Quantum Circuits using Pennylane

QCNN and RQNN Working Mechanism

---

1. Introduction

This document presents the detailed architecture and functioning of the **Quantum Convolutional Neural Network (QCNN)** and the **Recurrent Quantum Neural Network (RQNN)** circuits, both designed using the **Pennylane** quantum framework. Each circuit operates with **8 qubits** (`n_qubits` = 8) and employs **3 variational layers**, integrating parameterized rotations and entanglement to model feature extraction and temporal evolution.

2. Circuit Architecture Overview

2.1 Entangling Layer (`entangle_layer`)

This component forms the backbone of both QCNN and RQNN circuits, enabling qubit interaction and correlation through controlled operations.

Component	Technical Detail	Working
Gate	**CNOT** (Controlled-NOT)	Flips the target qubit when the control qubit is $\|1\rangle$, generating **entanglement** that allows information exchange between qubits.
Connectivity	**Cyclic/Ring Topology**	Connects adjacent qubits ($[\mathtt{i}, \mathtt{i}+1]$) and includes a wrap-around CNOT ($[7, 0]$) to ensure complete cyclic entanglement.
Inputs/Outputs	Quantum states (not classical data).	Produces an entangled multi-qubit state as input to the next rotation layer.

2.2 QCNN Layer (`qcnn_layer`)

The QCNN acts as a **quantum feature extractor**, analogous to a convolutional layer in classical deep learning, encoding spatial correlations through quantum entanglement.

Phase	Operation	Gate Detail	Inputs & Outputs
Data Encoding	`RY(inputs[i], wires=i)`	$\mathbf{R_Y}(\theta) = e^{-i\frac{\theta}{2}Y}$. Rotates the qubit around the Y-axis by angle θ.	**Input:** 8 classical features ($0 \leq \theta \leq \pi$). **Output:** Initial encoded 8-qubit state $\lvert\psi_{\text{in}}\rangle$.
Variational Layer	`RX` rotation	$\mathbf{R_X}(\phi) = e^{-i\frac{\phi}{2}X}$. Applies a trainable rotation around the X-axis.	**Input:** Quantum state from previous step. **Parameter:** 8 learnable angles per layer.
	`entangle_layer`	**CNOT** entanglement.	**Input:** Rotated qubit states. **Output:** Entangled feature state.
Output	`expval(PauliZ(i))`	$\langle\sigma_Z\rangle = \langle\psi\lvert Z\rvert\psi\rangle$. Measures the expectation value along Z.	Produces a feature vector of 8 values (-1 to 1).

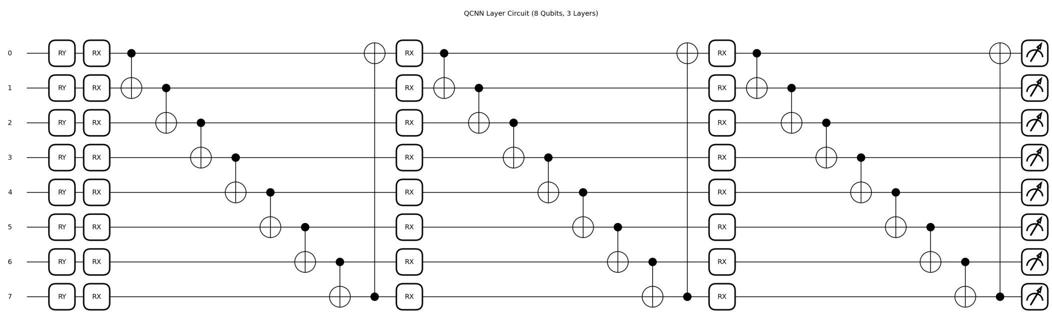

Figure 1: Quantum Convolutional Neural Network (QCNN) — showing RY feature encoding, RX parameter rotations, cyclic CNOT entanglement, and Z-basis measurement for feature extraction.

2.3 RQNN Cell (`rqnn_cell`)

The RQNN cell captures temporal dependencies in quantum states, allowing sequential modeling analogous to a recurrent neural network's hidden memory mechanism.

Phase	Operation	Gate Detail	Inputs & Outputs
State Encoding	$\mathtt{RY}(\mathtt{x}+\mathtt{h}, \mathtt{wires}=\mathtt{i})$	$\mathbf{R_Y}(\theta)$ rotations combine current input and memory.	**Input:** 8 input features (x) and 8 hidden states (h). **Output:** Combined encoded quantum state $\lvert\psi_{\text{combined}}\rangle$.
Variational Layer	`RZ` rotation	$\mathbf{R_Z}(\phi) = e^{-i\frac{\phi}{2}Z}$. Introduces phase rotation controlled by trainable parameters.	**Input:** State from encoding layer. **Parameter:** 8 trainable phase angles per layer.
	`entangle_layer`	**CNOT** entanglement.	**Input:** Post-rotation state. **Output:** Entangled hidden representation.
Output	`expval(PauliZ(i))`	$\langle\sigma_Z\rangle = \langle\psi\lvert Z\rvert\psi\rangle$.	Generates a new 8-dimensional hidden state for sequential time-step processing.

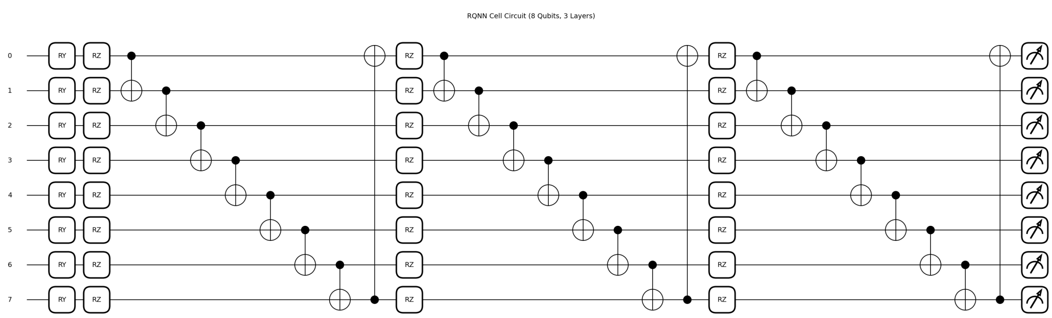

Figure 2: Recurrent Quantum Neural Network (RQNN) — combining RY encoding of input and hidden states, RZ-based variational processing, cyclic CNOT entanglement, and Z-basis measurement.

3. Working of the Quantum Circuits

3.1 QCNN Layer (Feature Extraction)

1. **Input Encoding:** Classical feature values are encoded as Y-rotations, generating an initial quantum state.

2. **Variational Processing:** The repeated RX and CNOT layers transform and entangle features, learning nonlinear spatial correlations.

3. **Feature Readout:** The expectation values $\langle Z \rangle$ are measured, forming the quantum feature vector.

3.2 RQNN Cell (Sequential Memory)

1. **State Integration:** The RY encoding merges new input and prior memory, allowing quantum memory retention.

2. **Temporal Transformation:** RZ and CNOT layers evolve this state across time, learning progression patterns.

3. **Hidden State Update:** Z-measurements produce the next hidden state, reused recursively across sequence steps.