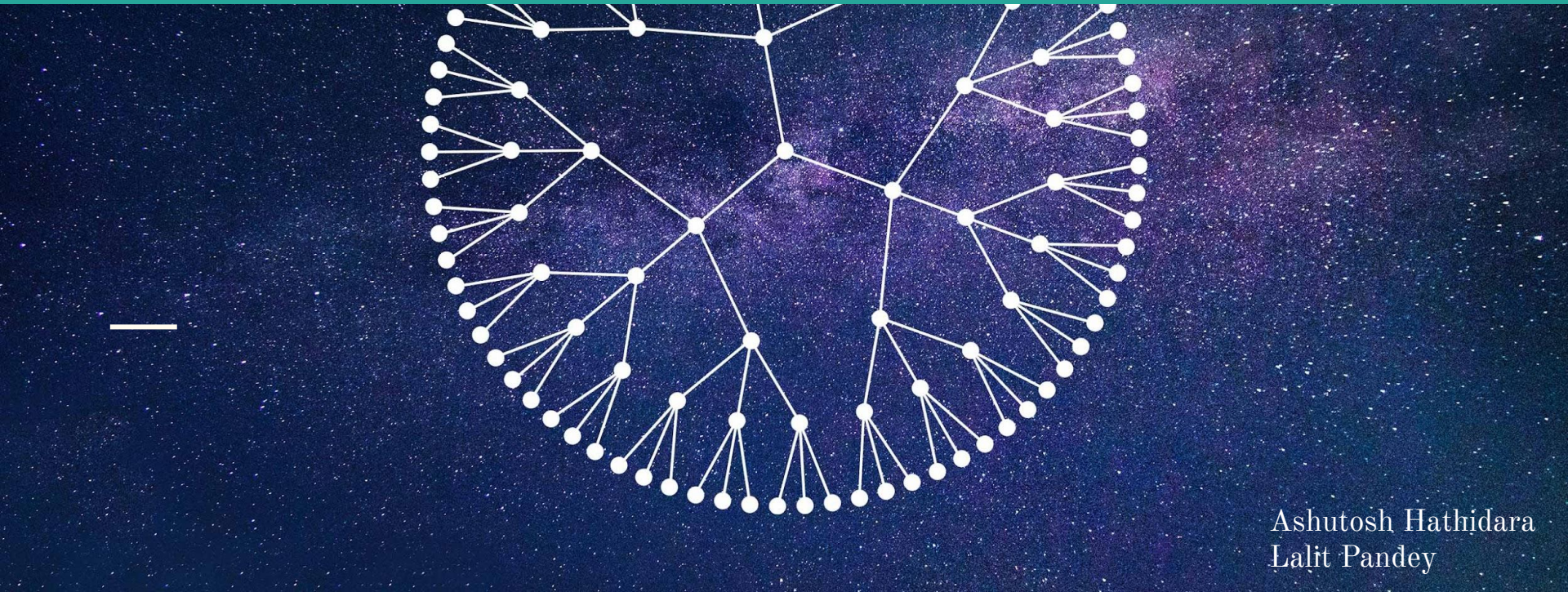
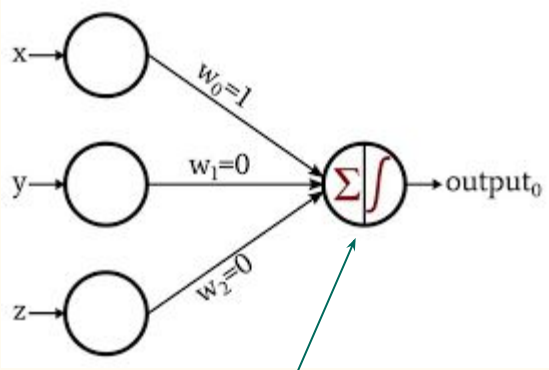
An Artificial Neuron Implemented on an Actual Quantum Processor



