Variational Auto Encoders

Meeting 1 - 09/05/23

Group 9 - Bacilieri, Barbiero, Bordin, Pitteri

Generating the dataset

Potential Energy

Fluid in a cubic container of side L

Potential energy: - gravitational potential $U_g = mgr_z$

- Lennard Jones Potential $U_{LJ} = 4\epsilon[(\frac{\sigma}{r})^{12} - (\frac{\sigma}{r})^6]$

A change of variables is performed $r' = \frac{r}{\sigma}$

And so $U' = 4[(\frac{1}{r'})^{12} - (\frac{1}{r'})^6] - \frac{mg\sigma}{\epsilon} r'_z$

The only trainable parameter is $\gamma = \frac{mg\sigma}{4\epsilon}$

Generating the dataset

Algorithm

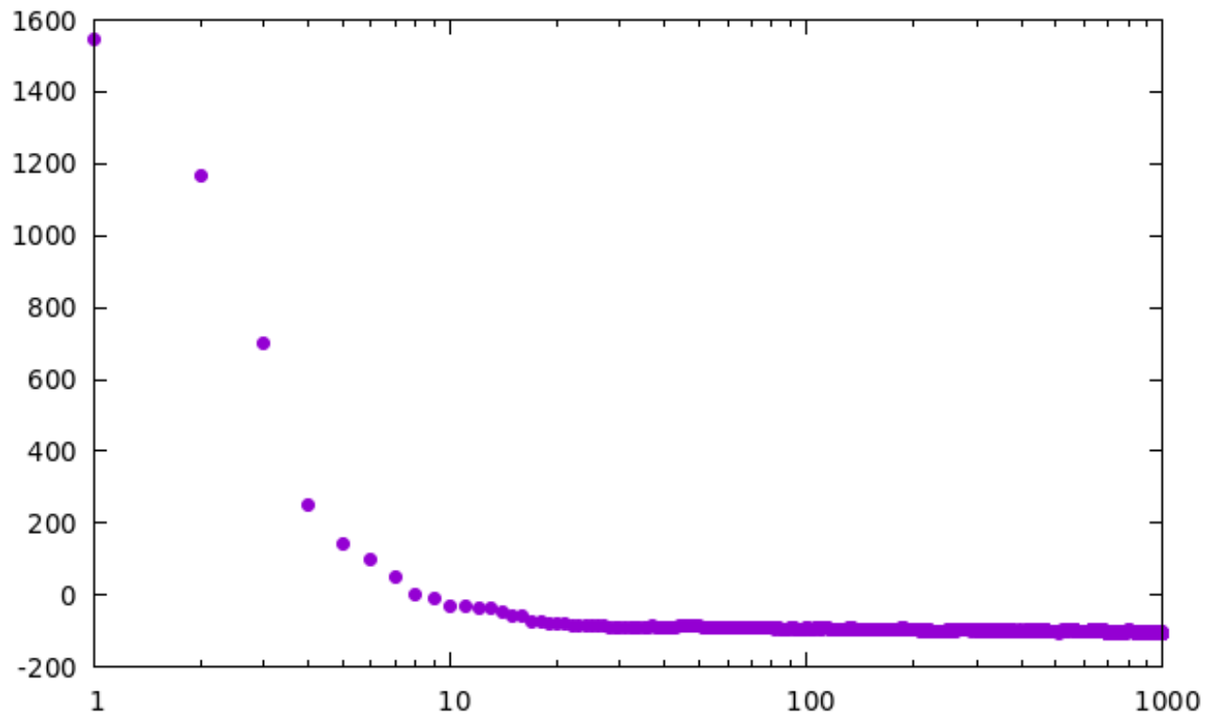
Data is generated in C++ with the Metropolis Algorithm

