Due date: Wednesday, November 24, 2021

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

1 Neural network potential for Lennard-Jones clusters

In this worksheet, we develop a neural network (NN) potential for small Lennard-Jones (LJ) clusters. There will be three steps that will be worked on over the next three weeks:

(1) setting up the LJ clusters and creating several datasets (week 1)

(2) optimizing the hyper-parameters and training the NN (week 2)

(3) application, transferability, and limitations (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 Atomic simulation environment

To setup the cluster structures and evaluate the LJ potential, we use the 'Atomic Simulation Environment' (ASE, `https://wiki.fysik.dtu.dk/ase/`), a set of Python tools for atomistic simulations. ASE can do quite a few things, most importantly for the current task will be the 'Atoms' object (`https://wiki.fysik.dtu.dk/ase/ase/atoms.html`) and the LJ calculator (`https://wiki.fysik.dtu.dk/ase/ase/calculators/others.html#lennard-jones`). Have a look at the corresponding webpages for detailed information.

3 Initial cluster setup (10 pt)

Use the ase atoms object to setup the initial cluster structures. You will need to import 'Atoms':

```
from ase import Atoms
```

For a dimer oriented along the x-axis, the corresponding atoms object would, for example, look like:

```
dimer = Atoms('2Ar', [(-0.5, 0.0, 0.0), (0.5, 0.0, 0.0)]
```

3.1 Task 1: 2D and 3D clusters with 7 atoms

Create atom objects for 2 cluster. Both clusters consist of 7 atoms. For the 2D cluster, place the central atom at the origin and the other 6 atoms on the vertices of a hexagon around the central atom in the xy-plane. For the 3D cluster, place the central atom again at the origin and the other 6 atoms on the vertices of an octahedron. Set the initial distance between the central atom and its 6 neighbours to 1.1 Å.

Next, we want to compute the potential energy of the clusters using a LJ potential. Write down the functional form of the LJ potential for a system with N particles. Set the *ase* calculator to the LJ potential with the default setting. What is the meaning of the variables σ and ϵ in the LJ potential? What are the default values of σ and ϵ in ASE? Compute the potential energy for the initial configuration of both clusters (`get_potential_energy()`)and report the initial positions using `get_positions()` together with the potential energy. What is the unit of the potential energy?

3.2 Task 2: Optimizing the initial cluster structures

By minimizing the forces, the initial positions of the atoms in the two clusters can be optimized. Use the BFGS minimizer in ASE (https://wiki.fysik.dtu.dk/ase/ase/optimize.html?highlight=bfgs#ase.optimize.BFGS) and perform an energy minimization with a maximum force of 0.05.

For the 2D cluster, the atoms need to be constraint in the xy-plane. ASE provides a `FixedPlane` constraint that can be used to for this (see also the example in the jupyter notebook).

Report the positions of the atoms and the potential energy of the 2D and 3D cluster after the optimization. Visualize the clusters (e.g using `view` from `ase.visualize`, but you can also print out an xyz-file and use your preferred visualization program) and inspect the structures visually.

3.3 Task 3: Pairwise distances

It is often not the best choice to directly use the Cartesian coordinates of the atoms as input to the NN. What is the main disadvantage when directly using Cartesian coordinates?

As descriptors for the NN, we will use here a sorted list of pairwise distances, $d_{ij} = |\mathbf{r}_{ij}|$, between the atoms in the cluster, with $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$, and $\mathbf{r}_i$ is the position of atom i. Since $d_{ij} = d_{ji}$, we only need to record one distance for each pair of atoms. For a cluster with N atoms, how many pairwise distances without double counting are there? Correspondingly, for the 7 atom clusters, how many distances (NN input descriptors) do we have?

The function `get_all_distances()` returns an array with all distances between all atoms. Use this function and sort the distances, remove 0 entries for d_{ii}, and remove the double entries for $d_{ij} = d_{ji}$ to obtain a set of input descriptors. For both the 2D and 3D cluster report the input descriptors for the optimized structures.

