Martin BRICENO
jbriceno@telecom-paris.fr

Lab session # 7
ALTEGRAD 2020

02/02/21

1 Question 1

In the case implemented in the laboratory, all sets having the same sum of elements must have the same representation in order to predict the same sum. This is because our objective is a regression with loss MAE. In this context, if we feed our model with the sets in Figure 2, each set would have the same representation since the sum of each is [0,0].

However, if we modify our model to perform a **multi-class classification as in the case of Points Cloud, we may force the model to output a different representation for each set since we would consider each set to be a different class**.

Another way to obtain different representations for each set is to **predict the variance of each set instead of the sum of its elements, since this is different for each set, we will necessarily obtain different representations of these**.

2 Question 2

In the graph classification tasks we have seen, the task of finding a representation of the graphs is important and even the first task to deal with before performing the classification.

Since graphs have variable sizes and their nodes are sets where to apply operations, these need to be invariant, then **applying DeepSets on graphs to obtain representations of these could be very useful in the classification task indicated above**.

Consequently, within the architecture of a graph neural network, **DeepSets would consist of an initial sub-module where the input would be a graph and its output would be the embedding representation of it**, which we will use to perform the classification.

3 Question 3

Consider that we will calculate the influence spread of a set S of k nodes in a network of n nodes.

To calculate the exact influence spread of this set S we must consider **all possible combinations of k-node subgraphs within our n-node graph** to determine which of these subgraphs will maximize the influence spread.

This calculation is determined as follows:

$$nb\_subgraph = \binom{n}{k} \tag{1}$$

4 Question 4

The greedy algorithm ensures to compute the set that maximizes the influece spread since it relies on the properties of monotonicity and sub-modularity. This is true even in the approximate case using monte carlo since as many iterations are performed, this method will converge to the exact value. So **any other way of constituting a set that maximizes the influence spread can at most reach the accuracy of the gready algorithm but not surpass it**.

Thus, if we build sets based on some central measure such as degree, they have a high chance of being made up of nodes that do not contribute as much to the influence spread as the nodes obtained by greedy.

If we consider $g_1, g_2...g_k$ the nodes obtained successively by greedy and $d_1, d_2...d_k$ the nodes obtained successively under some centrality measure.

Since the nodes obtained by greedy are the ones that maximize the influence spread in each iteration, if any of the nodes obtained by the second method is different from the first nodes, then we will obtain a smaller spread according to the properties of monotonicity and submodularity. Then in general the influence spread of the latter will be lower than the one obtained by greedy.