Quantum Hopfield Networks

Mini-project for ID5841 Quantum Lab course

Rajarsi Pal

Indian Institute of Technology, Madras
e-mail: ph20c032@smail.iitm.ac.in
git: `https://github.com/pafloxy/QHAM21_project`

ABSTRACT

In this work I have provided a review of the work presented by N.Miller et al. on Quantum Hopfield Associative Memory (QHAM) [MM21], along with a brief overview of various methods of implementing quantum neural networks. I have discussed the results to some of the experiments that I have conducted on my own implementation of the QHAM and also commented on the various issues that hinders the functioning of this model.

Keywords: Quantum Neural Networks, Machine Learning , Pattern Retrieval, Associative Memory

INTRODUCTION

Hopfield networks came up originally as models for content addressable memory, referring to their ability to regenerate corrupted form of a stable pattern, also their use in solving quadratic optimisation problem were well investigated by many authors, for more details refer to [Hop82], [JD85]. Here we will give a brief overview of classical Hopfield neural networks and then move on to the idea of a Quantum Hopfield network as introduced by N.Miller et al[MM21].

Network architecture of Hopfield model

Hopfield networks fully-connected recurrent network, where every neuron affects every other neuron on their next time step through their weights.

Say the activation of a particular neuron is $X_i(\mathbf{X})$ and corresponding bias $b_i(\mathbf{b})$, then the energy of configuration is given by $E(X) = -\mathbf{X^T W X} + \mathbf{b^T X}$, where $\mathbf{W}$ is the weight matrix. Now any neuron can be updated as, $X_i(t+1) = g(-\frac{\partial E}{\partial X_i}) = g(\mathbf{W_i}.\mathbf{X(t)} - \mathbf{b})$, $\mathbf{g}$ being the non-linear activation function.

This particular update strategy guarantees that if $\mathbf{W}$ is symmetric and has diagonal elements zero then **Energy** decreases monotonically with iterations, and as there are only a finite number of configurations available the network configuration must converge to an energy minimum.

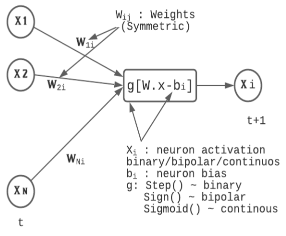


Figure 1. Structure of Hopfield Networks

Problem in designing a quantum neuron

The most fundamental unit for neural computation is a perceptron. A perceptron simply involves mapping the given input vector to single output through a linear transformation followed by a non-linear transformation. The main difficulty in designing an quantum neuron comes from the step of imposing a non-linear activation function on the neurons, since in quantum mechanical formalism state evolution is inherently unitary there exists no non-trivial way to implement a non-linear operation on the neurons.

Some of the techniques to get past this problem involves **measurement based methods**, where the basic idea is to introduce ancillary qubits to our system and and measure them at subsequent stages of the circuit (**corresponds to tracing out a sub-system mathematically), measurement being a non-linear operation. Also there are other methods to introduce non-linear activation function like sigmoid, ReLU, tanh, etc. based on basis encoding, the most useful of them being **Quine-McClusky's Method**. A more technical description could be found at [Sch18]
For our model we will use the scheme of **rotation encoding**. The basic idea here is to use controlled rotation (preferably $CR_y(\phi)$) where the values corresponding to the data v is encoded in the rotation angle $\phi = \phi(v)$, as shown in figure 2.

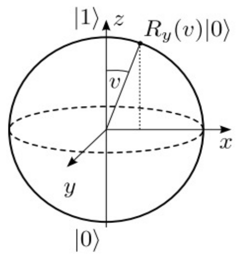


Figure 2. Rotation Encoding

MILLER'S MODEL [MM21]

Design and Working principle

In hopfield model every neuron can take value of either $\{\pm 1\}$, for developing a quantum hopfield model we first need a way to encode our statevector into a quantum state, for our purpose we use the following strategy,

$$|\psi(x_i)\rangle = cos(\frac{\pi}{4}x_i + \frac{\pi}{4})\,|0\rangle + sin(\frac{\pi}{4}x_i + \frac{\pi}{4})\,|1\rangle \tag{1}$$

where $x_i \in \{+1,-1\}$. Thus for any state $\vec{x} = \{x_1, x_2, x_3..\}$ of the hopfield network there exists a corresponding product state $|\psi(x_1,x_2..)\rangle = |\psi(x_1)\rangle\,|\psi(x_2)...\rangle$.

