TDDE70 Lab 3 preliminaries and illustrations

Filip Ekström Kelvinius

February 2024

1 Introduction

In this lab, the goal is to learn more about working with graph neural networks in practice. For implementing such network, we will use `Pytorch Geometric` , and in particular, the `MessagePassing` class.

NOTE: For this lab, use the conda environment `lab3`. In case you are not using the lab computers, `lab3.yml` specifies the required packages.

Before the lab session To best prepare for the lab, please do the following steps before the lab session

1. Make sure you are familiar with the course material (i.e., the lecture) about GNNs.
2. Do Task 0 in the lab notebook to familiarize yourself with data and batching in `Pytorch Geometric`
3. Have a look at the `Pytorch Geometric` intro tutorial[1], in particular the introduction, the part about the `MessagePassing` base class, and about implementing GCN.
4. Familiarize yourself with the data that you will be working with by reading the next section

2 Molecules as graphs

The dataset that you will work with is a molecular dataset, and as that probably is a new type of application for you, this section aims at giving a brief introduction to how molecules can be represented as graphs.

In the lab, the task of the GNN is to predict the energy E of a molecule consisting of N atoms. We can describe this mathematically as

$$\hat{E} = \mathrm{GNN}(\mathbf{X}, \mathbf{z}),$$

[1] `https://pytorch-geometric.readthedocs.io/en/latest/notes/create_gnn.html`

where $\mathbf{X} \in \mathbb{R}^{N \times 3}$ denotes the positions of each atom, and $\mathbf{z} \in \mathbb{Z}^N$ denoting the type (atomic number) of each atom.

From the input to the GNN ($\mathbf{X}$ and $\mathbf{z}$), it is rather straight forward to construct a graph: each atom correspond to a node, and we can create edges between atoms that are within a certain distance from each other. The node features could then be the type of atom, and edge features could incorporate geometric information like distance between the atoms.

An illustration in two dimensions of a toy molecule and how a graph could be constructed is shown in Figure 1-3.

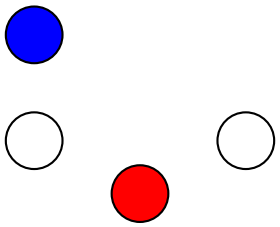

Figure 1: A toy molecule in two dimensions. Different colors correspond to different types of atoms.

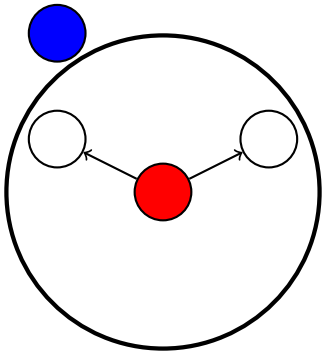

Figure 2: To construct edges, we can connect atoms within a certain distance from each other. In this example, we add edges from the red atom to atoms within a certain cutoff radius illustrated by the black circle. In three dimensions, the circle would correspond to a sphere.

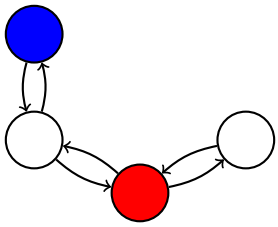

Figure 3: By applying the process from Figure 2 to the other atoms, we obtain our graph. In this case, the edges are symmetric in the sense that for each edge (i, j) there is always another edge (j, i). However, depending on how the molecule looks like and if we impose the additional constraint that an atom can only have a certain maximum number of neighbors, this might not be the case.