Due date: Wednesday, December 15, 2021

!!Please make sure that you properly comment your code in the jupyter notebook!!

1 Neural network for classification of bulk phases

In this worksheet, we develop a neural network (NN) for the classification of local structural environments in bulk phases. There will be three steps that will be worked on over the next three weeks:

(1) **setting up the bulk phases and defining the descriptors (week 1)**

(2) optimizing the hyper-parameters of the descriptors (week 2)

(3) training the NN and application (week 3)

Each step will be submitted separately and the solution will be presented in the tutorial, so that it will be possible to work on subsequent tasks even if the solution to the previous task could not be found.

2 Setting up bulk phases and running MD (10 pt)

In this exercise, we will work with four different bulk phases: face-centred cubic (fcc), body-centred cubic (bcc), hexagonal closed-packed (hcp), and liquid (liq). As interaction potential between the particles we will use again the Lennard-Jones potential.

2.1 Task 1: Setting up the bulk structures

To setup the four structures, you can use the `lattice` module of ase:
https://wiki.fysik.dtu.dk/ase/ase/lattice.html?highlight=lattice#general-crystal-structures-and-surfaces
An example can also be found in the jupyter-notebook for this exercise. For the liquid, use the `SimpleCubic` (sc) structure. Setup all structures such that the interatomic distance $d = 1.1$Å. What are the corresponding lattice constants a in terms of d for the cubic fcc, bcc, and sc cells and what are a and c in terms of d for the hcp lattice assuming an ideal c/a ratio? As atom type use `symbol='H'`. Furthermore, we need to multiply the initial unit cell to have a reasonable number of atoms in the 3D periodic simulation cell. We will use $(7 \times 7 \times 7)$ fcc, $(8 \times 8 \times 8)$ bcc, $(10 \times 10 \times 10)$ sc, and $(9 \times 9 \times 7)$ hcp.

Set up all four structures and report the number of atoms in each simulation cell and the cell vectors. Have a look at the ase `Atoms` object (https://wiki.fysik.dtu.dk/ase/ase/atoms.html) to see which parameters are directly available and how to access, for example, the cell parameters.

To check your setup, you might also want to visually inspect your simulation cells.

2.2 Task 2: Initial lattice optimization

Before we run molecular dynamics (MD) simulations with these cells, we first want to optimize the lattice parameters using an energy-volume curve fitted by an equation of state (EOS). An example is given in the jupyter-notebook and the corresponding ase documentation is https://wiki.fysik.dtu.dk/ase/ase/eos.html.

For the three crystalline phases (fcc, bcc, hcp), set the calculator to `LennardJones` and find the equilibrium volume using an EOS fit. From the equilibrium volume, determine the optimized cell vectors and use `set_cell` to update the cell vectors of the simulation cells. Make sure that you rescale the atomic positions when doing so! Report the new cell volume (`get_volume()`) and the potential energy both for the whole system and per atom.

Save the original positions of each structure in a separate array.

2.3 Task 3: Molecular dynamics

To create a variety of structural environments, we will run MD simulations at different temperatures in the NVT ensemble. Have a look at https://wiki.fysik.dtu.dk/ase/tutorials/md/md.html to get an idea of how to run MD using ase. A corresponding example is also given in the jupyter-notebook for this exercise.

Run an MD simulation for each of the four (fcc, bcc, hcp, li q) bulk structures with the following settings:

- set the initial positions to the original ones after lattice optimisation
- run fcc, bcc, and hcp at $T = 600$ K and liq at $T = 3000$ K
- set the initial velocities using the `MaxwellBoltzmannDistribution` with the corresponding temperature
- use an `Andersen` thermostat with a time step $\Delta t = 0.001$ and a collision frequency $\nu = 0.02$
- run the simulation for a total of 10 000 MD steps
- save a trajectory snapshot to a file every 100 steps (so that you will have 100 frames in total)
- print the potential, kinetic, and total energy per atom and the temperature every 500 steps

You can also convert the saved trajectory to an xyz-file and view it with your favourite visualisation software.

3 Defining descriptors (10 pt)

We will use the radial Behler-Parrinello symmetry functions (G_1, G_2, G_3) as input descriptors for our classification problem. The corresponding functions are given by:

$$G_1^{(i)}(\mathbf{r}) = \sum_{j \neq i} f_c(\mathbf{r}_{ij}) \quad , \tag{1}$$

$$G_2^{(i)}(\mathbf{r}) = \sum_{j \neq i} e^{-\eta(|\mathbf{r}_{ij}|-R_s)^2} f_c(\mathbf{r}_{ij}) \quad , \tag{2}$$

and

$$G_3^{(i)}(\mathbf{r}) = \sum_{j \neq i} \cos(\kappa|\mathbf{r}_{ij}|) f_c(\mathbf{r}_{ij}) \tag{3}$$

with the cutoff function

$$f_c(\mathbf{r}_{ij}) = \begin{cases} 1 & \text{if } |\mathbf{r}_{ij}| \leq r_{\min} \\ \frac{1}{2}\left(\cos\left[\frac{(|\mathbf{r}_{ij}|-r_{\min})}{(r_c-r_{\min})}\pi\right]+1\right) & \text{if } r_{\min} < |\mathbf{r}_{ij}| \leq r_c \\ 0 & \text{if } |\mathbf{r}_{ij}| > r_c \end{cases} \tag{4}$$

For each atom i, the symmetry functions are defined as the sum over the respective neighbours, that is to compute the symmetry functions, we require a list of neighbours and the corresponding distances.

3.1 Task 1: Setting up neighbour lists

Have a look at:
https://wiki.fysik.dtu.dk/ase/ase/neighborlist.html?highlight=neighborlist#ase.neighborlist.neighbor_list for information on neighbour lists. An example is also given in the jupyter-notebook of the exercise.

Read the trajectory of the MD simulation and take the final frame of each of the four bulk structures to setup neighbour lists with `cutoffs` of $3\mathring{A}$.

The neighbour list provides a list of arrays containing a variable amount of information thanks to its 1^{st} argument. What kind of information do these array contain? Which ones are necessary for the computation of the symmetry functions? List the distances between the 3^{rd} atom to all of its neighbors along with the neighbor indices.

3.2 Task 2: Symmetry functions

Write a function for each of the symmetry functions and the cutoff function. In addition to the list of distances, you also need to pass information concerning the parameters $r_{\min}$, r_c, R_s, η, and κ to the corresponding functions. The solutions should not include explicit `for-loops`.

3.3 Task 3: Computing symmetry functions

For fcc, bcc, hcp, and liq, compute and report the values of the symmetry functions G_1, G_2, G_3 for the 2^{nd} atom from the last frame of the MD simulations with the following set of parameters:

- $r_{\min} = 2.6$ and $r_c = 2.8$
- $R_s = 1.4$ and $\eta = 100$
- $\kappa = 3.5$