# Deep Learning – Tutorial 9

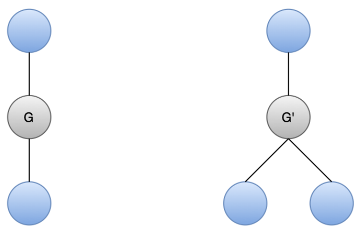

Imperial College London
Department of Computing


January 6, 2021

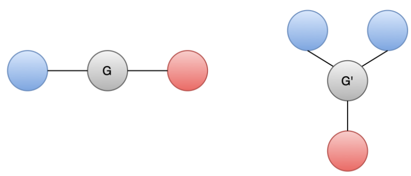
Lecturers: Michael Bronstein, Yingzhen Li, Bernhard Kainz.

## Problem 1

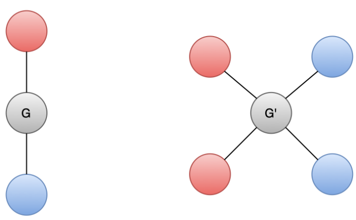
**a)** An aggregator function is a mathematical tool that is used to express the significance of some input data by combining them into a single value. Examples of aggregator functions are the SUM, MEAN and MAX functions. Aggregator functions are used in Graph Neural Networks to extract feature information from a node's local neighborhood e.g. in GraphSAGE (https://arxiv.org/pdf/1706.02216.pdf). In the following diagrams, same coloring indicates nodes of the same value. Use one or more from the three above mentioned aggregator functions to distinguish graph **G** from graph **G'** in the following four scenarios.



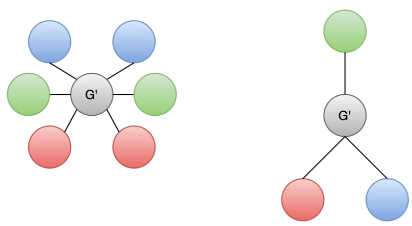

(a) Graph




(b) Graph

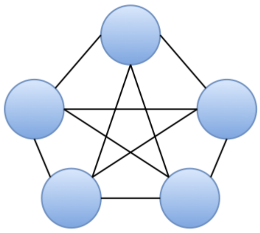



(c) Graph

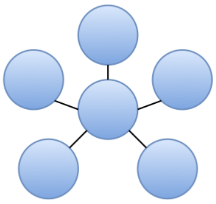



(d) Graph

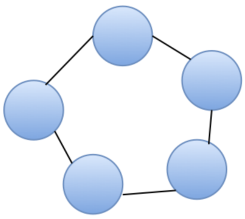
**b)** For each graph in column 1, choose the corresponding Laplacian Matrix from column 2.



| | | | | | |
|---|---|---|---|---|---|
| 5 | -1 | -1 | -1 | -1 | -1 |
| -1 | 1 | 0 | 0 | 0 | 0 |
| -1 | 0 | 1 | 0 | 0 | 0 |
| -1 | 0 | 0 | 1 | 0 | 0 |
| -1 | 0 | 0 | 0 | 1 | 0 |
| -1 | 0 | 0 | 0 | 0 | 1 |



| | | | | |
|---|---|---|---|---|
| 4 | -1 | -1 | -1 | -1 |
| -1 | 4 | -1 | -1 | -1 |
| -1 | -1 | 4 | -1 | -1 |
| -1 | -1 | -1 | 4 | -1 |
| -1 | -1 | -1 | -1 | 4 |



| | | | | |
|---|---|---|---|---|
| 2 | -1 | 0 | 0 | -1 |
| -1 | 2 | -1 | 0 | 0 |
| 0 | -1 | 2 | -1 | 0 |
| 0 | 0 | -1 | 2 | -1 |
| -1 | 0 | 0 | -1 | 2 |

## Problem 2

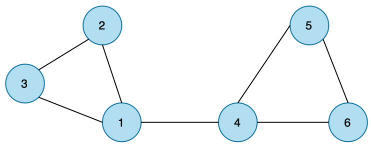
In this exercise you need to compute a message propagation in a Spectral Graph Convolutional Neural Network where the information is propagated along the neighboring nodes within the graph. The figures below show two graphs, one non-directional graph and one directional graph. In Spectral Graph Convolutional Neural Networks the forward passing equation is defined as:

$$\mathbf{Y}^{i+1} = \sigma(\mathbf{H}^{i} * \mathbf{W}^{i}) = \sigma(\mathbf{A} * \mathbf{X} * \mathbf{W}^{i})$$

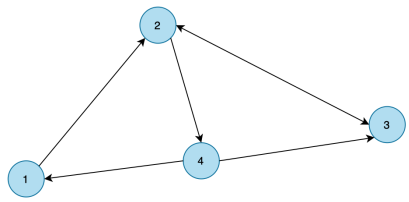
where $\sigma()$ is a non-linear activation function, $\mathbf{X}$ is the node feature vector, $\mathbf{W}$ is the weight vector and $\mathbf{A}$ is the adjacency matrix. Although this relationship captures the message forward passing, through different research evidence it has been modified to:

$$\mathbf{Y}^{i+1} = \sigma(\mathbf{D}^{-1} * \mathbf{A}_{hat} * \mathbf{X} * \mathbf{W}^{i})$$

where $\mathbf{A}_{hat}$ is the inserted-self-loops Adjacency matrix and $\mathbf{D}$ is the graph degree matrix. In the next questions, we will see why this modification is essential.




