A PRELIMINARY STUDY ON SMALL SCALE QUANTUM NEURAL NETWORK SIMULATION USING NUMPY

[1]NISHITA AGRAWAL, [2]PRANAV RB, [3]RAJA DAS

[1]Postgraduate Student, Department of Mathematics, VIT Vellore, Vellore, Tamil Nadu
[2]Postgraduate Student, Department of Mathematics, VIT Vellore, Vellore, Tamil Nadu
[3]Associate Professor Sr., Department of Mathematics, VIT Vellore, Vellore, Tamil Nadu
Email: [1]nishitaagrawal28608@gmail.com, [2]rbpranav1326@gmail.com

Abstract - This paper presents a basic, small-scale implementation of a Quantum Neural Network (QNN) using only NumPy. The objective is to demonstrate quantum-inspired learning concepts in a pedagogical and resource-efficient manner. The model employs a two-qubit circuit, with RY rotation gates for data encoding and a CNOT gate to introduce entanglement. Classical input data are projected into quantum states, and measurement probabilities are mapped to binary class predictions. The model is trained on the Breast Cancer Wisconsin dataset using the parameter-shift rule and gradient descent optimization. Despite its simplicity and execution on limited classical hardware, the QNN achieves over 80% accuracy, confirming that meaningful learning is possible even with minimal architectures. This approach offers a foundational framework for teaching and exploring quantum-enhanced learning concepts.

Keywords - Quantum Neural Networks (QNN), NumPy, RY gate, CNOT gate, Quantum Machine Learning.

I. INTRODUCTION

Quantum computing introduces a fundamentally new way of representing and processing information using quantum mechanical principles such as superposition and entanglement. Parallelly, machine learning has revolutionized how we derive patterns from large and complex datasets. Combining both, Quantum Neural Networks (QNNs) aim to exploit quantum mechanical effects to enhance learning capability, model expressiveness, and computational efficiency.

In this study, we implement a minimal two-qubit QNN entirely using NumPy, without relying on any quantum simulators or frameworks such as Qiskit. The simplicity of this model makes it both computationally lightweight and pedagogically useful for understanding the core ideas of quantum-inspired learning. The network uses parameterized RY rotation gates for data encoding and a CNOT gate to introduce entanglement between the two qubits, followed by a measurement on the second qubit.

The resulting circuit structure—illustrated in Figure 1—encodes classical data into quantum amplitudes, applies entanglement to capture feature correlations, and measures the output qubit to produce a probabilistic prediction used for binary classification. Despite being implemented on standard classical hardware, this setup closely mirrors the logic of real quantum circuits and offers valuable intuition about how QNNs process and learn from data.

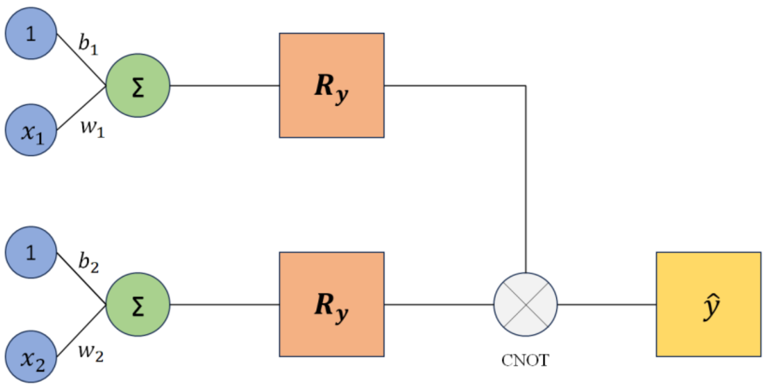

Figure 1: Two-Qubit Quantum Neural Network Circuit.

The circuit consists of two qubits initialized in the $|00\rangle$ state. Classical inputs are encoded using RY(θ_1) and RY(θ_2) rotations, entangled via a CNOT gate, and measured on the second qubit to obtain prediction probabilities.

II. DATA ENCODING

The Breast Cancer Wisconsin dataset was used to evaluate the model. Two principal components were extracted using PCA and normalized for use as rotation angles (θ_1 , θ_2) on the Bloch sphere. These angles are defined as:

$$\boldsymbol{\theta_1} = \mathbf{w_1} + \mathbf{b_1}, \quad \boldsymbol{\theta_2} = \mathbf{w_2} + \mathbf{b_2}$$

where w_1, w_2 are weights and b_1, b_2 are biases. These parameters are optimized during training.

