Automatically aligning crystal lattices to neutron diffraction instruments using reinforcement learning

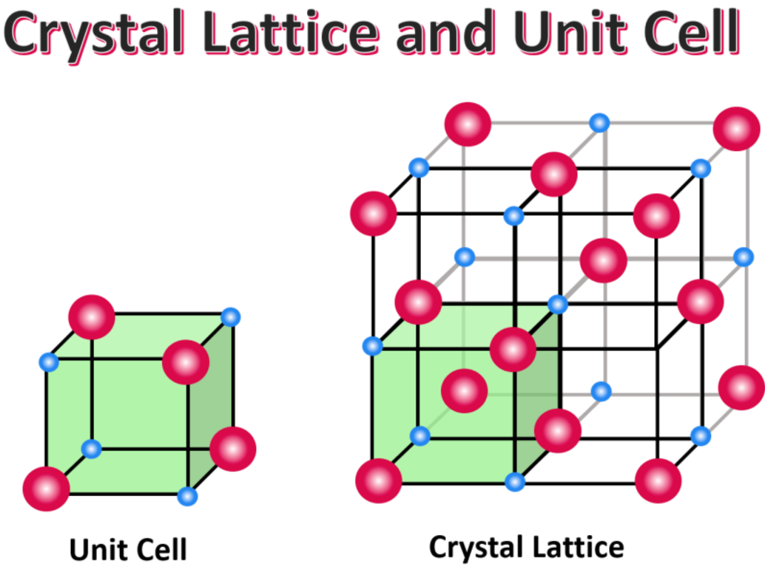
Dylan Zhang

Mentor - Dr. William Ratcliff

University of Michigan

Crystals

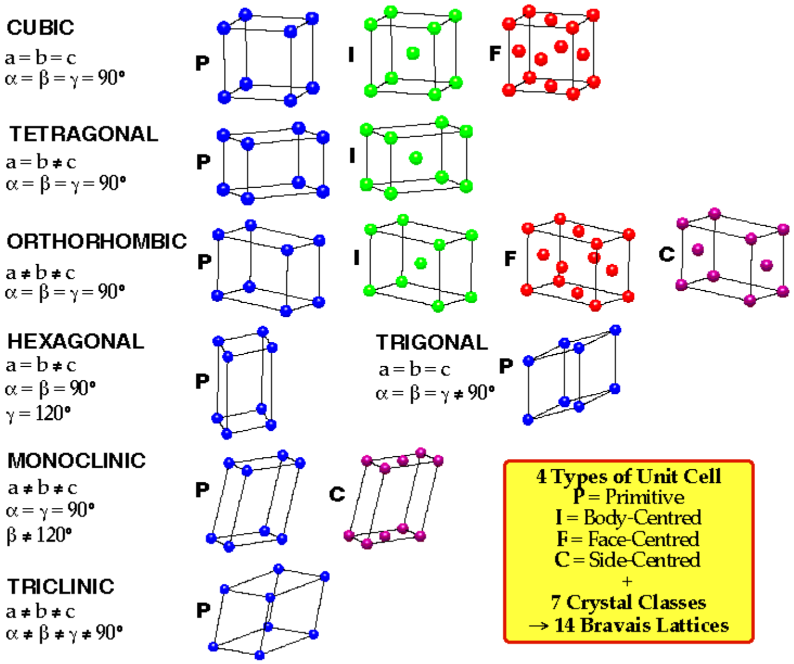
- A type of solid material composed of atoms or groups of atoms that are arranged in a three-dimensional pattern that is very ordered
- Lattice determines the different diffraction patterns



https://www.expii.com/t/crystal-lattice-structure-formation-7999

Bravais Lattice

- Symmetrical three-dimensional structural arrangements
- One of the 14 different types of unit cells that a crystal structure can be made up of
- Determines the different diffraction patterns