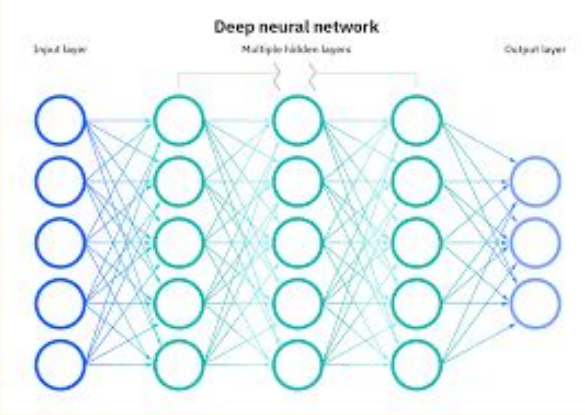
Ashutosh Hathidara
Lalit Pandey

Classical Perceptron



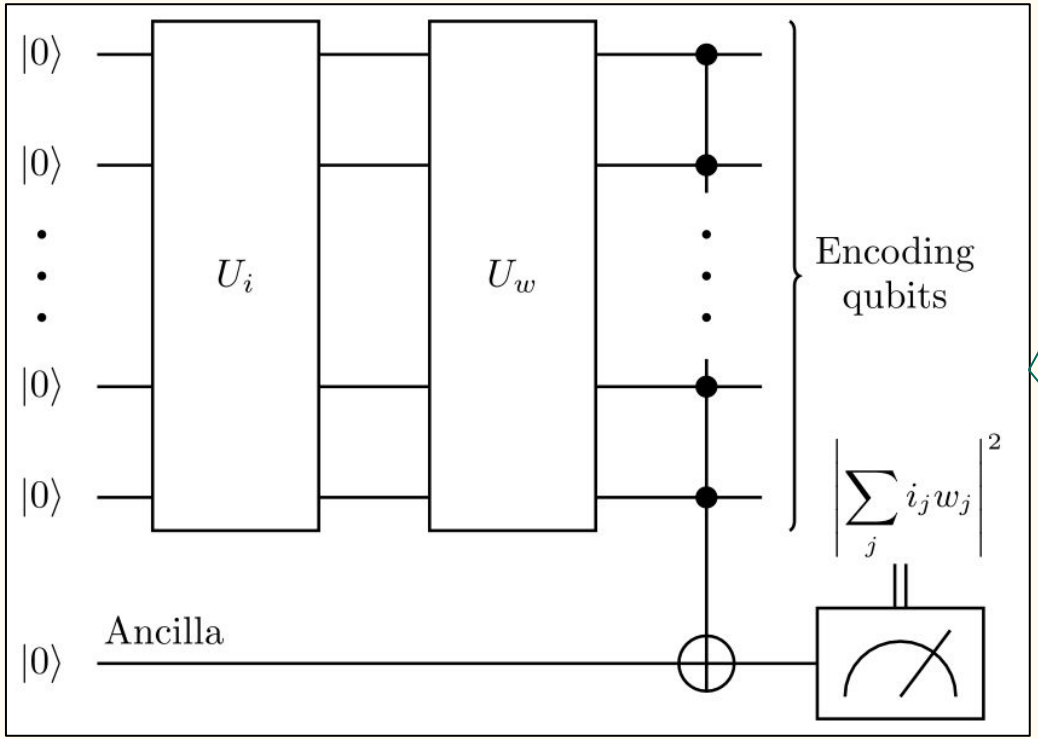
A single Perceptron

Classical Neural Network



A Deep Neural Net
(Multiple Perceptrons)

Quantum Perceptron



A quantum perceptron has:

- A Unitary Function to prepare the **input states**.

$$\vec{i} = \begin{pmatrix} i_0 \\ i_1 \\ \vdots \\ i_{m-1} \end{pmatrix}$$

$$|\psi_i\rangle = \frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} i_j |j\rangle$$

- A Unitary Function to prepare the **weights** for the perceptron.