III. QUANTUM CIRCUIT COMPOSITION

The model employs two qubits initialized in the $|00\rangle$ state. Each qubit undergoes an RY(θ) rotation, followed by a CNOT gate that introduces entanglement. Measurement of the Pauli-Z expectation value on the second qubit yields the model output.

Each qubit in the model is initialized in the $|0\rangle$ state. The classical inputs are encoded as rotation angles θ_1 and θ_2 , obtained from the input data and current trainable parameters:

$$\boldsymbol{\theta_1} = \mathbf{x_1 w_1} + \mathbf{b_1}, \quad \boldsymbol{\theta_2} = \mathbf{x_2 w_2} + \mathbf{b_2}$$

The quantum state of each qubit after applying the RY rotation is given by:

$$|\psi_1\rangle = \cos(\tfrac{\theta_1}{2})|0\rangle + \sin(\tfrac{\theta_1}{2})|1\rangle$$
$$|\psi_2\rangle = \cos(\tfrac{\theta_2}{2})|0\rangle + \sin(\tfrac{\theta_2}{2})|1\rangle$$

The combined two-qubit system is obtained through the tensor product:

$$|\psi_{in}\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$$

Next, a CNOT gate is applied, where the first qubit acts as the control and the second as the target. The CNOT operation flips the target qubit if and only if the control qubit is in the $|1\rangle$ state, creating entanglement between the two qubits. The resulting state vector encodes joint information from both features, enabling the circuit to model correlations between inputs.

IV. TRAINING AND OPTIMIZATION

After applying the CNOT operation, we measure the Pauli-Z expectation value of the second qubit:

$$\langle Z\rangle = P_0 - P_1$$

where:

$$P_0 = |\langle 00|\psi\rangle|^2 + |\langle 01|\psi\rangle|^2,$$
$$P_1 = |\langle 10|\psi\rangle|^2 + |\langle 11|\psi\rangle|^2$$

These represent the probabilities of observing the target qubit in the $|0\rangle$ and $|1\rangle$ states, respectively. The predicted output probability is then obtained as:

$$\hat{y} = \frac{1-\langle Z\rangle}{2}$$

The Mean Squared Error (MSE) is used as the loss function:

$$L = (\hat{y} - y)^2$$

To update the parameters, we compute the gradient of the loss with respect to each parameter using the parameter-shift rule:

$$\frac{\partial\langle Z\rangle}{\partial\theta} = \frac{1}{2}[\langle Z\rangle_{\theta+\frac{\pi}{2}} - \langle Z\rangle_{\theta-\frac{\pi}{2}}]$$

The full gradient for a weight (e.g.,w_1) is given by the chain rule:

$$\frac{\partial L}{\partial w_1} = 2(\hat{y} - y)\cdot(-\frac{1}{2})\cdot\frac{\partial\langle Z\rangle}{\partial\theta_1}\cdot x_1]$$

Similarly, for the biases:

$$\frac{\partial L}{\partial b_1} = 2(\hat{y} - y)\cdot(-\frac{1}{2})\cdot\frac{\partial\langle Z\rangle}{\partial\theta_1}]$$

Finally, the parameters are updated using gradient descent:

$$w_i^{(new)} = w_i^{(old)} - \eta\cdot\frac{\partial L}{\partial w_i}$$
$$b_i^{(new)} = b_i^{(old)} - \eta\cdot\frac{\partial L}{\partial b_i}$$

where η is the learning rate.

V. EXPERIMENTAL RESULTS AND DISCUSSION

.

Training was conducted for 100 epochs using a fixed learning rate. During this process, the model demonstrated steady convergence, with the loss decreasing smoothly and stabilizing after approximately 80 epochs. Despite using only a minimal two-qubit architecture, the QNN was able to learn meaningful relationships from both synthetic and real-world data. The final classification accuracy on the Breast Cancer dataset reached 81%, showing that even such a lightweight model can yield competitive results when trained efficiently.

