Topological and geometric deep learning

TODO FIXME

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

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 graps — 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.

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.

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:
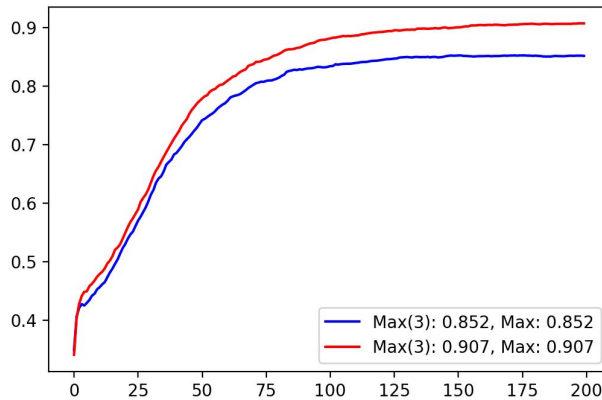

Figure 1: Default Laplacian (in blue) versus our Laplacian (in red). Y-axis is the accuracy, X-axis is the number of epochs

1.3 Subject of research

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[**9046288**].
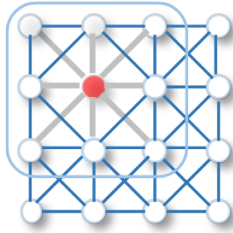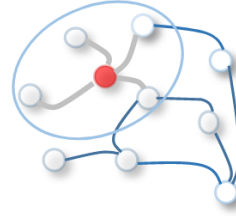

Figure 2: Convolution on image

Figure 3: 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 [**velickovic2018graph**].

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

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