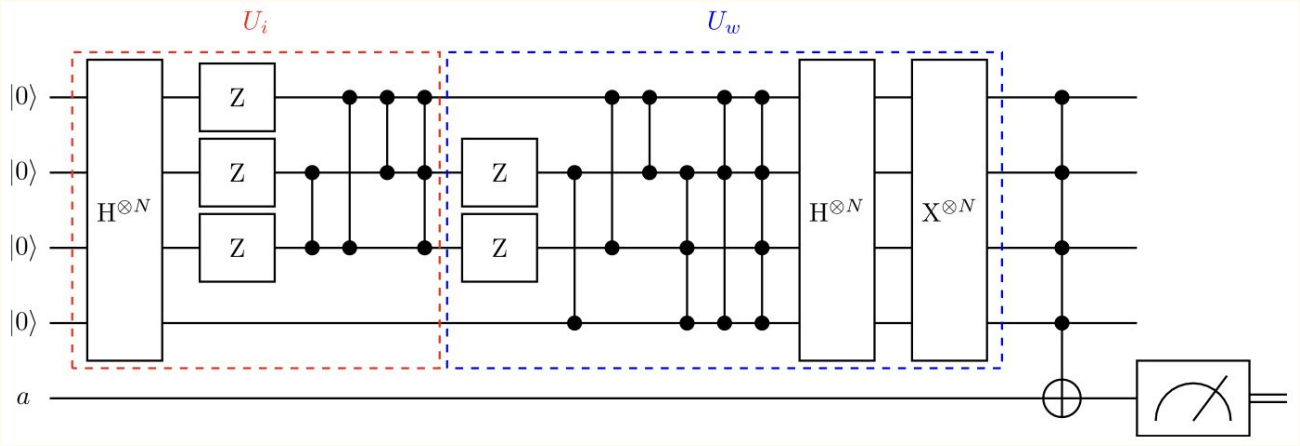
$$\vec{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{m-1} \end{pmatrix}$$

$$|\psi_w\rangle = \frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} w_j |j\rangle$$

Quantum Perceptron Implementation

Unitary Function U_i

Ui applies Z gate when the input vector element corresponding to state is -1.



Unitary Function U_w

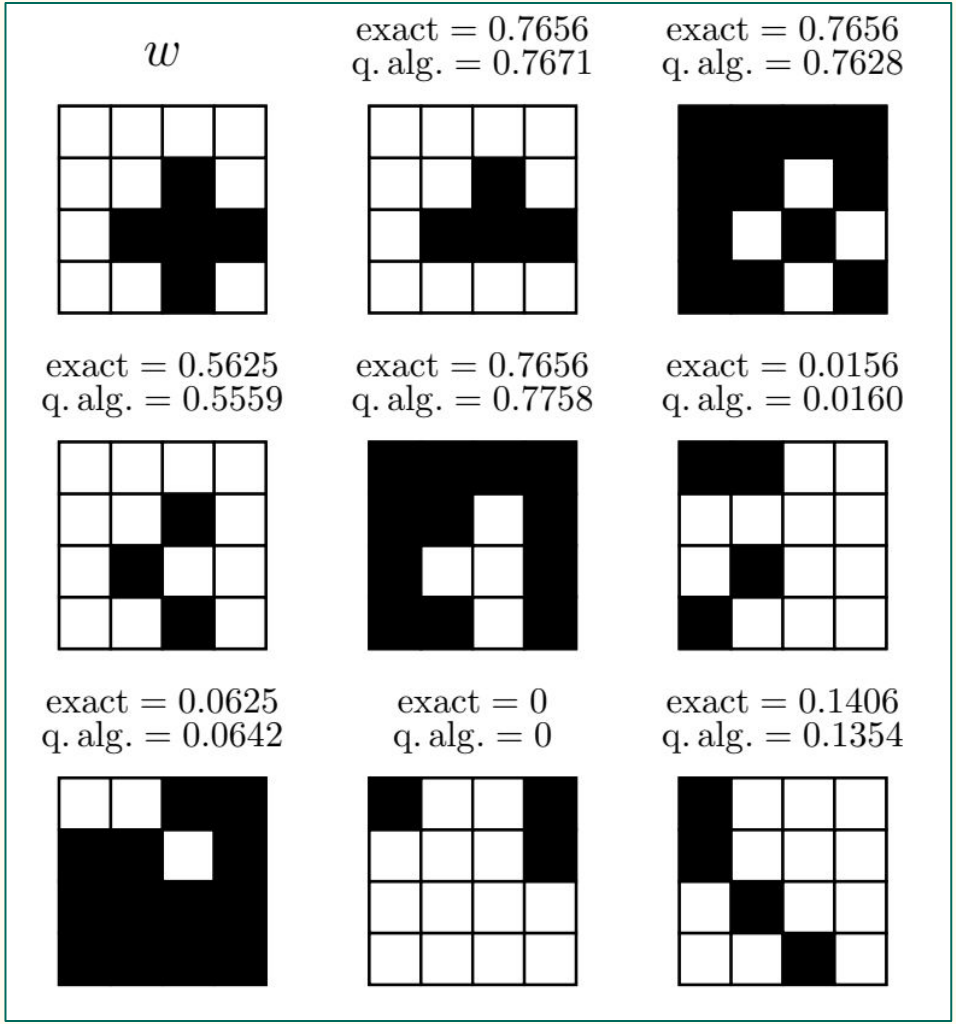
Uw applies Z gate when the weight vector element corresponding to state is -1.