To visualize the feature relationships before training, we performed Principal Component Analysis (PCA) on the Breast Cancer dataset and plotted the first two components. As shown in Figure 2, the data exhibits clear, though slightly overlapping, clusters corresponding to the two target classes. This provides an intuitive starting point for evaluating how well the QNN can separate these clusters in its learned decision space.

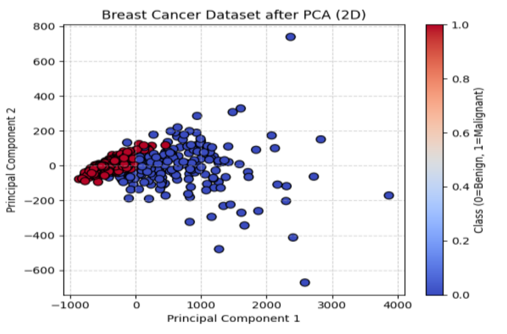

Figure PCA scatter plot of the Breast Cancer dataset highlighting class separation.

To better understand how the model learns from synthetic data, we visualized the probability of measuring the ∣ 1⟩ state across a range of input values. Figure 3 shows that the model learns to assign high probabilities to one region and low probabilities to another, effectively forming a nonlinear decision boundary. The periodic nature of the RY rotation leads to oscillatory patterns in the probability space, demonstrating how quantum operations naturally encode nonlinearity.

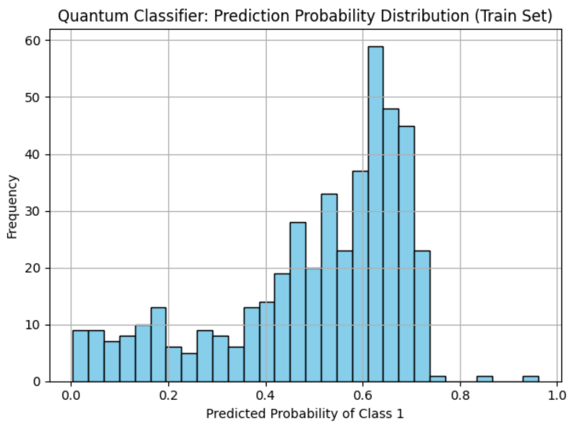

Figure 3: Training probability distribution for synthetic dataset inputs.

The overall training progression is shown in Figure 4. The binary cross-entropy loss decreases steadily over 100 epochs, confirming that the model effectively updates its parameters using the gradient descent rule. The smooth downward trend demonstrates that even with limited computational resources, stable convergence can be achieved in a purely NumPy-based simulation.

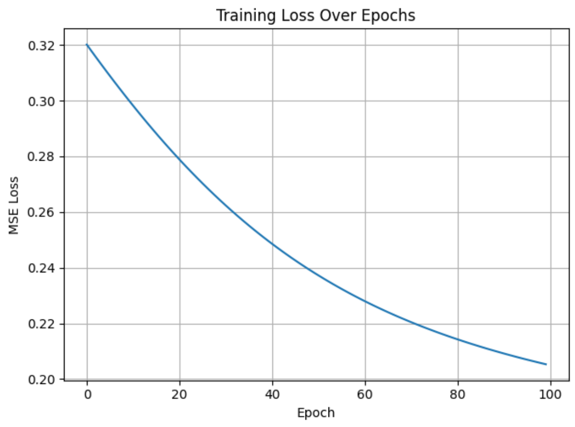

Figure 4: Binary cross-entropy loss curve across 100 epochs.

VI. CONCLUSION

This study demonstrates that a very basic, two-qubit QNN implemented in NumPy can perform meaningful binary classification. The simplicity and transparency of this implementation make it an ideal tool for educational and experimental purposes, especially where quantum hardware is not available. The model bridges classical and quantum concepts and serves as a baseline for scaling towards more complex hybrid quantum-classical architectures in future research.

REFERENCES

[1] M. Schuld, I. Sinayskiy, and F. Petruccione, "An Introduction to Quantum Machine Learning," Contemporary Physics, 2015.

[2] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2010.

[3] V. Dunjko and H. J. Briegel, “Machine Learning & Artificial Intelligence in the Quantum Domain,” Reports on Progress in Physics, 2018.

[4] B. M. Terhal and D. P. DiVincenzo, “Problem of Decoherence in Quantum Computers,” Phys. Rev. A, 1999.

★★★