \qquad W^3 \in \mathbb{R}^{h_3\times n_{class}}$$

We therefore have that :

$$Z^0 \in \mathbb{R}^{n\times h_1} \qquad Z^1 \in \mathbb{R}^{n\times h_2}$$
$$Z_G \in \mathbb{R}^{n_{graphs}\times h_2} \qquad y \in \mathbb{R}^{n_{graphs}\times n_{class}}$$

The READOUT operator therefore takes the output Z^1 as its input to produce the matrix Z_G, by summing over the first dimension the rows of Z^1 associated with the same graph.
The trainable parameters can be used successively on each graph individually, but this would mean re-slicing the A matrix to find the adjacency matrices of each graph and also calculating each intermediate output individually per graph. This would make parameter updating tedious, as each operation would have to be taken into account individually.
It is also more difficult to filter a matrix product to retain only the results of interest.

3 Question 3

3.1 *neighborhood aggregation*=mean, *readout*=mean

All values in the output tensor seem to be relatively small and positive. The result appears to be stable across rows and columns, indicating a consistent application of mean aggregation both for neighbor aggregation and *readout*.

3.2 *neighborhood aggregation*=mean, *readout*=sum

The values in the output tensor are more varied, including both positive and negative values. The sum operation in the *readout* phase allows for a wider range of values, and the output is influenced by both the mean-aggregated node features and the graph structure.

3.3 *neighborhood aggregation*=sum, *readout*=mean

Similar to the first configuration (*neighborhood aggregation*=mean, *readout*=mean), the values are positive, and there seems to be a certain stability across rows and columns. Using the sum operation for neighbor aggregation and mean operation for *readout* still results in a relatively stable pattern.

3.4 *neighborhood aggregation*=sum, *readout*=sum

The values in the output tensor are more varied and have both positive and negative values. The combination of sum operations in both neighbor aggregation and *readout* allows for a broader range of influences from the node features and graph structure, leading to more diverse values.

3.5 General Observations

The choice of aggregation function (mean or sum) in both the neighbor aggregation and *readout* phases significantly impacts the nature of the output. Using the sum operation tends to result in a wider range of values, allowing for a more expressive representation that captures the influence of both node features and graph structure. The specific values in the tensors are not discussed, as the focus is on the overall patterns and trends across configurations.

4 Question 4

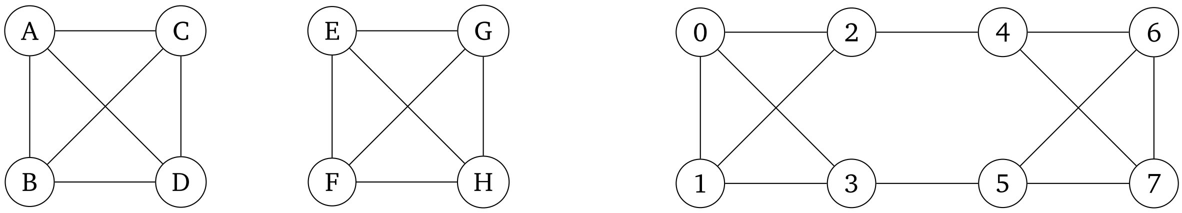

Figure 1: G_3 on left, G_4 on right

The graphs G_3 and G_4 below are not isomorphic but are indistinguishable with our GNN network with the configuration *neighborhood aggregation*=sum and *readout*=sum.