Project 4: Ising Annealing

July 22, 2020

Motivation

If we go back to a couple of weeks ago to Projects 1 and 2, you were able to simulate molecules in a couple of different ways: using machine learning (Restricted Boltzmann machines) and using the Variational Quantum Eigensolver (VQE). In Project 1, you were able to see that machine learning gives a huge advantage in data-driven problems in terms of outputting a compressed version of something quite large (exponential scaling – 2^{100} – versus a few hundred numbers). Not only this, you saw that if we needed to collaborate with an experimentalist in a lab, machine learning doesn't demand much from experimentalists (we don't need that much data from them). In Project 2, you saw that solving chemistry problems via simulation on a quantum circuit also has compression benefits, but we didn't need any input data to actually solve our problem like in Project 1. Rather, the *variational method* just requires a tunable *trial solution* to our problem that we then change based on how good of an energy it gives.

At the end of the day, VQE and machine learning are both extremely useful in different settings, but at heart of each is an *optimization problem.*[1] Sometimes this optimization problem can be extremely difficult to tackle. What if we can bypass this entirely? If we can bypass the optimization problem, do we lose anything? It turns out that answer to both of these questions is yes!

This week, you'll be exploring the well-known simulation method called Monte Carlo (MC). As we alluded to, Monte Carlo simulations do not require solving an optimization problem. However, at the end of a Monte Carlo simulation we do not obtain a compressed version of something much larger; we cannot run a Monte Carlo simulation, store the "output" on a USB stick, and then easily sample it (as in Projects 1 and 2). That being said, Monte Carlo simulations can be *very, very* efficient and fast.[2] This week, your tasks will involve the use of a single-spin-flip Metropolis-Hastings (MH) MC simulation. Let's dive into your tasks!

Your Tasks

This week, you will be tasked with performing various MC simulations on the Ising model,

$$H = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j, \tag{1}$$

where σ_i are classical spin-1/2 variables (-1 or 1). You will be walked through a simulation we've provided along with an annealing procedure. Then, you'll be given slight variations to the Ising model where you must come up with an annealing procedure yourself!

[1] In machine learning we must optimize the cost function and in VQE we must optimize the energy.

[2] You now see the daily struggles of computational scientists: which algorithm or method do I use to solve my problem?!

Task #1

Navigate to the `Task_1.ipynb` notebook. We've given you a full solution to a MC simulation employing the MH algorithm for a 2D ferromagnetic ($J > 0$) on a square lattice with periodic boundary conditions. Code blocks 1 to 5 outline a MC simulation for this system for a temperature $T = 1.0$. After this, we now wish to perform a simulation at a temperature $T_{\text{final}} = 0.01$ using an annealing procedure starting at $T_{\text{initial}} = 100$. Here, we employed an annealing procedure that decreases T_{initial} exponentially (see code block 7). Run this notebook and make sure that you understand what is going on under the hood (see `abstract_ising.py` and `ising_animator.py`).

Task #2

Navigate to `Task_2.ipynb`. Here's what we'd like you to do!

1. Having understood Task #1, we're now going to task you with finding an annealing procedure for the random-bond Ising model in 1D,

$$H = J \sum_{\langle i,j \rangle} B_{ij} \sigma_i \sigma_j, \tag{2}$$

   where $B_{ij} = \pm 1$. Do this for various system sizes (10, 20, 50, 100).

2. The fully-connected random bond Ising model

$$H = J \sum_{i<j} B_{ij} \sigma_i \sigma_j, \tag{3}$$

   is even a little harder to perform a good MC simulation for low temperatures. Find an annealing procedure for this model, as well.

We've given you the required classes in order to get your started on these models in `Task_2.ipynb`. In your annealing procedures, start with as high of a starting temperature as needed and get to as low of a temperature as you can!

Further Challenges

1. In `Task_2.ipynb`, we also gave you some code to get you started on determining an annealing procedure for the fully-connected Mattis model. Try finding an annealing procedure for this model at various system sizes.

2.

References