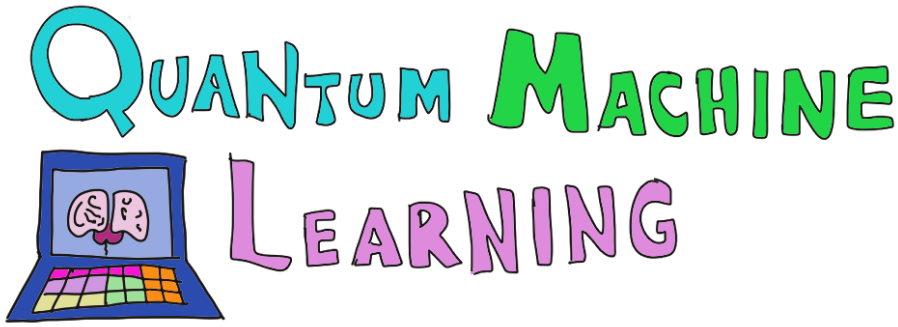

QML: Ising on the Cake [300 points]

Version: 1

Quantum Machine Learning

In this set of challenges, you'll explore methods and applications for training variational quantum circuits and quantum neural networks. Both play critical roles in quantum machine learning. They typically have a layered structure, and a set of tunable parameters that are learned through training with a classical optimization algorithm.

The circuit structure, optimization method, and how data is embedded in the circuit varies heavily across algorithms, and there are often problem-specific considerations that affect how they are trained. In the **Quantum Machine Learning** challenges, you'll explore how to construct and optimize a diverse set of circuits from the ground up, and become acquainted with some of today's popular quantum machine learning and optimization algorithms.

Problem statement [300 points]

Classification tasks are now commonplace in the world of machine learning, both classical and quantum. In this challenge, you will be constructing a quantum variational classifier that can learn the phases of the Ising model with inspiration taken from Ref [1]. The Ising model, whose Hamiltonian is given by

$$H = -\sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z,$$

is a classical toy model that describes magnetism via nearest-neighbour interacting binary spins. There exists a finite-temperature phase transition for the Ising model where, below the "critical" temperature (the temperature where the phase transition occurs), favoured spin-configurations are all up/down (the "ordered" phase). Above the critical temperature, favoured spin configurations are random (the "disordered" phase). See Figure 1.

Figure 1: The Ising model phase transition

Your job is to create a variational classifier that can distinguish between configurations from the "ordered" and "disordered" phases with an accuracy of more than 90%. The input data we've provided you with contains 250 spin configurations for 4 spins (0 and 1 for each spin) and labels for each configuration: 1 (-1) for ordered (disordered). Specifically, your code will do the following:

- Define a variational circuit.
- Define a protocol for how to train the circuit for the given learning task.

The provided template file `ising_classifier_template.py` contains a couple of functions:

- `accuracy`: This returns the accuracy of your model's predictions compared to the actual labels. Use this to benchmark your code. 90% accuracy gets you 300 points!
- `classify_ising_data`: This is the function that will execute your training protocol (you may `break` your training loop once your accuracy gets above the tolerance if you wish).

Input

- `list(int)`: A list of Ising spin configurations and labels indicating which configuration belongs to which phase.

Output

- `list(int)`: A list of model-predicted labels.

Acceptance Criteria

In order for your submission to be judged as "correct":

- The outputs generated by your solution when run with a given `.in` file must match those in the corresponding `.ans` file to within the `Accuracy` specified below.

- Your solution must take no longer than the `Time limit` specified below to produce its outputs.

You can test your solution by passing the `#.in` input data to your program as stdin and comparing the output to the corresponding `#.ans` file:

```
python3 ising_classifier_template.py < 1.in
```

WARNING: Don't modify the code outside of the `# QHACK #` markers in the template file, as this code is needed to test your solution. Do not add any print statements to your solution, as this will cause your submission to fail.

Specs
Tolerance: $\geq$ **90%**
Time limit: **90 s**

References

[1] Carrasquilla, J., Melko, R. Machine learning phases of matter. Nature Phys 13, 431–434 (2017). https://doi.org/10.1038/nphys4035

Version History

Version 1: Initial document.