The trick to update our model is to use controlled rotation gates $CR_y(\phi)$ where the rotation angle ϕ is dependent on the elements of the *weight matrix* W_{ij}, apart from the weight matrix we also add a bias term to every qubit. So, updating a single qubit involves controlled rotation on that qubit conditioned on every other qubit and a single qubit unitary depending on the bias term, see figure 3.

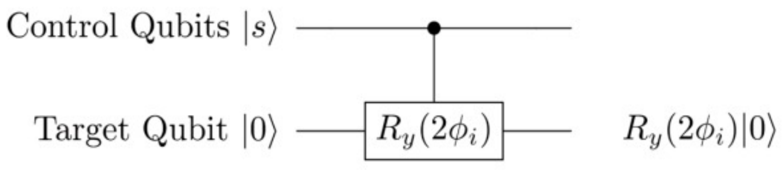


Figure 3. Updating $|target\rangle$ using $|control\rangle$

The details of implementation for the weight and bias terms could be summarised into follows. see figure.. However, since the qubit to be updated must be in the $|0\rangle$ we can not directly use the $CR_y(\phi)$ to update the target qubit, instead we introduce an ancillary qubit initialised to $|0\rangle$ on which the controlled rotations act, and then swap it with target qubit. If we are using quantum devices which support reset operations the above can be implemented with a single ancilla qubit, otherwise we will require as many ancilla qubits as the number of updates, however in both cases the qubit overhead is only linear in the number of updates, for example see figure 4.

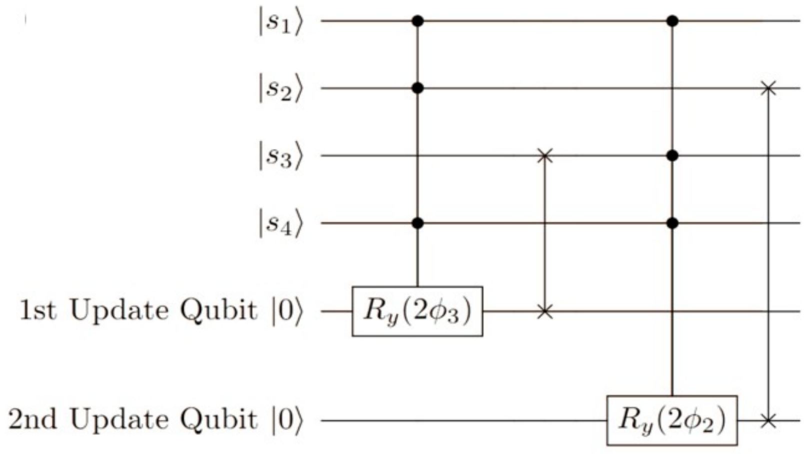


Figure 4. A network of $n = 4$ qubits, we are updating qubits $|s_3\rangle$ and $|s_2\rangle$ using two different ancillaries initialised to $|0\rangle$. Note the use of **SWAP** operation following every $\mathbf{CR_y}$ operation.

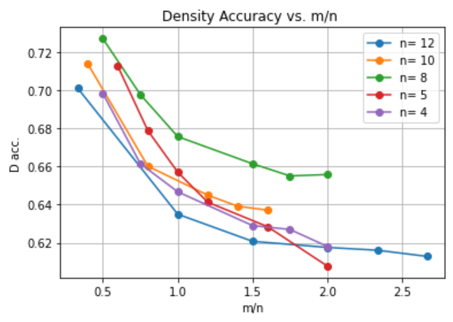
To get the classical representation of the state of the network we might adopt the majority vote method as discussed in the original paper, which simply states that replace the quantum state by $|1\rangle$ if for that particular qubit $P(|1\rangle) > 0.5$, or $|0\rangle$ otherwise. Whether it is more beneficial to apply majority vote after each update or after a certain number of updates is still an open-ended question, and will be discussed in a later section.