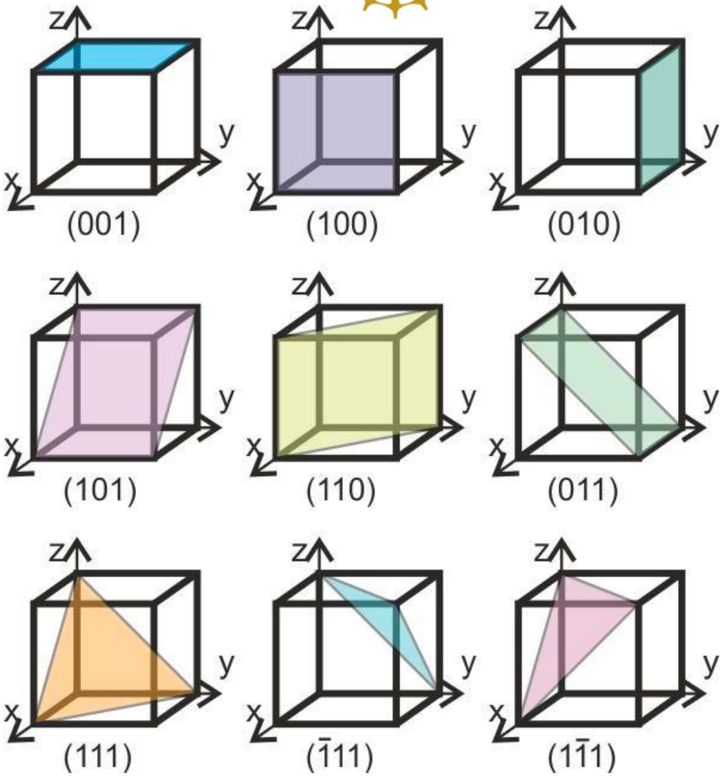


https://www.chm.bris.ac.uk/webprojects2003/cook/latticetypes.htm

Crystallographic Planes

Miller Indices (HKL)

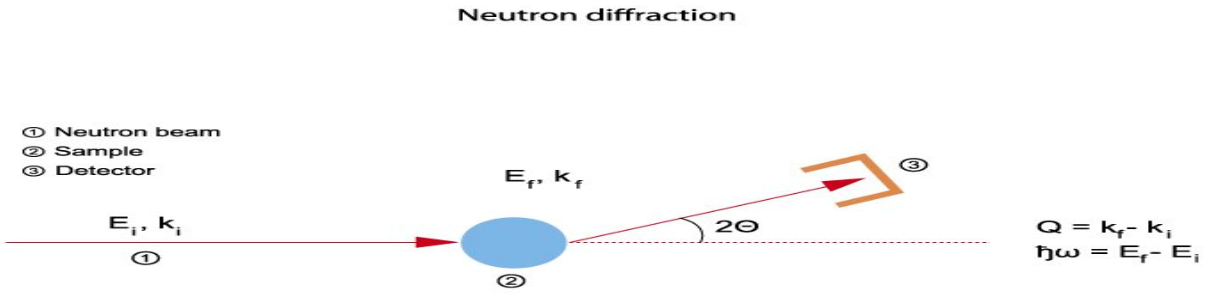
- Represent the orientation of crystallographic planes within the crystal lattice



https://www.researchgate.net/figure/Miller-indices-indicating-the-plane-perpendicular-to-the-vector-given-for-the-cubic_fig7_302838100

