

Structure recognition with graph neural networks

**An intermediate report for the course "Advanced Projects in
Computational Physics 2"**

From
Stephen Weybrecht
December 8, 2024

Tutor: Jonas Buba
Supervisor: Prof. Dr. Schmiedeberg

Contents

1	Theoretical introduction	1
1.1	Bravais lattices	1
1.2	Graph neural networks	3
2	Results so far	4
3	Plans for the future	4
	Bibliography	5

1 Theoretical introduction

1.1 Bravais lattices

In the first part of the project, our task was to deal with crystal lattices in 2 and 3 dimensions. For the following discussion an introduction of the terms "crystal", "basis" and "Bravais lattice" is therefore needed.

Following the discussion of [1] an ideal crystal is a periodic, infinite arrangement of atoms in a solid. These atoms are arranged in blocks, a so called basis, in a regularly spaced grid, the lattice. In other words the lattice represents a schema after which individual atoms or groups of atoms (the basis) are arranged to form the crystal. A lattice in d dimensions can be defined by a set of d translation vectors. The superposition of integer multiples of these vectors then makes up the lattice. In principle the length and direction of these vectors can be arbitrary. In this case the lattice would generally not map into itself under translations and rotations – the lattice is called oblique. There are however special sets of translation vectors which form lattices of high symmetry. These fundamental lattices are called the Bravais lattices. For $d = 2$ there are 5 (4 special and one oblique) Bravais lattices, while for $d = 3$ there are 14 (13 special cases and one oblique, so called triclinic lattice). These are depicted in Figure 1 and Figure 2 respectively.

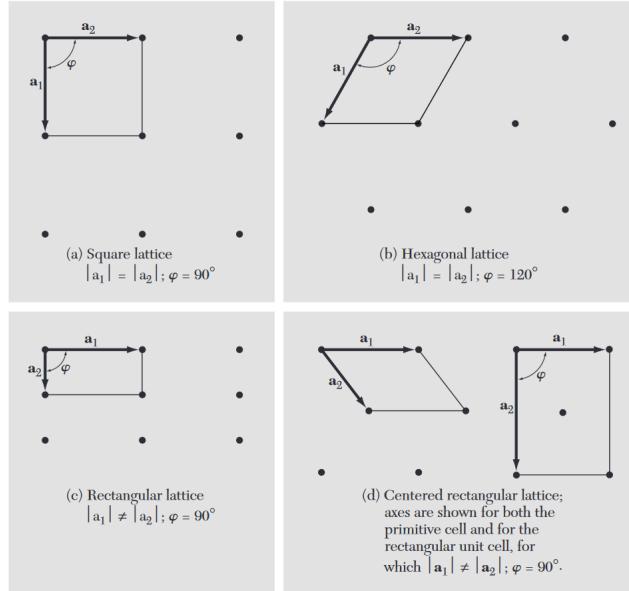


Figure 1: The five Bravais lattices for 2 dimensions. The graphic also illustrates the length of the translation vectors a_i and the angle between them in order to make up the corresponding lattice. Taken from [1, Fig. 7].

