Zakaria ECHCHAIR
Zakaria.ECHCHAIR@polytechnique.edu

1 Question 1

For directed graphs, we just need to adapt the random walks (choose only the outgoing flows and sample between them randomly to create our random walk) (the function G.neighbors gives only nodes in the correct direction which is perfect as implementation) and the following parts of the algorithms remain the same . Neverthless, the weight problem could be solved by the sampling edges for each walk based on the weight of the edge as the probability to be picked. Indeed for the implementation we can just add the argument *probability* to random.choice(). Such that, the probability of each edge to be picked is exactly it's weight. Moreover, we notice that the word2vec skip gram approach treats the output of the random walks the same way, either it's derivated from a directed or undirected graph. So the skip gram approach would'nt take into account the positianal sequence of the walk (which should compulsory be from left to right in the directed graph case). And to upgrade the performance, maybe we could implement instead of the skipgram of word2vec either CBow or positional embedding layer.

2 Question 2

In case where we have not connected nodes, we find ourselves in this case.
In case we have only nodes with features but with no edges. So the we will fall in the case of Multilayer Perceptron (MLP).
INdeed, the adjacnt matrix would be null, then the normalized adjacent matrix would be the identity. Therefore the multiplication of the normalized adjacent matrix will disappear from all the equations (multiplication of identity).Thus, the result would be absolutely a simple layer with activation function (relu) which is the case of MLP.

3 Question 3

In general, in a GCN architecture with k message passing layers, the maximal number of edges separating a given node i from the nodes of which are taken into account in the prediction $\hat{Y}$i is *K edge* because the value at node j of the function centred at node i is 0 if the the minimum number of edges connecting two nodes on the graph is greater than K. So the maximum value of edge that could be taken with a non zero value is K.Typically, for our case, with two hidden layers K=2 is often used, which corresponds to 5x5 convolutional filters (as analogy)

4 Question 4

We consider a fully connected graph on 4 nodes.

Firstly we calculate the adjacent Matrix A : We have

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Then we will calculate : $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$

$$\tilde{\mathbf{A}} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Then :

$$\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij} = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix}, So \quad \tilde{\mathbf{D}}^{\frac{1}{2}} = \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{pmatrix}$$

And :

$$\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{pmatrix}$$

Now let's calculate Z_1 for K_4 :

We Have :

$$\mathbf{Z}^0 = f\left(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}^0\right) = \begin{pmatrix} 0 & 0.5 \\ 0 & 0.5 \\ 0 & 0.5 \\ 0 & 0.5 \end{pmatrix}$$

So,

$$\mathbf{Z}^1 = f\left(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}^1\right) = \begin{pmatrix} 0 & 0.3 & 0.25 \\ 0 & 0.3 & 0.25 \\ 0 & 0.3 & 0.25 \\ 0 & 0.3 & 0.25 \end{pmatrix}$$

We consider now a star graph S4.

Firstly we calculate the adjacent Matrix A : We have

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Then we will calculate : $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$

$$\tilde{\mathbf{A}} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

Then :

$$\tilde{\mathbf{D}}^{\frac{1}{2}} = \begin{pmatrix} 0.50 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.71 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.71 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.71 \end{pmatrix}$$

And :

$$\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} = \begin{pmatrix} 0.25 & 0.35 & 0.35 & 0.35 \\ 0.35 & 0.50 & 0.00 & 0.00 \\ 0.35 & 0.00 & 0.50 & 0.00 \\ 0.35 & 0.00 & 0.00 & 0.50 \end{pmatrix}$$

Now let's calculate Z_1 for S_4 :

We Have :

$$\mathbf{Z}^0 = f\left(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}^0\right) = \begin{pmatrix} 0.00 & 0.66 \\ 0.00 & 0.43 \\ 0.00 & 0.43 \\ 0.00 & 0.43 \end{pmatrix}$$

So,

$$\mathbf{Z}^1 = f\left(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}^1\right) = \begin{pmatrix} 0.00 & 0.37 & 0.31 \\ 0.00 & 0.27 & 0.22 \\ 0.00 & 0.27 & 0.22 \\ 0.00 & 0.27 & 0.22 \end{pmatrix}$$

We observe that the first, second and third rows are identical for S4 and that all rows are identical for K4. This is logical because in our representation, similar rows correspond to nodes with the same structural characteristics and with the same degree. Moreover, it is expected since, in this small context, the graph is neither weighted nor directed. Thus, each of them has the same position in the graph, which gives the same representation.
If we had randomly sampled the characteristics of the nodes X from a random uniform distribution, we would have, the structure would be different for the nodes with many connections as in the case of S4. On the other hand, the representation of K4 would not be similar to the one we have now because of the random factor we add.