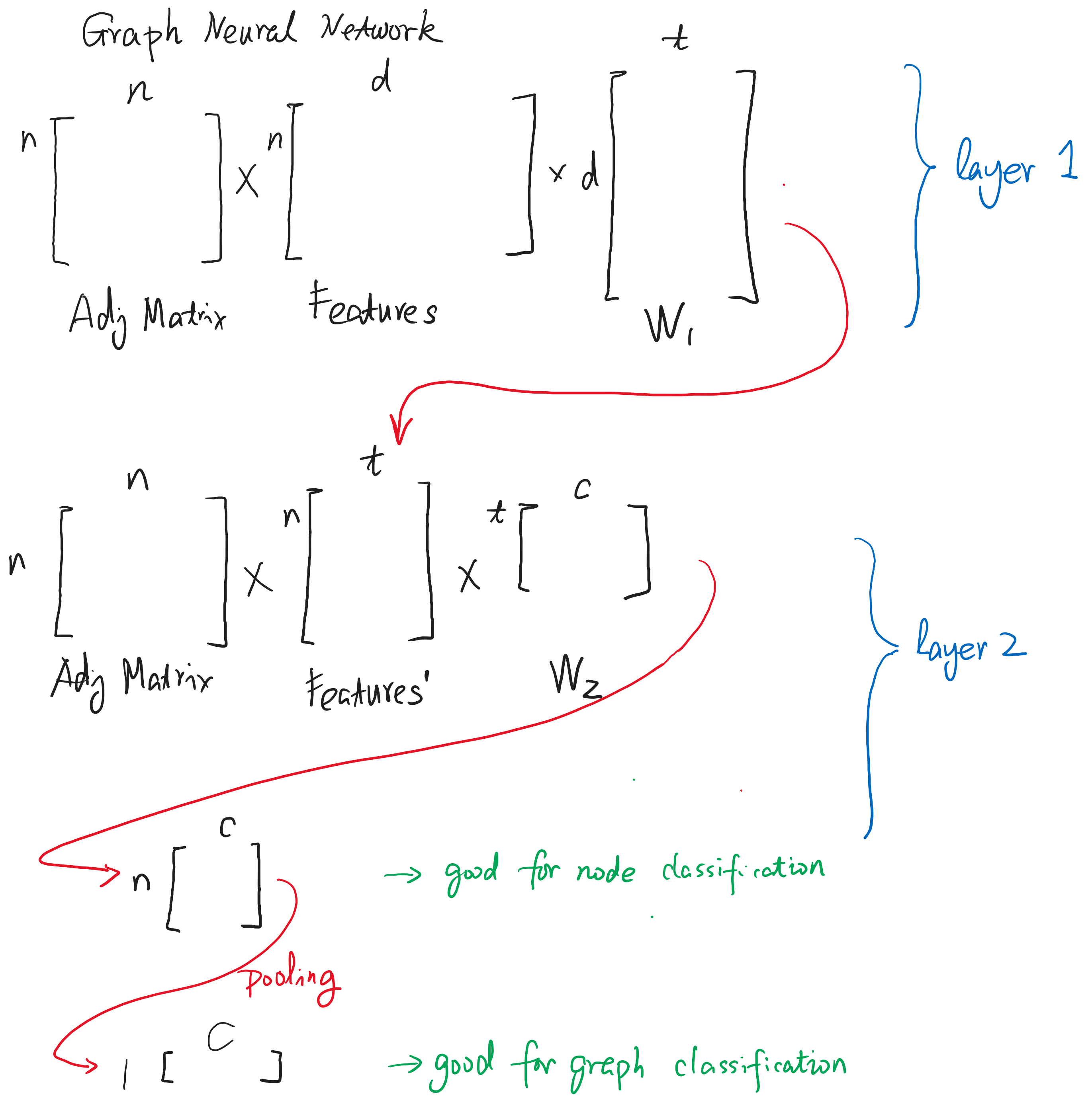


GCN does not need to care about the size of the graph. In fact, what to be learnt are W_1 and W_2 in this example.

- W_1 is decided by the number of features and the size of the hidden layer.
- W_2 is decided by the size of the hidden layer and the num of classes.

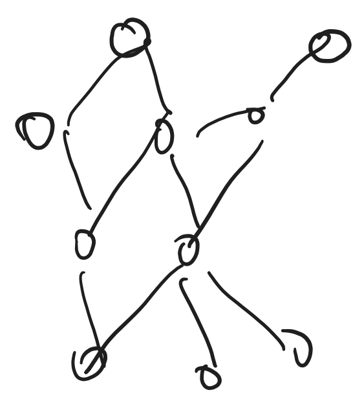
d: # of features

t: size of the hidden layer

c: # of classes.

This GCN does not care about the size of the graph because the adj matrix and features of nodes are used as inputs. However, d, t, and c should be fixed for a training task.

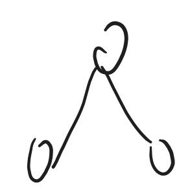
For example:



$G_1 \left\langle \begin{matrix} A_1 \\ X_1 \end{matrix} \right. \Rightarrow GCN(A_1, X_1) \Rightarrow$

$$\overset{n_1\times t}{O} = \overset{n_1\times n_1}{A_1} \times \overset{n_1\times d}{X_1} \times \overset{d\times t}{W_1}$$

$$\overset{n_1\times c}{O} = \overset{n_1\times n_1}{A_1} \times \overset{n_1\times t}{O} \times \overset{t\times c}{W_2}$$



$G_2 \left\langle \begin{matrix} A_2 \\ X_2 \end{matrix} \right. \Rightarrow GCN(A_2, X_2) \Rightarrow$

$$\overset{n_2\times t}{O} = \overset{n_2\times n_2}{A_2} \times \overset{n_2\times d}{X_2} \times \overset{d\times t}{W_1}$$

$$\overset{n_2\times c}{O} = \overset{n_2\times n_2}{A_2} \times \overset{d\times t}{O} \times \overset{t\times c}{W_2}$$

So, what really matters for GCN are d, t, and c.