4 Creating datasets using Monte Carlo sampling (10 pt)

To train the NN, a number of different configurations of the clusters is required to sample the potential energy surface of the clusters. Here, we will use Monte Carlo (MC) sampling at different temperatures to create a set of meaningful configurations, that is configurations that have a significant contribution to the partition function in the canonical ensemble. An MC step consists of two parts: (i) propose a trial configuration and (ii) accept/reject the trial configuration based on the detailed balance criterion. If the trial configuration is rejected, the old configuration is recounted in the ensemble.

To create a trial configuration $\mathbf{r}^{(\mathrm{n})}$ from the old one $\mathbf{r}^{(\mathrm{o})}$, a displacement is added to all atoms in the cluster

$$\mathbf{r}_i^{(\mathrm{n})} = \mathbf{r}_i^{(\mathrm{o})} + \Delta\mathbf{r} \qquad (1)$$

where the displacements $\Delta\mathbf{r}$ are Gaussian distributed random number with 0 mean and a standard deviation σ_G (width of the Gaussian, controls the magnitude of the displacement). The acceptance probability for the trial configuration is

$$p_{\mathrm{acc}}(\mathrm{o} \to \mathrm{n}) = \min\left(1, \exp\left(-\beta(E_{\mathrm{pot}}^{(\mathrm{n})} - E_{\mathrm{pot}}^{(\mathrm{o})})\right)\right) \quad , \tag{2}$$

with $\beta = 1/(k_{\mathrm{B}}T)$. An MC step thus includes the following:

1. compute energy of old configuration and save old positions
2. create new configuration by adding a random displacement to all atoms according to Eq. (1)
3. compute energy of new configuration
4. accept/reject new configuration according to Eq. (2)

4.1 Task 1: MC sampling function

Write a python function that performs an MC step. In addition to the Atoms object, the function should take information about the temperature T, the magnitude of the displacement σ_G, and the dimensionality D (for the 2D cluster, positions should only be displaced in x and y!). To set new positions of an Atoms object, the function `set_positions()` can be used. Also, make sure that you record if a trial move was accepted or rejected.

4.2 Task 2: MC sampling and dataset for 3D LJ cluster

The acceptance ratio is the ratio between accepted MC moves and the total number of MC steps, $n_{\mathrm{acc}}/n_{\mathrm{MC}}$. A good sampling is usually achieved if the acceptance ratio is around $0.4 - 0.5$ and depends on the temperature and the displacement magnitude. For lower T, the displacement magnitude σ_G has to be chosen smaller to achieve a suitable acceptance ratio.

Starting with the initial optimized configuration of the 3D cluster, run 20 000 MC steps at a temperature $T = 10$ K. Find a value for σ_G such that the acceptance ratio is around 0.5. Report the chosen value for σ_G. Print the energy every 500 steps. To collect the dataset, compute the corresponding descriptors (list of distances) and save them in an array $\mathbf{X}$ together with the potential energy in an array $\mathbf{y}$. Collect a configuration every 20 MC steps, so that your dataset contains a total of 1000 data points.

Starting again with the initial optimized configuration, collect another dataset at $T = 800$ K. Make sure to adjust σ_G. Run again 20 000 MC steps, print the energy every 500 steps and collect your data every 20 steps.

For both datasets at $T = 10$ and 800 K, plot the energy vs. the index of the configuration.

4.3 Task 3: MC sampling and dataset for 2D LJ cluster

Similar to the datasets for the 3D cluster, collect 1000 data points for the 2D cluster using MC sampling. Create two datasets ($\mathbf{X}$,$\mathbf{y}$), one at $T = 200$ K and one at $T = 2000$ K. Make sure that you only displace the atoms in the xy-plane!

For both datasets at $T = 200$ and 2000 K, report the chosen σ_G value and plot the energy vs. the index of the configuration.