Kyllian Asselin de Beauville
kyllian.asselin-de-beauville@polytechnique.edu



# 1 Question 1

Long Short-Term Memory (LSTM) [1] networks are a type of recurrent neural network (RNN) specifically designed to capture long-range dependencies and avoid the vanishing gradient problem commonly encountered in standard RNNs. However, LSTMs are not inherently permutation invariant. The order of input sequences is significant in LSTMs, as they process data sequentially. This characteristic implies that changing the order of elements in the input sequence can result in different representations and outputs.

In the context of sets, which are fundamentally unordered collections, the inherent sequence processing nature of LSTMs does not align well with the permutation invariance property. Sets do not have a specific order of elements, and any model processing sets should ideally produce the same output regardless of the order of elements in the set.

Considering this, we do not recommend using LSTMs for set-based data unless the data can be meaningfully ordered in a consistent manner that contributes to the learning task.

# 2 Question 2

Graph Neural Networks (GNNs) [4] and the DeepSets [3] architecture, while both designed for non-Euclidean data, differ significantly in their architectural design and the types of data they are intended to process.

- **Architectural Differences:** The primary difference lies in their handling of relationships between data elements. GNNs are specifically designed to handle graph-structured data. They incorporate information about node connectivity and edge attributes through message passing mechanisms. In a GNN, node representations are updated iteratively by aggregating information from their neighbors, considering both node features and the graph topology. This process effectively allows GNNs to capture the structural information inherent in graphs.

  In contrast, the DeepSets architecture is designed to process sets, which are unordered collections of elements without explicit relational information. DeepSets treat each element of the set independently, applying a shared function to each element and then aggregating the results using a permutation-invariant function, such as summation. This approach respects the fundamental property of sets, permutation invariance, where the order of elements does not influence the output.

- **Sets vs. Graphs without Edges:** While a set in the context of DeepSets and a graph without edges may appear similar, they represent different conceptual models. A set is a collection of distinct elements without any inherent ordering or relational structure. In DeepSets, elements are processed independently, and their features are aggregated without considering any relationships between them.

  An edgeless graph, on the other hand, can be thought of as a special case of a graph where all nodes are isolated. While it lacks edges, GNNs can still process it by considering the potential edges and node features. In practice, processing a graph without edges with a GNN would still involve mechanisms designed for graph data, but without edge-related computations. However, this also highlights the limitations of GNNs in such scenarios, as the absence of edges prevents the use of their full capabilities to capture relational structures and interactions between nodes.

In conclusion, while both GNNs and DeepSets can process collections of elements, they are tailored for fundamentally different data structures — GNNs for graphs with relational data and DeepSets for sets where elements lack explicit relationships.

# 3 Question 3

The stochastic block model (SBM) [2] is a generative model for graphs in which nodes are partitioned into communities, and edges are randomly generated based on an edge probability matrix $P$.

## 3.1 Edge Probability Matrices

For an SBM with $r = 2$ (two communities), we can define the edge probability matrices for homophilic and heterophilic structures as follows:

- **Homophilic Structure:** Nodes within the same community are more likely to be connected. A suitable probability matrix $P$ could be:
$$P_{\text{homophilic}} = \begin{bmatrix} 0.8 & 0.05 \\ 0.05 & 0.8 \end{bmatrix} \tag{1}$$
- **Heterophilic Structure:** Nodes are more likely to connect with nodes from the other community. A suitable probability matrix $P$ could be:
$$P_{\text{heterophilic}} = \begin{bmatrix} 0.05 & 0.8 \\ 0.8 & 0.05 \end{bmatrix} \tag{2}$$

## 3.2 Expected Number of Edges Between Nodes in Different Blocks

In an SBM with $n = 20$ nodes, partitioned into 4 blocks of 5 nodes each, and given the edge probability matrix $P$ where diagonal elements are 0.8 and off-diagonal elements are 0.05, the expected number of edges between nodes in different blocks can be calculated as follows:

Let $n_i$ be the number of nodes in block $i$. Then, the expected number of edges between two different blocks $i$ and $j$ (where $i \neq j$) is given by:
$$E_{ij} = n_i \times n_j \times P_{ij} \tag{3}$$

Since each block has 5 nodes, we have $n_i = 5$ for all $i$. The off-diagonal elements of $P$ are 0.05, so $P_{ij} = 0.05$ for $i \neq j$. Therefore, for each pair of different blocks:
$$E_{ij} = 5 \times 5 \times 0.05 = 1.25 \tag{4}$$

There are $\binom{4}{2} = 6$ unique pairs of different blocks. Hence, the total expected number of edges between different blocks is:
$$E_{\text{total}} = 6 \times E_{ij} = 6 \times 1.25 = 7.5 \tag{5}$$

# 4 Question 4

For weighted graphs where adjacency matrix entries represent edge weights and can take on a continuous range of values, the binary cross entropy loss used for unweighted graphs is not the most suitable choice. Instead, a loss function that can handle the continuous nature and the magnitude of the weights should be used. One such loss function is the Mean Squared Error (MSE) loss, defined as:

$$\mathcal{L} = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} (A_{ij} - \hat{A}_{ij})^2 \tag{6}$$

The MSE loss measures the average squared difference between the actual values $A_{ij}$ and the estimated values $\hat{A}_{ij}$, making it sensitive to the magnitude of the edge weights and thus more appropriate for weighted graph

reconstruction tasks. This loss function ensures that the model is penalized for the difference between the true weight and the predicted weight of each edge, reflecting the performance in reconstructing the weighted connectivity structure of the graph.

## References


[1] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[2] Clement Lee and Darren J. Wilkinson. A review of stochastic block models and extensions for graph clustering. *Applied Network Science*, 4(1), December 2019.

[3] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets, 2018.

[4] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications, 2021.