Neutron Diffraction

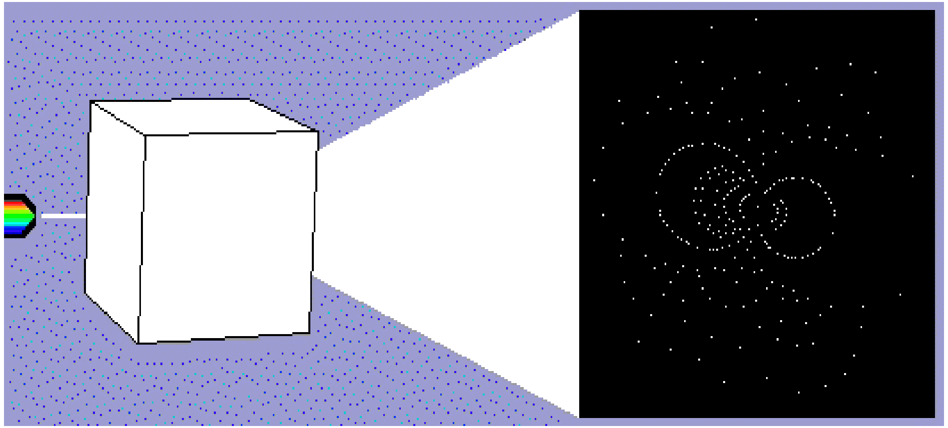
- Neutron Diffraction is the application of neutron scattering to determine the atomic and magnetic structure of a material
- A sample is placed in a beam of neutrons to obtain a diffraction pattern that provides information about the structure of the sample



https://nmi3.eu/neutron-research/techniques-for-/structural-research.html

Objective

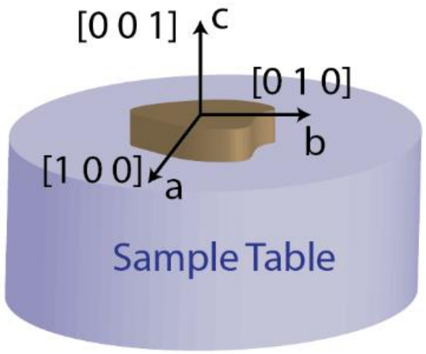
- Automatically align crystal structures to instruments
- Assist in fully automating the crystal diffraction process



https://www.ill.eu/users/instruments/instruments-list/orientexpress/how-it-works/principle-of-neutron-laue-diffraction

Alignment

1. Choose a particular plane
2. Choose two projection vectors corresponding to the plane
3. Rotate table
4. Take alignment scans
5. Analyze scans
6. Repeat 3, 4, and 5 until aligned

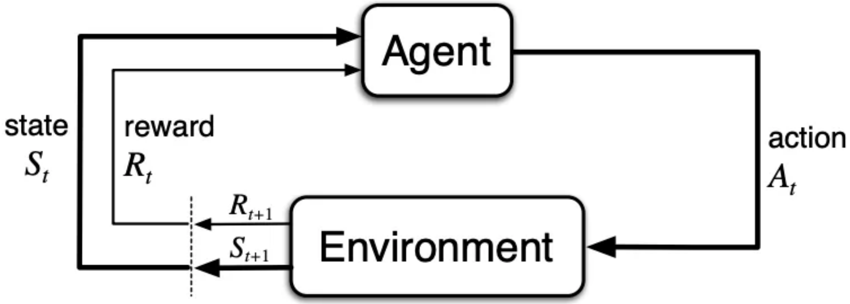


Dr. Leland Harriger

Reinforcement Learning - Key Elements

Area of machine learning where agents take actions in an environment in order to maximize the notion of cumulative reward

- Agent
- Environment
- State
- Action
- Reward



https://towardsdatascience.com/reinforcement-learning-101-e24b50e1d292

Reinforcement Learning - advantages

- Can be used to solve very complex problems
- Can outperform humans in many tasks
- It is similar to the learning of human beings and can correct errors during the training process
- It doesn't require large labeled datasets