EXPERIMENTS

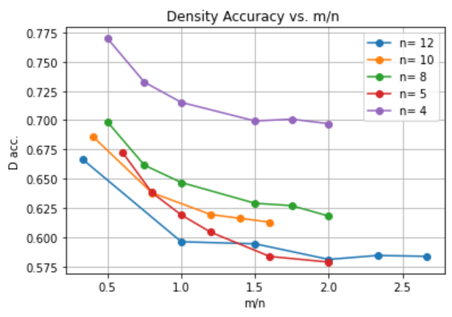
E.1 Testing effective retrieval capacity of QHAM

Here I tested how much effectively could the network retrieve patterns from the set of stored patterns, for our purpose effectiveness was quantified by **Density Accuracy**, *DAc* which describes the average rate of each of the individual qubits being measured in the correct state. Since there was no particular choices for the no. of updates prior the experiment, I also experimented with update steps varying from **n** to **2n**,where **n** is the number of qubits to observe variation in *DAc*. Note that the qubits were only measured at the end of the circuit.

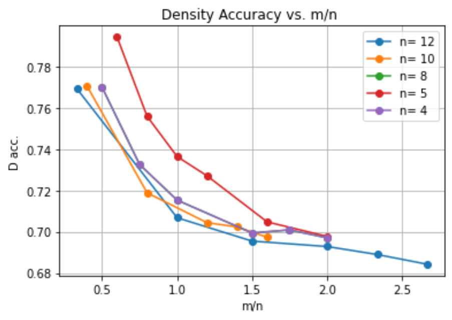
E.1.1 Here we stored m random patterns on n qubits and the variation of DAc against the ratio $\frac{m}{n}$. The cases for n=2,5,8,10,12 was explored here, the no. of updates u was kept at three different settings of n, $2n$ and also to an intermediate value between them, as shown in figure 5. Each instance of this experiment was run for 50 times to obtain statistical accuracy in our measurements.
The cases for n=4, 5 were run on `ibm_lima`, other cases were run were on `QASM` simulator with noise model imported from `ibm_quito`.



(a) case: random u



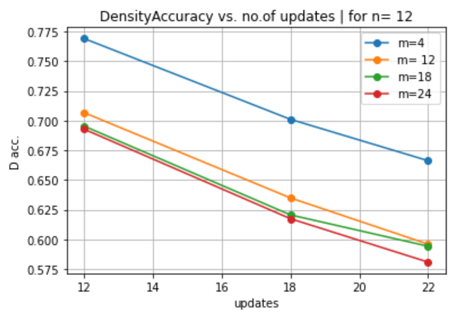
(b) case: $u = 2n$



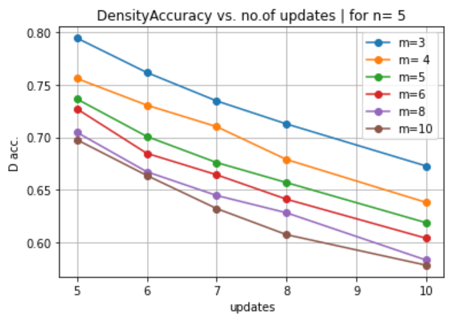
(c) case: $u = n$

Figure 5. Variation of DAc with $\frac{m}{n}$

E.1.2 Here we directly tested the variation of DAc against no. of updates u for the cases of n= 5, 12 qubits, against various values of m as shown in the figure 6



