Topological and geometric deep learning

TODO FIXME

Student: **Nikolay Chechulin**, DSBA-201
Academic Supervisor: **Oleg Kachan**

NATIONAL RESEARCH UNIVERSITY
Faculty of Computer Science
Bachelor’s programme ‘Data Science and Business Analytics’
Moscow, Russia
2022

Contents

1 Introduction

1.1 Abstract

Nowadays, a need to analyze more complex data arises. Some objects and relations can not be represented as vectors in Euclidean space, and, therefore, we have to consider graphs — sets of nodes and connections between them — as a subject of analysis. This poses a huge problem: we have to invent new algorithms, adapt known techniques and constantly improve them in order to work with such a complex data. Our goal is to research the efficiency of several tweaks of existing models.

1.2 Relevance

The field of research (graph neural networks) might be considered relatively new, and, therefore, there is a huge number of possible improvements to be made to existing models and approaches. Our ultimate goal is to improve the accuracy of node and graph classification.

For example, one of the proposed changes is to modify a Laplacian in such a way that it does not break existing model and improves it. Our initial results have shown that our approach indeed works well on Karate club dataset, where we had to classify nodes:

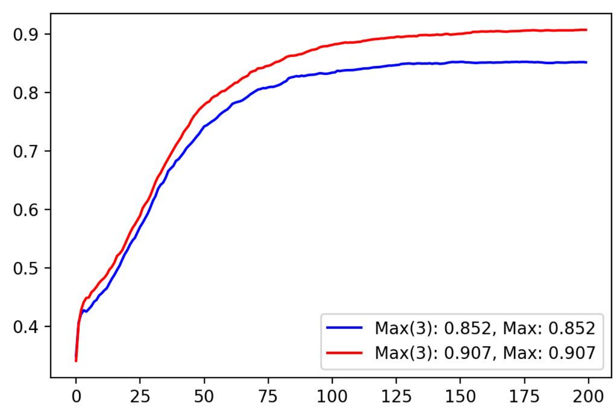

Figure 1: Default Laplacian (in blue) versus our Laplacian (in red). Y-axis is the accuracy, X-axis is the number of epochs

1.3 Subject of research

As we established, we want to consider several changes in order to improve the accuracy. The tweaks we propose include but are not limited to:

- *Altering the way we compute Laplacian* — a characteristic matrix of a graph.
- *Edge embeddings*
- *Using connectivity over simplices of higher dimension*. This means that in some cases we might want to consider a group of nodes as a separate object, therefore, increasing the connectivity factor. We will only work with 3-simplices:

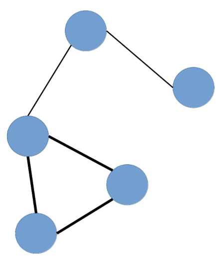

Figure 2: A part of some graph

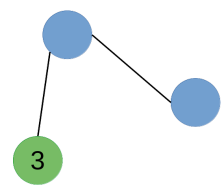

Figure 3: Three nodes from the left image united in 3-simplex having properties of the initial vertices

2 Definitions

CGN *(Convolutional Graph Neural Network)* — A type of GNN which generalizes the convolution operation to graphs. Often we encounter convolution while we work with grid-structured data like images, but here we use same idea (aggregate features of the neighbors) on nodes instead of pixels[3].

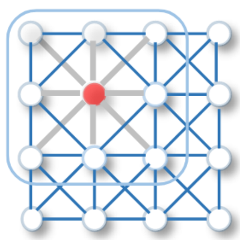

Figure 4: Convolution on image

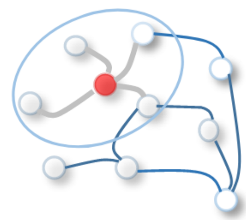

Figure 5: Convolution on graph

GAT *(Graph Attention Network)* — A type of GNN which uses attention mechanism (also borrowed from 'casual' neural networks) which allows us to work with inputs of variable sizes and to focus on the most important features [2].

Laplacian matrix — A matrix representation of a graph. Usually is calculated using the following formula [1]:

$$L_{i,j} = \begin{cases} \deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise,} \end{cases}$$

References

[1] "Laplacian matrix". In: *Wikipedia* (). URL: https://en.wikipedia.org/wiki/Laplacian_matrix.

[2] Petar Veličković et al. "Graph Attention Networks". In: *International Conference on Learning Representations* (2018). accepted as poster. URL: https://openreview.net/forum?id=rJXMpikCZ.

[3] Zonghan Wu et al. "A Comprehensive Survey on Graph Neural Networks". In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1 (2021), pp. 4–24. DOI: 10.1109/TNNLS.2020.2978386.