KALLEL Mahdi
mahdi.kallel@telecom-paris.fr

1 Question 1

In our current implementation we use an mean square error coupled with negative sampling to predict whether a link exists or not. Although we use this scheme for classification (target is either 0 or 1). This same architecture and without change can be used for a weighted graph where the target will be a real number corresponding to the link weight, in which case we will be dealing with a regression task.

2 Question 2

For this task we propose to, instead of outputting a single "Z" matrix, we output a second one as well that we will call "D". If X is of dimensions (n,p) and Z of dimension (n,m) , "D" should be of dimension (m,l) such that Z * D = $\hat{X}$. Matrix multiplication being a differentiable operation, this would allow our network to find node embeddings "Z" from which we can reconstruct both node characteristics and the adjacency matrix.

3 Question 3

$$Z = \begin{bmatrix} 0.77 & -1.26 & -0.63 \\ 1.15 & -1.90 & -0.94 \\ 0.77 & -1.26 & -0.63 \\ 1.15 & -1.90 & -0.94 \\ 1.15 & -1.90 & -0.94 \\ 1.15 & -1.90 & -0.94 \\ 1.15 & -1.90 & -0.94 \\ 0.77 & -1.26 & -0.63 \\ 0.77 & -1.26 & -0.63 \end{bmatrix}$$

SUM : $\implies \begin{bmatrix} 2.69 & -3.42 & -2.2 \\ 4.6 & -7.6 & -3.76 \\ 1.54 & -2.52 & -1.26 \end{bmatrix}$

MEAN : $\implies \begin{bmatrix} 1.23 & -1.14 & -0.733 \\ 1.15 & -1.9 & -0.94 \\ 0.77 & -1.26 & -0.63 \end{bmatrix}$

MAX : $\implies \begin{bmatrix} 1.15 & -1.26 & -0.63 \\ 1.15 & -1.9 & -0.94 \\ 0.77 & -1.26 & -0.63 \end{bmatrix}$

4 Question 4

In both G1 and G2 every node is connected to exactly 2 other nodes. And each node i has the same initial representation. $X_i = X_j \, \forall (i,j)$ let's denote by Z_i^t the embedding of node i after t message passing layers.
By this definition f $Z_i^0 = Z_j^0 \, \forall (i,j)$ Let's prove by induction that this is the case $\forall t$.
Suppose that for t=T $Z_i^T = Z_j^T = Z^T \, \forall (i,j)$:

$$Z_i^{T+1} = ReLU(A_t * \sum_{j \in N(i)} Z_j^T) = ReLU(A_t * 2Z^T) = Z_j^{T+1}$$

$$Z_{g_1} = \sum_{i \in G_1} Z_i^4 = 6 * Z^4 = Z_{g_2}$$

5 Plots

:
We report our plots in the following section.

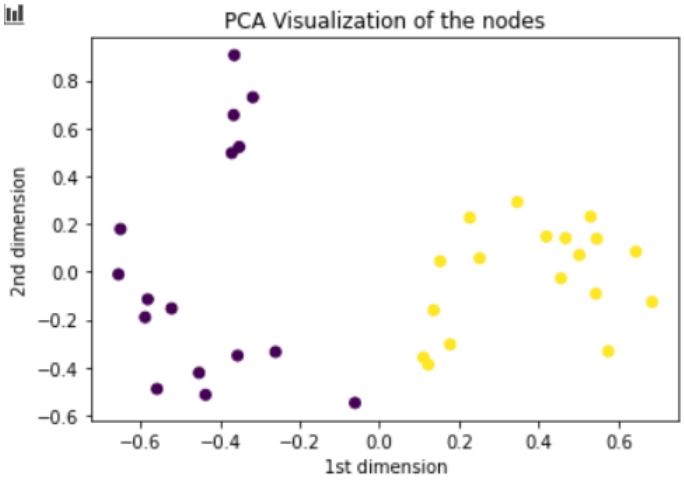

Figure 1: GNN VAE Embeddings of nodes of different classes.

```
Epoch: 0001 loss_train: 11.2314 acc_train: 0.4222 time: 0.1905s
Epoch: 0011 loss_train: 0.5314 acc_train: 0.7222 time: 0.1418s
Epoch: 0021 loss_train: 0.5518 acc_train: 0.7556 time: 0.1433s
Epoch: 0031 loss_train: 0.5629 acc_train: 0.6778 time: 0.1642s
Epoch: 0041 loss_train: 0.5398 acc_train: 0.7111 time: 0.1391s
Epoch: 0051 loss_train: 0.4920 acc_train: 0.7444 time: 0.1685s
Epoch: 0061 loss_train: 0.4923 acc_train: 0.7333 time: 0.1351s
Epoch: 0071 loss_train: 0.4808 acc_train: 0.7444 time: 0.1239s
Epoch: 0081 loss_train: 0.4732 acc_train: 0.7556 time: 0.1634s
Epoch: 0091 loss_train: 0.4364 acc_train: 0.7444 time: 0.1577s
Epoch: 0101 loss_train: 0.4312 acc_train: 0.7778 time: 0.1569s
Epoch: 0111 loss_train: 0.4133 acc_train: 0.7889 time: 0.1516s
Epoch: 0121 loss_train: 0.4318 acc_train: 0.8222 time: 0.1423s
Epoch: 0131 loss_train: 0.3662 acc_train: 0.8222 time: 0.1726s
Epoch: 0141 loss_train: 0.3942 acc_train: 0.7778 time: 0.1372s
Epoch: 0151 loss_train: 0.3205 acc_train: 0.8778 time: 0.1377s
Epoch: 0161 loss_train: 0.3322 acc_train: 0.8778 time: 0.1536s
Epoch: 0171 loss_train: 0.3497 acc_train: 0.8444 time: 0.1601s
Epoch: 0181 loss_train: 0.2732 acc_train: 0.9000 time: 0.1552s
Epoch: 0191 loss_train: 0.2656 acc_train: 0.9000 time: 0.1622s
Optimization finished!
loss_test: 0.3269 acc_test: 0.8000 time: 0.1923s
```

Figure 2: Train and test accuracy of graph classification.

References