(a) case: $n = 12$

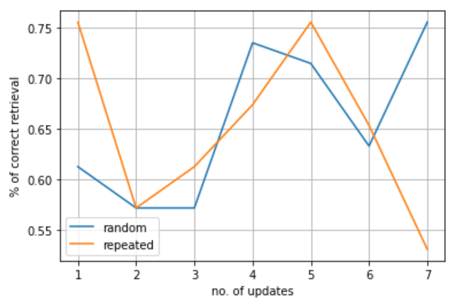


(b) case: $n = 5$

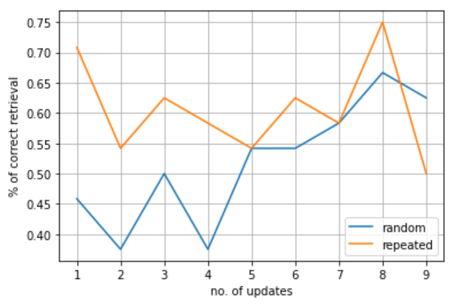
Figure 6. Variation of DAc with u

E.2 Testing effect of mid-circuit measurements on retrieval efficiency

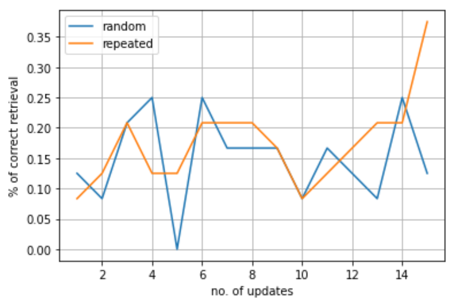
Here we took into account the fact measuring the the qubits immediately after an update step might help increase accuracy of the circuit. So for this experiment we implement two generically different type of circuits, one is where we do measurements *repeatedly* after each update steps and another is where we postpone measurements unitill the end of the circuit, refer to figure 7



(a) case: $n = 4$



(b) case: $u = 5$



(c) case: $u = 8$

Figure 7. Variation of accuracy with no. of updates

DISCUSSION

Issues

- Theoretically, the maximum number of patterns that could be stored scales as $m = 1.6n$, where n is the number of qubits. At least, for the case of of *u*=*n* the *DAc* is greater than 0.7 at $\frac{m}{m} = 1.5$ for all cases of *n* discussed here. This verifies that the fact that this QHAM has retrieval efficiency similar to that of a classical hopfield network.

- The variation of *DAc* with *n* is not very clear, but it is apparent that *DAc* is always lower for higher *n*, however this statement does not hold true for the whole range of *m/n*. As can be seen in figure 6

- From the results of experiments **E.1.1** and **E.1.2** it is evident that the quality of the retrieval process only degrades with increasing number of updates. However it should be noted that each of the qubits must be updated at least once for the retrieval process to yield anything meaningful.

- One should note that the in contrast to the classical setting updating a qubit corresponds to applying a non-trivial unitary operation which are in general non-commuting, thus one might expect that the order in which the qubits are updated is supposed to affect the retrieval process significantly. However, in my implementations no significant among the both were observed.

- The experiment **E.2** was based on our expectation that the by projecting the states to computational basis states repeatedly would make the dynamics analogous to classical hopfield networks and thus would ensure a better retrieval, however the results of our experimentation doesn't support this claim as could be seen in figure 7. I was unable to deduce anything meaningful from this plots, neither provide an explanation against this observations.

Challenges

- By taking majority vote of a given state onto the computational basis states we are loosing essential information regarding the quantum mechanical state of the system, thus the working principle of the model is not truly quantum in this sense. Though the original paper remarks the fact, that states could be in superposed states could be used to enhance the retrieval capacity, the exact way to implement the same was not discussed.

- From the point of view of performance, the efficiency of retrieval process does not seem to be better than that of classical hopfield networks, neither any evidence of real quantum-advantage over the classical model was observed. Also, the quantum model lacks the simple fixed point dynamics of the classical hopfield model which guarantees convergence of the state to one of the attractors, whereas here we saw that an increased number of update steps could rather lead to a lower accuracy of the retrieval process.

- Another issue with this quantum model is that it is not possible to scope into the inherent dynamics of the network without disturbing the state-evolution using measurements. Whereas in the case of classical hopfield networks we could keep track of quantities like *energy* and *hamming distance* from stored patterns to scope into the state of the system. To devise methods to obtain such metrics of the state in the quantum case would be a valid goal and should be investigated further.

REFERENCES

[Hop82] J J Hopfield. "Neural networks and physical systems with emergent collective computational abilities." In: *Proceedings of the National Academy of Sciences* 79.8 (1982), pp. 2554–2558. DOI: 10.1073/pnas.79.8.2554. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.79.8.2554. URL: https://www.pnas.org/doi/abs/10.1073/pnas.79.8.2554.

[JD85] J.Hopfield and DW.Tank. "Neural Computation of Decisions in Optimization Problems". In: *February 1985,Biological Cybernetics 52(3):141-52* (1985). DOI: 10.1007/BF00339943.

[Sch18] Petruccione Francesco Schuld Maria. *Supervised Learning with Quantum Computers*. 1st. Springer Publishing Company, Incorporated, 2018. ISBN: 3319964232.

[MM21] N.E. Miller and S. Mukhopadhyay. "A quantum Hopfield associative memory implemented on an actual quantum processor". In: *Scientific Reports* (Dec. 2021). DOI: https://doi.org/10.1038/s41598-021-02866-z. URL: https://rdcu.be/cMb1H.