Axel Dumont
axel.dumont@ensta-paris.fr

Lab session # 7
ALTEGRAD 2023

12/05/23

1 Question 1

An LSTM is not a permutation invariant model. An LSTM processes input sequences in order, capturing sequential dependencies in the data. As such, the output of an LSTM depends on the order of inputs.
In contrast, sets consist of unordered elements without a meaningful sequence. Using an LSTM on set data does not allow the model to leverage properties of sets, where order should not matter.
Therefore, I would not recommend using LSTMs for modeling set data. Other neural network architectures that natively encode permutation invariance, such as Deep Sets or transformers with set attention, are better suited for sets. These models can fully exploit the unordered, permutation-invariant attributes of sets for optimal performance.

2 Question 2

Graph neural networks (GNNs) and DeepSets are two neural network architectures designed for set-based and graph-based data respectively. GNNs leverage the graph structure by using message passing to aggregate neighborhood information at each node and update node representations. In contrast, DeepSets rely on permutation invariant architectures and aggregation functions that can process the entire set at once. While GNNs are optimized for graphs with features at both nodes and edges, DeepSets are built for set data where order does not matter. Despite these differences, both methods aim to generate embeddings that can be used for set-level and graph-level prediction tasks rather than at the individual element level.
A set and a graph without edges share core similarities in that they both represent an unordered collection of distinct elements. The key difference is that a set consists only of independent elements with no relationships whatsoever. In contrast, a graph is defined in terms of nodes and edges, so even without explicit edge connections, the representation still implies potential relationships between nodes. While a graph without edges behaves like a set in terms of lacking predefined structure, the node-based abstraction means that connections and relationships can be later added in graph form. Thus, while similar in the absence of ordering and links, a set is more rigidly restricted to independent elements, while a graph without edges retains potential relation between its nodes.

3 Question 3

Homophilic graph

$$P_{hom} = \begin{bmatrix} 0.8 & 0.05 \\ 0.05 & 0.8 \end{bmatrix}$$

Heterophilic graph

$$P_{het} = \begin{bmatrix} 0.05 & 0.8 \\ 0.8 & 0.05 \end{bmatrix}$$

Calculations of expected value for an edge between different blocks

We can distinguish in a graph with 4 blocks $\binom{4}{2} = 6$ different pairs of blocks.
Knowing that there is $5 \times 5 = 25$ possible edges between two different blocks. The probability of each node existing being 0.05, the expected value of the number of edges between two different blocks is :

$$E = 6 \times 25 \times 0.05 = 7.5$$

4 Question 4

For weighted graph reconstruction with adjacency matrix values not constrained to 0 or 1, I would propose using a mean squared error (MSE) loss instead.

The MSE loss compares the predicted continuous weighted edge values to the true weighted edge values and computes the average squared difference between them. It does not constrain the predictions or ground truth values, allowing the model to predict non-binary edge weights.
Specifically, with a predicted weighted adjacency matrix Â and true weighted adjacency matrix A, each containing edge weights a_{ij}, the MSE loss would be:

$$L(A, \hat{A}) = \frac{1}{|V|^2} \sum_{i,j=1}^{|V|,|V|} (a_{ij} - \hat{a}_{ij})^2$$

By removing the binary constraint on edge weights and moving to an MSE-based comparison of the full range of predicted and true values, the model can more effectively reconstruct graphs where weighted edges represent connection strengths or capacities rather than solely presence/absence status.
The continuous nature of the loss and ground-truth representations better aligns with capturing nuanced relationships in weighted graphs across domains.