Structure Recognition with Graph Neural Networks - Intermediate Report

Benedikt Wenzel

December 9, 2024

Abstract

The project lies at the intersection of Machine Learning and solid-state physics. A common task in solid-state physics is the classification of atomic structures, for example in crystals. Machine Learning on the other hand is well known for its ability in classification tasks. Combining these two worlds provides a powerful tool for classifying different crystal structures. As crystal structures are described by the so-called Bravais lattices, which closely relate to graphs, we will need machine learning tools capable of properly handling graph-like data. Fortunately, in the last few years, a new type of neural networks, exactly designed for this kind of data, emerged: Graph-Neural-Networks. The overall goal of this project is to get familiar with GNNs and apply them to classification tasks of Bravais lattices.

1 Background

This introductory section lays the theoretical foundation of Bravais lattices and GNNs. We start with a short recap of lattice structures in 2d and 3d and then continue with the most fundamental background of Graph-Neural Networks (GNN). Results on Bravais lattices can be found in many textbooks. Subsection 1.1 follows [4] and [3]. Introduction to Graph Neural Networks can be found for example in [1] which is also the main reference for section 1.2.

1.1 Bravais lattice

Let $d \in \mathbb{N}$ and $\{b_i\}_{i=1,\dots,d} \subset \mathbb{R}^d$ a basis of $\mathbb{R}^d$. The set

$$\Omega \coloneqq \left\{ \sum_{i=1}^{d} z_i b_i : z_i \in \mathbb{Z} \, \forall i \in \{1, \dots, d\} \right\}$$

is called a d-dimensional lattice. Given any subset $S \subset \mathbb{R}^d$, we define its point group G_S to be

$$G_S \coloneqq \{M \in O(d) : MS = S\} \subset O(d).$$

G_S is obviously a subgroup of $O(d)$. We say that two d-dimensional lattices $\Omega_1, \Omega_2 \subset \mathbb{R}^d$ are of the same Bravais type if there exists $g \in GL_n(\mathbb{R})$ such that $G_{\Omega_1} = g G_{\Omega_2} g^{-1}$ and $\Omega_1 = g\Omega_2$. Being of the same Bravais type introduces an

equivalence relation on the set of all d-dimensional lattices. We refer to the equivalence classes as Bravais classes. A natural question arising is about the total number of Bravais classes. Despite being a very interesting and challenging problem, we will leave this question to the mathematicians. For us, the result is more important than the actual proof. One obtains the following result: For $d = 2$ there are 5 Bravais classes and for $d = 3$ there are 14 Bravais classes. Roughly speaking, they can be distinguished by the relative lengths of the basis vectors b_i and the angles between them. Visualizations and further notes on each Bravais class can be found for example in [2].

1.2 Fundamentals of GNNs

To talk about graphs, we first have to agree on some notation: Let V be a set and $E \subset V \times V$. The tuple $G = (V, E)$ is called a graph. Furthermore, we call an element $x \in V$ a node and a tuple $(x, y) \in E$ a directed edge from x to y, and we say that x is a neighbor of y. In case $(x, y) \in E$ implies $(y, x) \in E$, we speak of an undirected graph. In that case, we can think of elements in E as unordered tuples $\{x, y\}$ instead of ordered ones. We still call $\{x, y\}$ an edge between x and y. For $y \in V$ we define the neighborhood N_y of y to be the set of all neighbors, i.e.

$$N_y \coloneqq \{x \in V : (x, y) \in E\} \subset V. \tag{1}$$

Furthermore, we assign to each node $x \in V$ a vector $v_x \in \mathbb{R}^n$ called node feature and to each edge $(x, y) \in E$ a vector $e_{x,y} \in \mathbb{R}^m$ called edge feature.

Roughly, a GNN takes a graph with all its nodes and edge features as an input and manipulates these features in each step. More than that, a GNN can transform the structure of the graph itself, e.g. by introducing new nodes or edges. However, we will not go into detail about this possibility and stick to the simpler case of manipulating only node and edge-features. Furthermore, we restrict ourselves to the case where the GNN does not alter the edge features. Let us make these ideas a bit more rigorous (see figure 1 for an illustration): Let (V, E) be a graph with node features $\{v_x \in \mathbb{R}^n : x \in V\}$ and edge features $\{e_{x,y} \in \mathbb{R}^m : (x, y) \in E\}$. For each $c \in V$ a new node feature $\widehat{v_c}$ is calculated according to the following rule

$$\widehat{v_c} = \phi_{upd}\left(v_c, \phi_{agg}\left(\{\phi_{mes}\left(v_y, v_c, e_{y,c}\right) : y \in N_c\}\right)\right), \tag{2}$$

where $\phi_{upd}, \phi_{agg}, \phi_{mes}$ denote differentiable functions. These three functions are commonly interpreted as follows: ϕ_{mes} takes as inputs the node value v_c and the node value v_y of one neighbor y of c as well as the value $e_{y,c}$ of the edge (y, c). Depending on these inputs, it then computes a value, which can be thought of as a message originating from node y which is sent to node c. ϕ_{agg} collects all these messages to node c and aggregates them in some way, so that the output can be thought of as one overall message to node c. ϕ_{upd} takes this overall message as well as the value of node c and computes, how the value of node c is altered. Unsurprisingly, this scheme is called Message-Passing-Layer. Given a specific problem, that is required to be solved by a GNN, the challenge is to make appropriate choices for $\phi_{upd}, \phi_{agg}, \phi_{mes}$ that suit the problem at hand. Furthermore, one has to decide how many iterations of the above procedure are suitable.