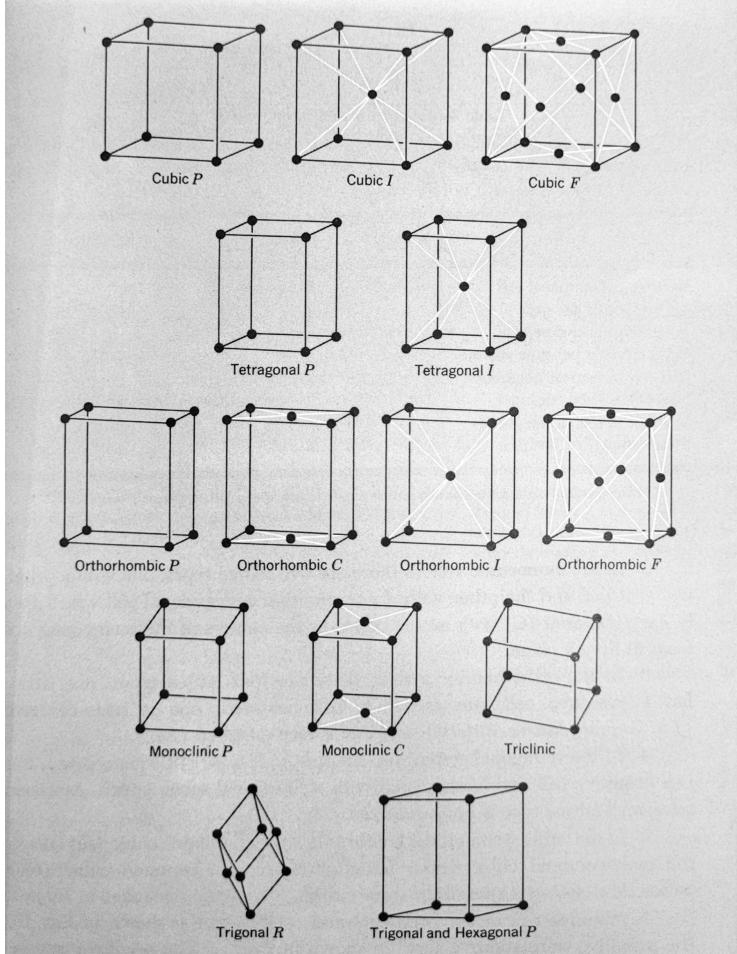


Figure 2: The 14 Bravais lattices for the $d = 3$ case. One further classifies these lattices into seven types of cells: Cubic ($a=b=c$, $\alpha=\beta=\gamma=90^\circ$), Tetragonal ($a=b \neq c$, $\alpha=\beta=\gamma=90^\circ$), Orthorhombic ($\alpha=\beta=\gamma=90^\circ$), Monoclinic ($\alpha=\beta=90^\circ \neq \gamma$), Triclinic, Trigonal ($a=b=c$, $\alpha=\beta=\gamma < 120^\circ \neq 90^\circ$), Hexagonal ($a=b \neq c$, $\alpha=\beta=90^\circ$, $\gamma=120^\circ$). Note that in the prior notation a , b and c denote the lengths of the translation vectors, α , β , γ the angles between them and omitted length or angle relations mean they can be arbitrary. These groups are again subclassified based on their lattice structure into simple "P", body-centred "I", base-centred "C", face-centred "F". As mentioned, all lattices are special cases of the general, triclinic case. The graphic is taken from [2, Fig. 14]

1.2 Graph neural networks

The lattices as discussed in the prior section are a collection of atoms linked by bonds and can therefore be suitably represented by a graph, consisting of nodes and edges. If we want to apply machine learning to crystal lattices, we therefore need models that are well suited for data organized in a graph-like manner. What follows is a basic introduction to such networks, specifically graph convolutional neural networks (GCNs).

GCNs take inspiration in the already well-established convolutional neural networks in which a typical layer consists of a trainable kernel that can be applied on ordered, grid-like training data of arbitrary size and shape (e.g. images) [3]. GCNs represent a generalisation of this concept onto unordered nodes with a variable number of neighbours. Each GCN-layer uses so called message passing in order to update the node state $h_u^{(t)}$ of a time t to the next step $h_u^{(t+1)}$ as shown in Equation 1 [3, eq. 4.1].

$$h_u^{(t+1)} = \text{UPDATE}^{(t)} \left(h_u^{(t)}, \text{AGGREGATE}^{(t)} \left(\{h_v^{(t)}, \forall v \in N(u)\} \right) \right) \quad (1)$$

In the above equation $\text{UPDATE}^{(t)}$ and $\text{AGGREGATE}^{(t)}$ could be any differentiable functions i.e. also neural networks, and $N(u)$ denotes the neighbourhood of u meaning all directly connected nodes in the graph. This equation implies the following update schema that is also depicted in Figure 3:

The starting point is a graph, consisting of nodes with feature vectors and connections, that could also have features. For each node, messages from neighbouring nodes are aggregated to a single message by taking a weighted mean of the neighbouring nodes feature vectors. This operation can also be weighted by use of edge features. This message is then passed to a nonlinear update function (e.g. ReLU), that updates the node in question for the next timestep. After the update is completed, further operations can be performed depending on the wanted classification scheme. In the following graph classification is used, which means all feature vectors are pooled in a last step, in order to get a single quantity that is descriptive of the entire graph [3].

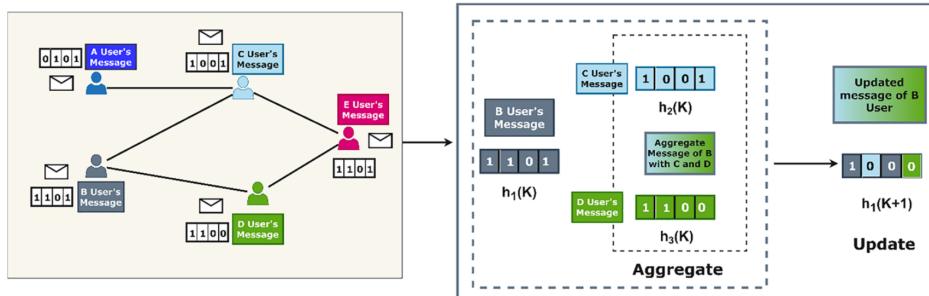


Figure 3: A graphic representation of the update schema for message passing graph neural networks. Taken from [3, Fig. 7].

2 Results so far

3 Plans for the future

Bibliography

- [1] Charles Kittel. “Chapter 1: Crystal Structure”. In: *Introduction to Solid State Physics*. 8. ed. Wiley, 2005. ISBN: 978-0-471-41526-8.
- [2] Charles Kittel. “Chapter 1: Crystal Structure”. In: *Introduction to Solid State Physics*. 4. ed. Wiley, 1971. ISBN: 0-471-49021-0.
- [3] Bharti Khemani, Shruti Patil, Ketan Kotecha, and Sudeep Tanwar. “A Review of Graph Neural Networks: Concepts, Architectures, Techniques, Challenges, Datasets, Applications, and Future Directions”. In: *Journal of Big Data* 11.1 (Jan. 16, 2024), p. 18. ISSN: 2196-1115. DOI: 10.1186/s40537-023-00876-4. URL: <https://doi.org/10.1186/s40537-023-00876-4> (visited on 12/08/2024).