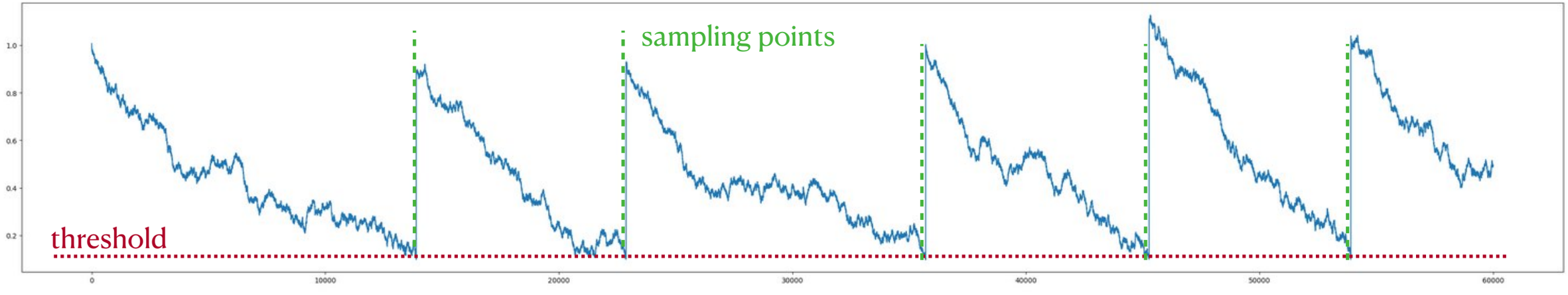
1. Start from a randomized configuration
2. Move one particle at random (kick)
3. if $\Delta U \geq 0$ the move is always accepted, else it is accepted with probability $p = \exp \frac{-\Delta U}{T}$
4. All moves that would end up outside the boundary are reflected back inside the box
5. Results are produced in cartesian coordinates

Generating the dataset

Results

After an initial transient phase the energy reaches, as expected, an equilibrium. From there we can start sampling data, after checking that the correlation between two consecutive state is below a set threshold (10% at the moment).





Generating the dataset

Optimization

In order to avoid huge outputs the mean time for decorrelation (MTFD) for a given sigma is computed in a small evolution then only data sampled at such intervals is saved for analysis

PRELIMINARY RUN

Compute all positions for a short time

Save all position

Compute correlation

Estimate MTFD

SAMPLING RUN

Compute for longer times

Save only positions for multiples

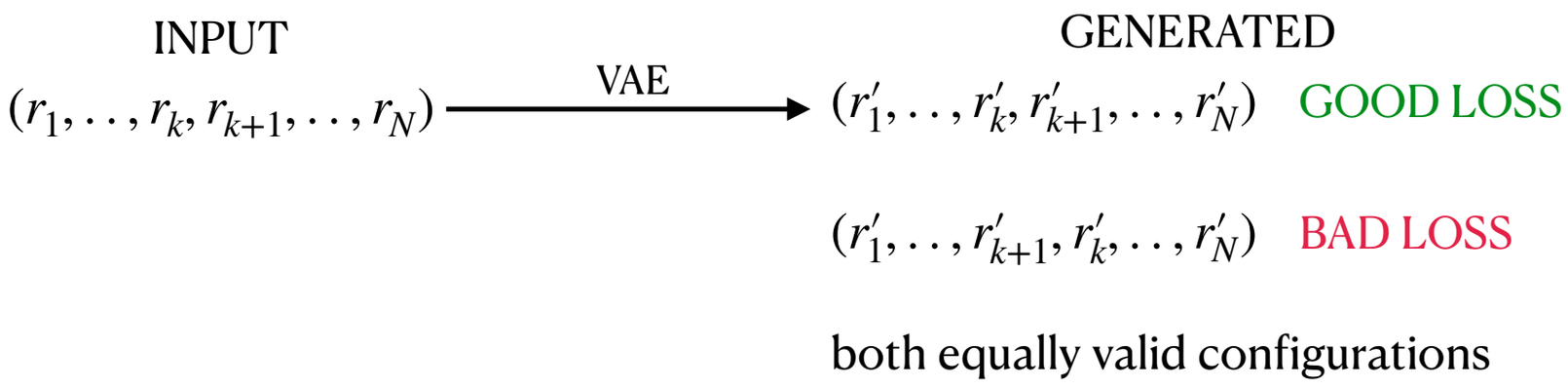
of MTFD

Variational Auto Encoder

General Framework

The main source of inspiration was the guide available in Keras' official website.

