Axel Dumont
axel.dumont@ensta-paris.fr

1 Question 1

For any particular vertex, the number of edges (degree) follows the law $\mathcal{B}(n-1,p)$, then the expexted value is :

$$E = (n-1)p$$

Hence for $n = 25$ and $p = 0.2$, $E_1 = 24 \times 0.2 = 4.8$.
For $n = 25$ and $p = 0.4$, $E_2 = 24 \times 0.4 = 9.6$

2 Question 2

The sum or mean operation is commonly used as a readout function because it aggregates the node-level representations into a graph-level representation. This aggregation is essential for capturing the global structure and characteristics of the graph. On the other hand, fully connected layers do not naturally provide a mechanism for aggregating information from all nodes in the graph.
Additionally, because graphs can have variable size, a linear readout would require fixing the size of the input representation. But an aggregation function like sum/mean works naturally on representations from graphs with varying numbers of nodes.

3 Question 3

```
tensor([[-0.0048,  0.0633,  0.2819, -0.5014],
        [-0.0048,  0.0633,  0.2819, -0.5014],
        [-0.0048,  0.0633,  0.2819, -0.5014],
        [-0.0048,  0.0633,  0.2819, -0.5014],
        [-0.0048,  0.0633,  0.2819, -0.5014],
        [-0.0048,  0.0633,  0.2819, -0.5014],
        [-0.0048,  0.0633,  0.2819, -0.5014],
        [-0.0048,  0.0633,  0.2819, -0.5014],
        [-0.0048,  0.0633,  0.2819, -0.5014],
        [-0.0048,  0.0633,  0.2819, -0.5014]], device='cuda:0',
       grad_fn=<AddmmBackward0>)
tensor([[-1.5158, -1.1768, -1.7710, -1.4911],
        [-1.6779, -1.3031, -1.9553, -1.6533],
        [-1.8401, -1.4295, -2.1397, -1.8154],
        [-2.0022, -1.5559, -2.3240, -1.9776],
        [-2.1643, -1.6822, -2.5084, -2.1398],
        [-2.3264, -1.8086, -2.6927, -2.3019],
        [-2.4886, -1.9350, -2.8771, -2.4641],
        [-2.6507, -2.0613, -3.0614, -2.6262],
        [-2.8128, -2.1877, -3.2457, -2.7884],
        [-2.9749, -2.3141, -3.4301, -2.9506]], device='cuda:0',
       grad_fn=<AddmmBackward0>)
tensor([[-0.8076, -0.3181,  0.1950, -1.3124],
        [-0.8076, -0.3181,  0.1950, -1.3124],
        [-0.8076, -0.3181,  0.1950, -1.3124],
        [-0.8076, -0.3181,  0.1950, -1.3124],
        [-0.8076, -0.3181,  0.1950, -1.3124],
        [-0.8076, -0.3181,  0.1950, -1.3124],
        [-0.8076, -0.3181,  0.1950, -1.3124],
        [-0.8076, -0.3181,  0.1950, -1.3124],
        [-0.8076, -0.3181,  0.1950, -1.3124],
        [-0.8076, -0.3181,  0.1950, -1.3124]], device='cuda:0',
       grad_fn=<AddmmBackward0>)
tensor([[ 0.1384, -2.0071, -0.6001, -4.0912],
        [ 0.1653, -2.1968, -0.6553, -4.4829],
        [ 0.1923, -2.3865, -0.7105, -4.8746],
        [ 0.2192, -2.5763, -0.7657, -5.2663],
        [ 0.2462, -2.7660, -0.8209, -5.6580],
        [ 0.2732, -2.9557, -0.8761, -6.0497],
        [ 0.3001, -3.1455, -0.9313, -6.4414],
        [ 0.3271, -3.3352, -0.9865, -6.8331],
        [ 0.3540, -3.5250, -1.0417, -7.2248],
        [ 0.3810, -3.7147, -1.0969, -7.6165]], device='cuda:0',
       grad_fn=<AddmmBackward0>)
```

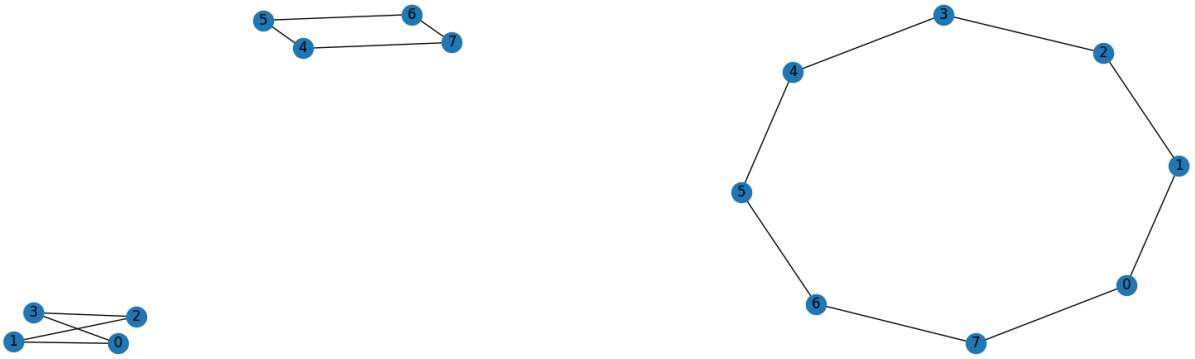
Figure 1: Output of Task 8

Based on the printed output, we observe that:

- When using mean aggregation and mean readout, the GNN generates the exact same representation for all graphs. This shows it is unable to distinguish between graphs of different sizes.
- When using sum aggregation and sum readout, the representation is different for graphs of different sizes. However, the difference seems to solely depend on the number of nodes, not the specific graph structure.
- Using sum aggregation and mean readout leads to the same representation for all graphs, showing again an inability to distinguish graph structures.
- With mean aggregation and sum readout, the representations capture differences in the number of nodes, but not the specific connectivity patterns.

Overall, none of these GNN variants are able to distinguish structural differences between the cycle graphs based on connectivity patterns, number of components etc. The representations are either identical, or only capture trivial differences in the number of nodes. This demonstrates limitations in the expressive power of such standard GNN architectures

4 Question 4



(a) Graph 1 (b) Graph 2

Figure 2: Two graphs non differentiable with GNN

These graphs have the same number of nodes and edges. But Graph 1 consists of two connected components while Graph 2 is a single connected component.
However, with sum aggregation and sum readout, the GNN representation only depends on the degree of each node. Since all the nodes have the same degree in both graphs, the GNN will produce an identical representation for both Graph 1 and Graph 2.