(a) Non-directional Graph




(b) Directional Graph

**a)** Compute the Adjacency matrix for each graph.

**b)** Calculate the $\mathbf{H} = \mathbf{A} * \mathbf{X}$ for each graph (The $\mathbf{X}$ feature values are displayed in the graphs). What do you notice ? Can you draw any conclusions regarding the dot product of the Adjacency matrix and the Node Feature matrix ? What is the problem ?

**c)** Compute the inserted self-loops Adjacency matrix $\mathbf{A}_{hat} = \mathbf{A} + \mathbf{I}$ for each graph where $\mathbf{I}$ is the identity matrix.

**d)** Calculate the $\mathbf{H}_{hat} = \mathbf{A}_{hat} * \mathbf{X}$ for each graph. Do you see anything different compared to $\mathbf{H}$ in question (b) ?

**e)** Briefly mention some issues that might happen due to non-normalized data in a neural network during training.

**f)** Compute the $\mathbf{D}$ degree matrix for both graphs. Calculate $\mathbf{D}^{-1} * \mathbf{A}_{hat} * \mathbf{X}$ for both graphs and compare with your results in question (d). Comment on the results.

## Problem 3

**a)** Let G be a d-regular graph. Prove that 0 is an eigenvalue for the Laplacian matrix ($\mathbf{L}$) of G.

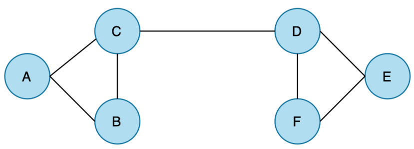
**b)** $\mathbf{L}$ denotes the Laplacian matrix of graph G and E denotes the set of edges in G. And let $\mathbf{x} \in R^n$, prove that

$$\mathbf{x}^T * \mathbf{L} * \mathbf{x} = \sum_{[i,j] \in E} (x_i - x_j)^2 \tag{1}$$

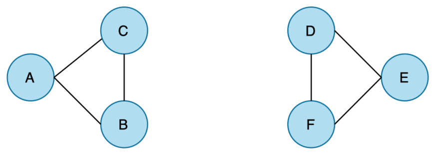
**c)** A connected component of an undirected graph is a connected subgraph such that there are no edges between vertices of the subgraph and vertices of the rest of the graph. Let G be a graph. Prove that if G is connected, the algebraic multiplicity of eigenvalue 0 for the Laplacian matrix is exactly 1.

**d)** Let G be a graph. Prove that a graph G has n connected components if and only if the algebraic multiplicity of 0 eigenvalue in the Laplacian ($\mathbf{L}$) is n.

(Hint: Let $\mathbf{C}$ be a block diagonal matrix with block matrices $C_1, C_2, ..., C_n$, then $det(\mathbf{C}) = det(C_1) * ... * det(C_n)$)




(a) Social Network 1

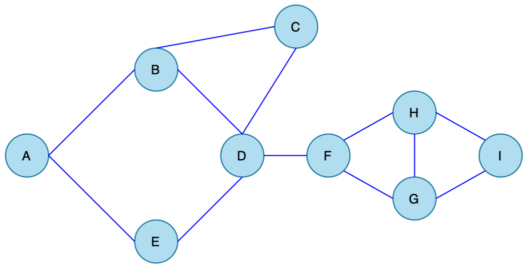



(b) Social Network 2

**e)** The two figures above display two social networks. Each node represents a user and each edge represents friendship status between the two nodes. For each social network, state the multiplicity of 0 eigenvalue of their respective Laplacian matrix. For the graph with 0 eigenvalue multiplicity of 1, suggest and perform clustering between the two groups of friends using the Laplacian Matrix.

**Problem 4**

a) Given the following graph G, calculate the clustering coefficients for each node, the distance table and the diameter of the graph.




b) Describe an example of a graph G where the diameter is more than three times as large as the average path length.

c) How could you extend your construction in question (b) to produce graphs in which the diameter exceeds the average path length by as large a factor as you like ? In other words, for every number k, can a graph be produced with the diameter is more than k times as large as the average path length ?