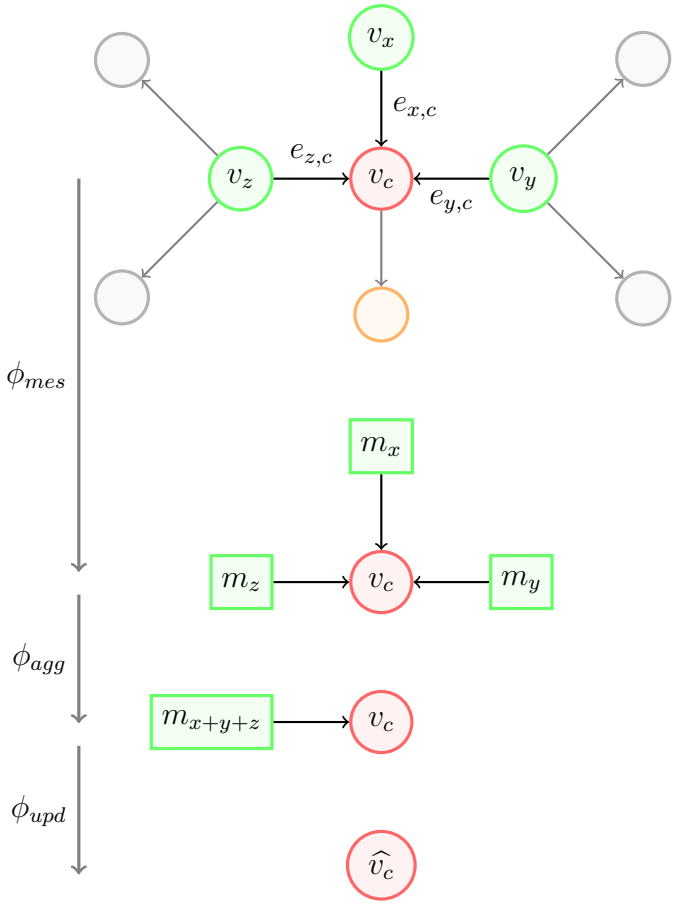

Figure 1: Illustration of the message passing procedure according to equation (2). The neighbors of node c (red) are the nodes x, y, z (green). Attention: According to equation (1) the orange node is not regarded as a neighbor of c as the edge points in the wrong direction. For each neighbor (x, y, z), ϕ_{mes} computes messages (m_x, m_y, m_z) which are sent to c. Then, ϕ_{agg} aggregates these three messages and outputs one overall message m_{x+y+z} that is sent to c. In the last step, ϕ_{upd} updates the node value v_c to a new value $\widehat{v_c}$.

2 Project Goals

The project can be split into two parts. The goal of the first part is to build a GNN that is capable of assigning a 2- or 3-dimensional lattice its Bravais class. For this instance, different lattices need to be created programmatically. Furthermore, they have to be labeled so that they can serve as training data. Once we have a good amount of training data at hand, we need to determine the optimal structure of the GNN. This amounts to finding the right functions $\phi_{upd}, \phi_{mes}, \phi_{agg}$ in equation 2 and to determine which edge-/node-features do best. In the second part of the project, we will build upon the results in the first part and take a look at more complex structures. The goal is to distinguish between percolating and non-percolating structures.

3 Schedule

The first goal mentioned in the previous section has already been achieved. The 2d and 3d lattice-generation is finished and some different GNN structures have already been tested and evaluated. Both, in 2d and in 3d, an accuracy of about 90% has been achieved but probably with further testing, the accuracy can be increased even more. By the end of December further experiments with the network structure are planned. The second goal mentioned in the previous section has not been achieved yet. The code for this is planned to be finished by the beginning of January. The final results are then expected by mid-January so that the report

can be written in the last two weeks of January.

References

[1] Justin Gilmer et al. *Neural Message Passing for Quantum Chemistry*. 2017. arXiv: 1704.01212 [cs.LG]. URL: https://arxiv.org/abs/1704.01212.

[2] Charles Kittel. *Introduction to Solid State Physics*. 8. ed. Wiley, 2005.

[3] Willard Miller. *Symmetry Groups and Their Applications*. Academic Press, 1972.

[4] R. L. E. Schwarzenberger. "Classification of crystal lattices". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 72.3 (1972), pp. 325–349. DOI: 10.1017/S0305004100047162.