Quantum Report 4

1. Objective

This report documents the fourth phase of experimentation with Quantum Recurrent Neural Networks (QRNNs). Building upon insights from earlier iterations, the primary goal was to refine the architecture and evaluate the impact of structural variations on model generalization. The experiments primarily focused on different QRNN simulation variants, each introducing subtle modifications to encoding, entanglement, and interaction strategies.

2. Experimental Setup

- **Dataset**: Amazon sentiment-labeled dataset (balanced positive/negative samples)

- **Input Representation:** Pre-trained 100-dimensional GloVe embeddings were compressed using an autoencoder into 4-dimensional latent vectors, enabling a direct 1:1 mapping to four qubits for angle encoding.

- **Quantum Configuration**:
  - 8-qubit circuits with designated input [0-3] and hidden wires [4-7].
  - Simple Angle encoding with parametrized rotations.
  - Variations across models in entanglement and interaction layers.

- **Training**:
  - Optimizer: Adam with parameter-shift differentiation.
  - Epochs: [20, 40, 50, 60] (On average an epoch of 20 was used)
  - Batch size: 1 (sentence-level training)

- **Evaluation**:
  - Test accuracy, loss, gradient and normalized gradient were tracked for each architecture.
  - Training logs included accuracy/loss plots for each run.

3. Results

The following table summarizes the testing performance across the nine model variants:

Model	Test Accuracy	Test Loss
QRNN_Sim_v1	0.5450	0.6962
QRNN_Sim_v2	0.5100	0.7015
QRNN_Sim_v3	0.5350	0.7121
QRNN_Sim_v4	0.5100	0.6991
QRNN_Sim_v5	0.5350	0.6886
QRNN_Sim_v6	0.6000	0.6837
QRNN_Sim_v7	0.5300	0.6907
QRNN_Sim_v8	0.5250	0.6906
QRNN_Sim_v9	0.5850	0.6869

Observations

- None of the models performed consistently well. While **QRNN_Sim_v6** logged a maximum accuracy of 60%, repeated evaluation runs showed large fluctuations, dropping as low as 54%. Increasing the number of shots (>1000) only degraded performance further.

- In conclusion, although some variants occasionally spiked to higher accuracies, none of the models demonstrated stable or reliable generalization.

4. Architecture

The models in this phase shared a common architectural design with variations in depth and entanglement. The core structure is as follows:

Model:

1. **Angle Embedding**: Each word vector (dimension 4) encoded on four input qubits using qml.RY.

2. **Input–Hidden Entanglement**: Input wires entangled with hidden wires using pairwise CNOT gates.

3. **QRNN Layers**: Each layer applied a sequence of parameterized qml.RY rotations, followed by CRY entanglement in ring topology, and another set of parameterized qml.RY gates.

4. **Interaction Layers**: Certain variants added additional entanglement layers (e.g., between subsets of hidden qubits) to improve mixing.

5. **Classifier Layer**: Final rotations (qml.Rot) applied on readout wires, producing two expectation values of PauliZ operators.

Layer Functions:

- **angle_embed_4**: Encodes the 4D latent word vector using RY rotations.

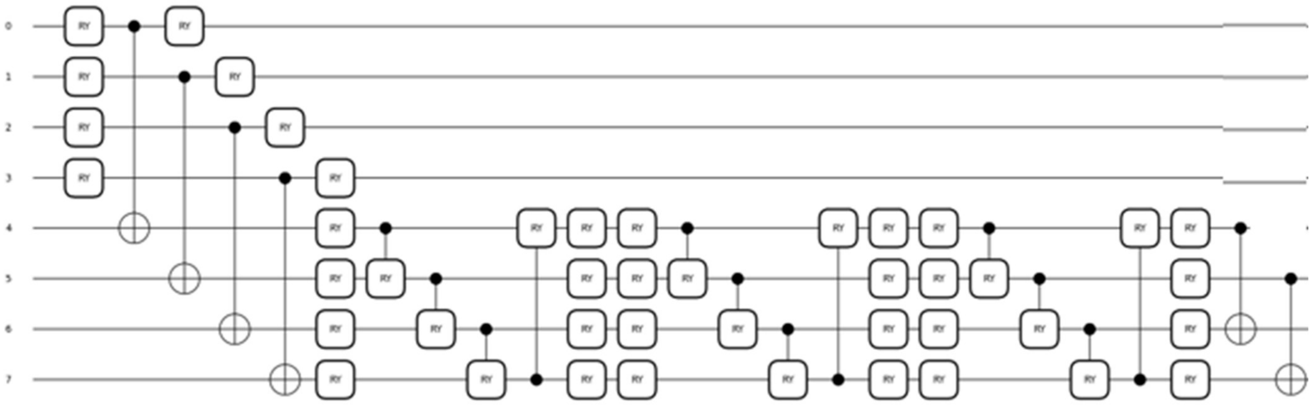
- **entangle_1to1**: Creates pairwise CNOT entanglement between input and hidden wires.

- **qrnn_layer_expressive**: Applies parameterized RY rotations → CRY entanglement (ring topology) → another round of RY rotations.

- **Classifier Block**: Uses qml.Rot (RZ-RY-RZ Euler Decomposition) gates on read wires for final transformation before measurement.

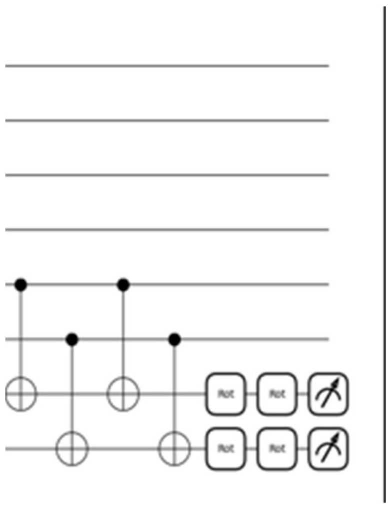
Quantum Circuit

Input + QRNN Cell



...... (9× QRNN Layer Repetition) ......

Final Measurement



5. **Architectural Motivation**

This architecture was motivated by the recurrent structure of classical RNNs, where each block processes both the current input and the previous hidden state. To simulate this, input qubits were entangled with hidden qubits at every timestep, allowing the hidden wires to carry combined information while the input wires were overwritten with new embeddings for the next word. Each QRNN layer consisted of parameterized rotations followed by entanglement and then another round of rotations.

We experimented with two main designs:

- **Simple Rotational Block:** A single **qml.Rot** (RZ-RY-RZ Euler Decomposition).
- **Expressive Block:** RX + RY + RZ rotations → entanglement (CNOT ring or parameterized CRY gates) → RX + RY + RZ rotations.

This full layer was repeated across two- and three-layer variants. At each timestep, the word embedding was processed through the entire pipeline before moving to the next word. After the stacked layers, an additional entanglement step combined hidden qubits **(1↔3, 2↔4)**, consolidating information into the final two hidden qubits used for measurement. Once all timesteps were processed, this consolidation was repeated, followed by two successive qml.Rot gates on the last two qubits for mixing before measurement.

5. Conclusion

This experimental phase highlighted the instability of the current QRNN simulation architectures. Despite occasional peaks (up to 60% accuracy), the models failed to maintain consistent performance across repeated evaluations, especially when tested with higher shot counts.

Name: Ashutosh Rai
Scholar No: 2422308
Guide: Dr. Partha Pakray