Application of Quantum Perceptron

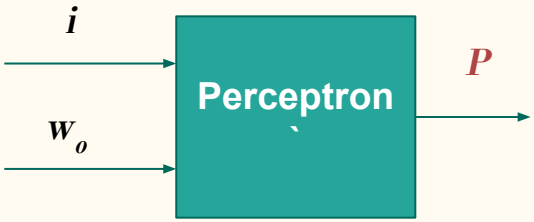
Pattern Recognition

- Remarkably, the quantum perceptron model can be used to sort out simple patterns such as horizontal or vertical lines among all possible inputs

Pattern Recognition for N=4



Quantum Perceptron: Data Preparation, Training and Evaluation



$$P \begin{cases} < 0.5 \text{ , label } 0 \\ > 0.5 \text{ , label } 1 \end{cases}$$

TRAINING A QUANTUM PERCEPTRON

① Let input $i = 12$,

then, $i = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}_{\text{classical}} \Rightarrow \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}_{\text{quantum}}$, $i \in \{-1, 1\}$

② If qubits = 2, it can handle:

$2^n = 4$

ex: 0 0 0 0 → 0

⋮

1 1 1 1 → 15

If qubits = 4, it can handle:

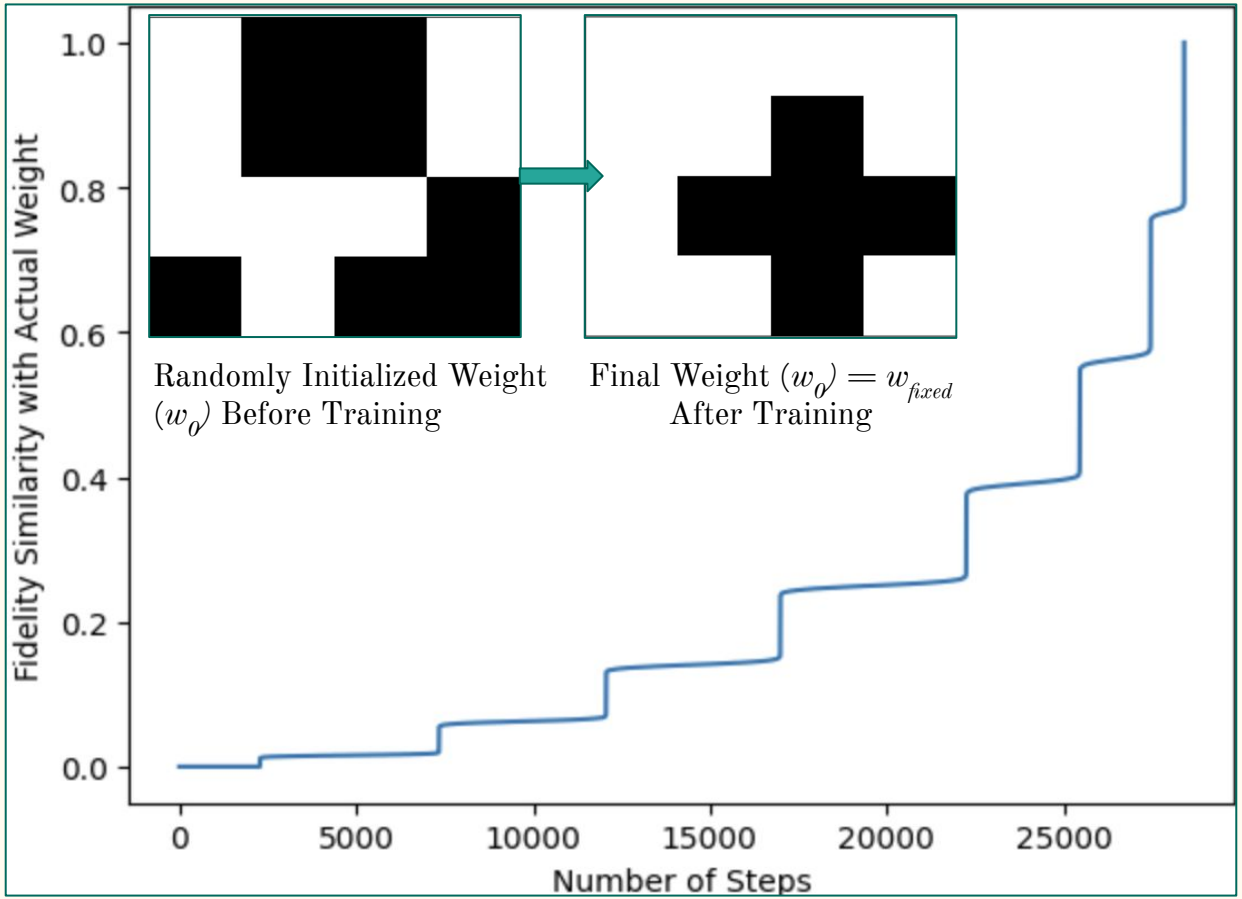
$2^4 = 16$

ex: 0 0 0 0 .. → 0

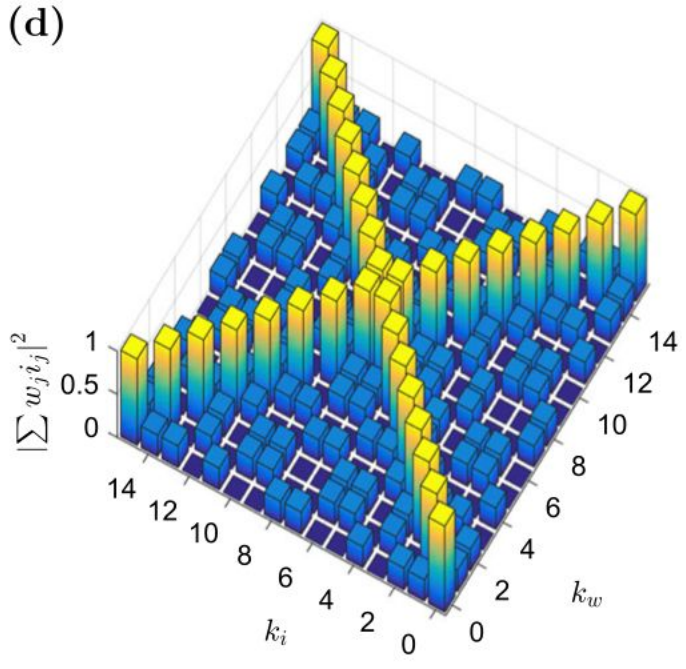
⋮

1 ... 1 1 → $2^{16} - 1$

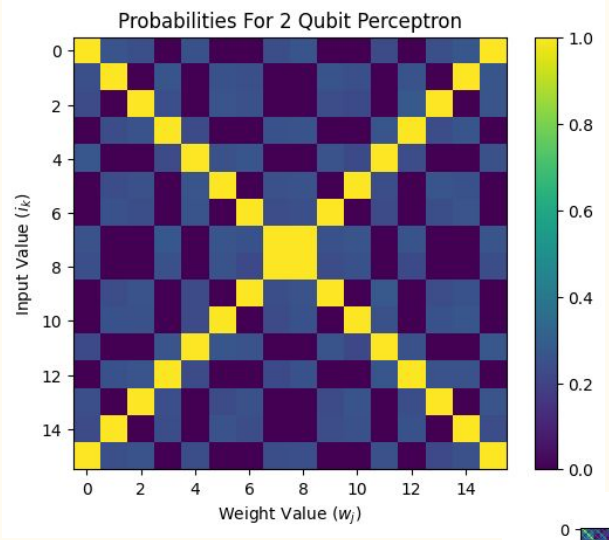
Result


Randomly Initialized Weight (w_0) Before Training
Final Weight (w_0) = w_{fixed} After Training
Fidelity Similarity with Actual Weight
1.0
0.8
0.6
0.4
0.2
0.0
0
5000
10000
15000
20000
25000
Number of Steps