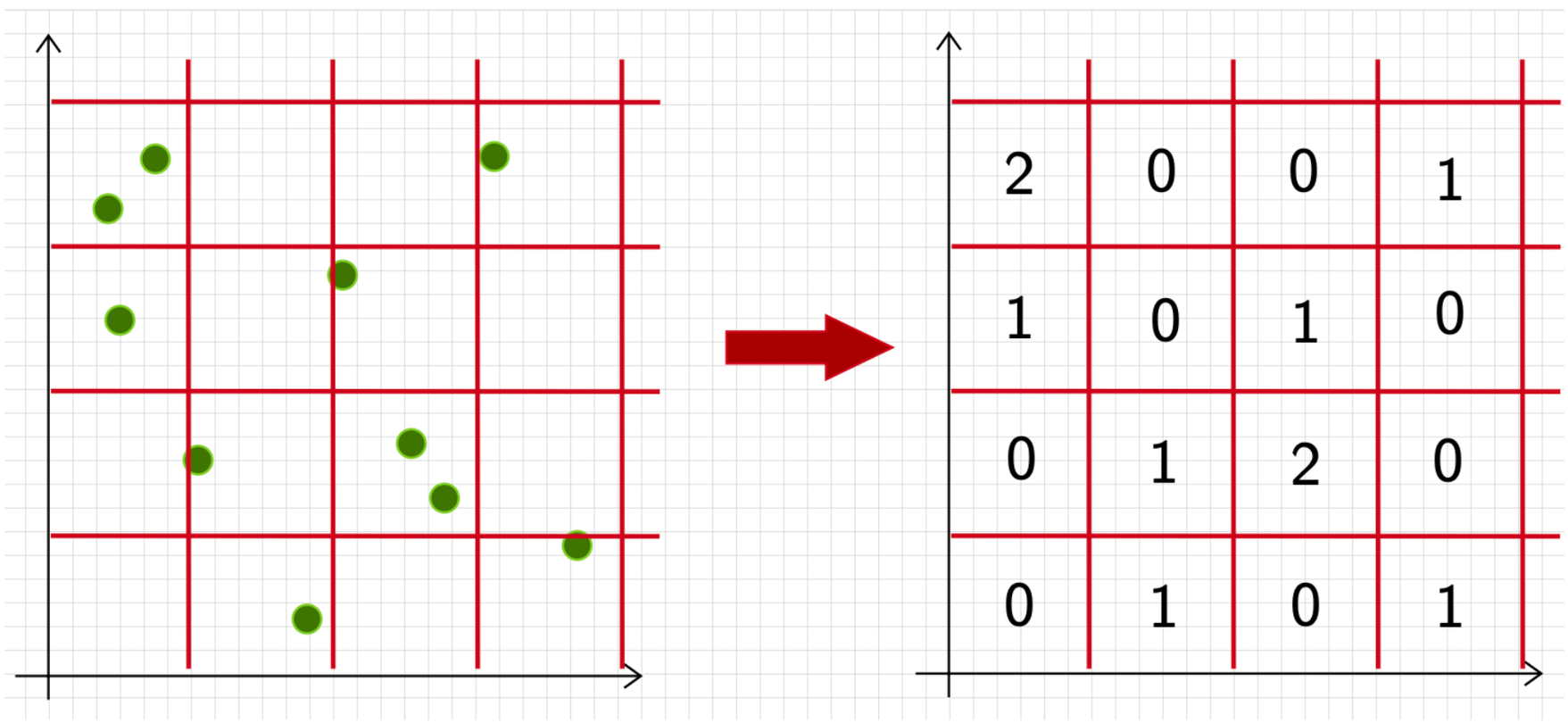
The first idea was to use a standard feed forward NN, this however turned out to be a bad choice since the learner would be sensible to the order of the tuples of the configuration



Variational Auto Encoder

General Framework

The next logical step is to find a way to decouple the positions of the particle from their order in the vector. This is achieved by a density measure in different regions of the box



Variational Auto Encoder

Input shape

After some manipulation the input for the VAE will be a tensor of shape $(n, n, n, 1)$ where

A. $n = \frac{L}{h}$ with h size of the cell (possibly optimizable)

B. Number of points in each cell are normalized to (0,1) range (value $= \frac{\#N}{\#N_{max}}$)

Note that the size/complexity of the input does not increase with the number of simulated particles, promising for both flexibility and scalability

See you next week