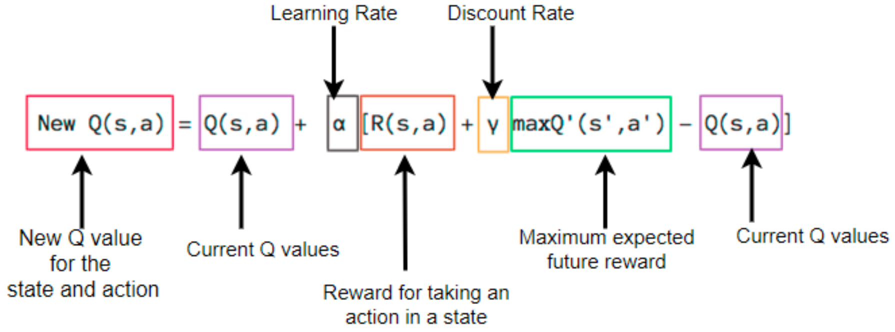
Reinforcement Learning - examples

- DeepMind
- AlphaFold
- AlphaGo

https://venturebeat.com/ai/deepminds-big-losses-and-the-questions-around-running-an-ai-lab/

Q-Learning

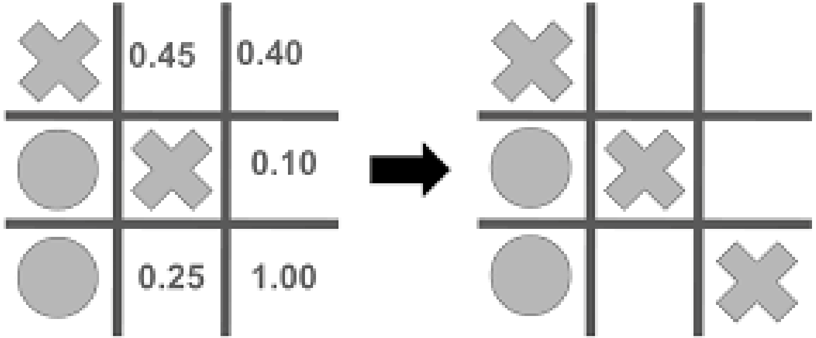
- Model-free reinforcement learning algorithm
- Q-table that stores the expected rewards for state-action pairs
- Q-values updated over multiple episodes
- Agent selects the action with the highest Q-value
- Effective for small state spaces
- Different from Value function



https://towardsdatascience.com/a-beginners-guide-to-q-learning-c3e2a30a653c

Q-learning with tic tac toe

```
    # only when game ends
    def giveReward(self):
        result = self.winner()
        # backpropagate reward
        if result == 1:
            self.p1.feedReward(1)
            self.p2.feedReward(0)
        elif result == -1:
            self.p1.feedReward(0)
            self.p2.feedReward(1)
        else:
            self.p1.feedReward(0.1)
            self.p2.feedReward(0.5)
```



https://towardsdatascience.com/how-to-play-tic-tac-toe-using-reinforcement-learning-9604130e56f6

```
        value_max = -999
        for p in positions:
            next_board = current_board.copy()
            next_board[p] = symbol
            next_boardHash = self.getHash(next_board)
            value = 0 if self.states_value.get(next_boardHash) is None else self.states_value.get(next_boardHash)
            # print("value", value)
            if value >= value_max:
                value_max = value
                action = p
    # print("{} takes action {}".format(self.name, action))
    return action
```

Implementation

- Training:
  - Reward
    - Punished for each rotation
    - Awarded for finding peak
- Decides how much to rotate
- Rotates automatically until finds diffraction peak

Hyperparameters

- Values that control the learning process
- Learning rate, discount factor, episodes

Parameter	Value
Discount factor γ	0.9
Critic net learning rate α_{critic}	0.001
Actor net learning rate α_{actor}	0.001
Batch size N	64
Soft update rate τ	0.01
Standard deviation of Gaussian noise	0.3
Maximum number of training episodes e_{max}	1000
Control time step dt	0.01 s
Maximum time step in an episode t_{max}	5 s
Reward function parameter a_1	1.0
Reward function parameter δ_1	0.01
Reward function parameter ρ_1	π
Reward function parameter a_2	0.1
Reward function parameter δ_2	0.1
Reward function parameter ρ_2	2π