Result

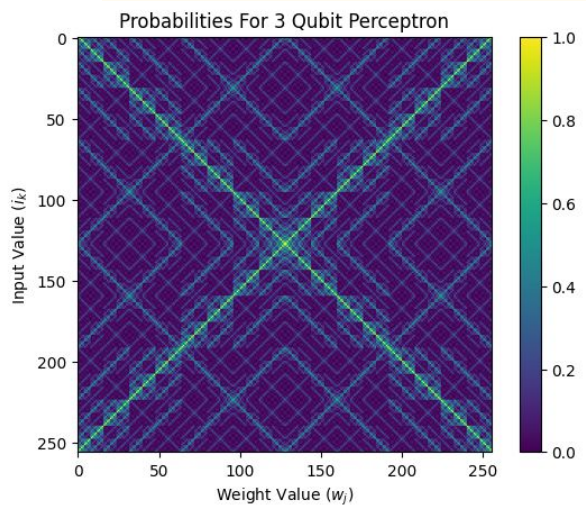


Probability Graph from paper



Probability Graph from our experiment

The probability is 1.0 only when i=w.

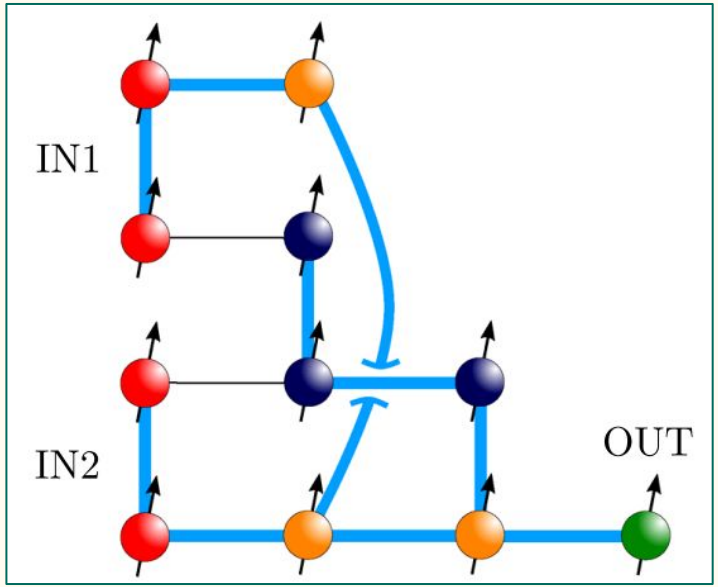


An Artificial Spiking Quantum Neuron

The Spiking Quantum Neuron has two objectives:

(1) Identifying if two Bell states are equal or not.
(2) Entangle the output state of the network with the input state.

The application of such network can be in quantum state preparation and communication.



Limitations

1. Generic quantum states or unitary transformations require an exponentially large number of elementary gates to be implemented.

2. A simple dataset with restricted binary inputs and weight vectors (the so called "McCollough-Pitts" neuron model.

Future Work

1. Make a feedforward deep net using multiple perceptrons.

2. Include more complex datasets.

References

- Tacchino, F., Macchiavello, C., Gerace, D., & Bajoni, D. (2018). An artificial neuron implemented on an actual quantum processor. *Npj Quantum Information*, *5*(1). **https://doi.org/10.1038/s41534-019-0140-4**

- Kristensen, L. S., Degroote, M., Wittek, P., Aspuru-Guzik, A., & Zinner, N. T. (2021). An artificial spiking quantum neuron. *Npj Quantum Information*, *7*(1). **https://doi.org/10.1038/s41534-021-00381-7**

- **GitHub Repository: https://github.com/ashutosh1919/quantum-perceptron**