4.2.3. CMA-ES Training Detail in the Simulation Environment

https://www.researchgate.net/publication/343594992/figure/tbl2/AS:932830192341025@1599415457664/Reinforcement-learning-RL-training-hyper-parameters.png

Why/How

Why

- automate some of the crystallography process
- significant because correct alignment is necessary

How

1. Build the q learning algorithm
2. Train the algorithm using simulated data
3. Evaluate/test the algorithm
4. Tune hyperparameters if needed
5. Evaluate again

Results

```
Episode 1: Number of Rotations = 114
Episode 2: Number of Rotations = 138
Episode 3: Number of Rotations = 87
Episode 4: Number of Rotations = 61
Episode 5: Number of Rotations = 12
Episode 6: Number of Rotations = 39
Episode 7: Number of Rotations = 28
Episode 8: Number of Rotations = 31
Episode 9: Number of Rotations = 39
Episode 10: Number of Rotations = 14
Episode 11: Number of Rotations = 83
Episode 12: Number of Rotations = 1
Episode 13: Number of Rotations = 8
Episode 14: Number of Rotations = 38
Episode 15: Number of Rotations = 54
Episode 16: Number of Rotations = 29
Episode 17: Number of Rotations = 6
Episode 18: Number of Rotations = 4
Episode 19: Number of Rotations = 69
Episode 20: Number of Rotations = 10
Episode 21: Number of Rotations = 2
Episode 22: Number of Rotations = 13
Episode 23: Number of Rotations = 35
Episode 24: Number of Rotations = 3
Episode 25: Number of Rotations = 36
Episode 26: Number of Rotations = 16
Episode 27: Number of Rotations = 4
Episode 28: Number of Rotations = 33
Episode 29: Number of Rotations = 145
Episode 30: Number of Rotations = 24
Episode 31: Number of Rotations = 7
Episode 32: Number of Rotations = 3
Episode 33: Number of Rotations = 1
Episode 34: Number of Rotations = 1
Episode 35: Number of Rotations = 1
Episode 36: Number of Rotations = 17
Episode 37: Number of Rotations = 18
Episode 38: Number of Rotations = 12
```

```
Episode 964: Number of Rotations = 9
Episode 965: Number of Rotations = 3
Episode 966: Number of Rotations = 1
Episode 967: Number of Rotations = 5
Episode 968: Number of Rotations = 5
Episode 969: Number of Rotations = 6
Episode 970: Number of Rotations = 2
Episode 971: Number of Rotations = 4
Episode 972: Number of Rotations = 4
Episode 973: Number of Rotations = 2
Episode 974: Number of Rotations = 1
Episode 975: Number of Rotations = 4
Episode 976: Number of Rotations = 2
Episode 977: Number of Rotations = 4
Episode 978: Number of Rotations = 6
Episode 979: Number of Rotations = 1
Episode 980: Number of Rotations = 3
Episode 981: Number of Rotations = 1
Episode 982: Number of Rotations = 2
Episode 983: Number of Rotations = 7
Episode 984: Number of Rotations = 6
Episode 985: Number of Rotations = 6
Episode 986: Number of Rotations = 6
Episode 987: Number of Rotations = 3
Episode 988: Number of Rotations = 4
Episode 989: Number of Rotations = 8
Episode 990: Number of Rotations = 2
Episode 991: Number of Rotations = 1
Episode 992: Number of Rotations = 6
Episode 993: Number of Rotations = 2
Episode 994: Number of Rotations = 4
Episode 995: Number of Rotations = 1
Episode 996: Number of Rotations = 4
Episode 997: Number of Rotations = 3
Episode 998: Number of Rotations = 3
Episode 999: Number of Rotations = 1
Episode 1000: Number of Rotations = 2
```

What's Next

- Mcstas needs to be implemented
- Implement into different types of instruments such as triple-axis spectrometer

Acknowledgements

- Dr. Ratcliff
- CHRNS
- Dr. Borchers
- Dr. Harriger
- Mr. Kienzle
- Mr. Krzywon
- Dr. Liu