Hybrid Quantum-Classical Learning on Iris Dataset: A Mathematical Perspective

Authors: SUNIL KANNEDI , MTech-UoH.

Under Prof. V Ravi, IDRBT and Dr. N. Naveen, UoH.

Presented By
SUNIL
24MCMI17
MTech-AI.

Agenda

- Introduction
- Hybrid Quantum Classical Deep Neural Network
- Quantum Encoding
- Application of Entanglement
- Application of Rotation Gates
- Application of Measurements
- Deep Neural Network
- Integration of Quantum with Classical Deep Neural Network
- Calculation of Total Loss
- Calculate Classical Gradient
- Calculation Quantum Gradient
- Updation of Both Classical and Quantum weights

Introduction: Iris Dataset Representation

The Iris dataset is a classic and widely used dataset in machine learning classification tasks. It categorized into three species: *Setosa*, *Versicolor*, and *Virginica*.

Data Point	Sepal Length	Sepal Width	Petal Length	Petal Width
1	5.1	3.5	1.4	0.2
2	4.9	3.0	4.7	0.1
3	4.7	3.2	1.3	2.5
4	4.6	3.1	5.5	1.2
5	5.0	3.6	1.4	1.5
• *(rows 6 to 149)* •				
150	5.9	3.0	5.1	1.8
Min-Max	(4.3, 7.9)	(2.0, 4.4)	(1.0, 6.9)	(0.1, 2.5)

Step 1: Quantum Encoding - Mathematical Explanation

Data Point 1: $x = (\text{SL} = 5.1, \text{SW} = 3.5, \text{PL} = 1.4, \text{PW} = 0.2)$

Sub-step 1.1: Normalization (Min-Max Scaling) Classical features are mapped to angles in $[0, \pi]$. Formula:

$$x_{\text{scaled}} = \frac{(x - x_{\min}) \cdot \pi}{x_{\max} - x_{\min}}$$

Feature-wise Scaling:

- Sepal Length: $\theta_{SL} = \frac{(5.1 - 4.3)\pi}{7.9 - 4.3} \approx 0.698\,\text{rad}$
- Sepal Width: $\theta_{SW} = \frac{(3.5 - 2.0)\pi}{4.4 - 2.0} \approx 1.963\,\text{rad}$
- Petal Length: $\theta_{PL} = \frac{(1.4 - 1.0)\pi}{6.9 - 1.0} \approx 0.214\,\text{rad}$
- Petal Width: $\theta_{PW} = \frac{(0.2 - 0.1)\pi}{2.5 - 0.1} \approx 0.131\,\text{rad}$

Result: Normalized feature angles for Data Point 1:

$$x'_{\text{norm}} = [\,0.698,\ 1.963,\ 0.214,\ 0.131\,]\ \text{radians}$$

Step 1.2: Quantum State Initialization (Angle Encoding)

We use the normalized angles to prepare the initial quantum state with 4 qubits q_0, q_1, q_2, q_3, each starting in state $|0\rangle$.

Single Qubit Initialization:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad R_y(\theta) = \begin{bmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix}$$

Example Encodings:

- Qubit q_0 (Sepal Length, $\theta_{SL} = 0.698$): $|\psi_0\rangle \approx \begin{bmatrix} 0.940 \\ 0.342 \end{bmatrix}$
- Qubit q_1 (Sepal Width, $\theta_{SW} = 1.963$): $|\psi_1\rangle \approx \begin{bmatrix} 0.556 \\ 0.831 \end{bmatrix}$
- Qubit q_2 (Petal Length, $\theta_{PL} = 0.214$): $|\psi_2\rangle \approx \begin{bmatrix} 0.994 \\ 0.107 \end{bmatrix}$
- Qubit q_3 (Petal Width, $\theta_{PW} = 0.131$): $|\psi_3\rangle \approx \begin{bmatrix} 0.998 \\ 0.065 \end{bmatrix}$

Tensor Product Examples

1. Two Column Vectors

$$\begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} a \begin{bmatrix} c \\ d \end{bmatrix} \\ b \begin{bmatrix} c \\ d \end{bmatrix} \end{bmatrix} = \begin{bmatrix} ac \\ ad \\ bc \\ bd \end{bmatrix}$$

2. Two 2×2 Matrices

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \otimes \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a \begin{bmatrix} e & f \\ g & h \end{bmatrix} & b \begin{bmatrix} e & f \\ g & h \end{bmatrix} \\ c \begin{bmatrix} e & f \\ g & h \end{bmatrix} & d \begin{bmatrix} e & f \\ g & h \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} ae & af & be & bf \\ ag & ah & bg & bh \\ ce & cf & de & df \\ cg & ch & dg & dh \end{bmatrix}$$

Step 1.3: Final Encoded Quantum State $|\psi_{\text{input}}\rangle$

Tensor Product Construction:

$$|\psi_{\text{input}}\rangle = |\psi_0\rangle \otimes |\psi_1\rangle \otimes |\psi_2\rangle \otimes |\psi_3\rangle$$

Resulting State Vector (dimension 16×1):

$$|\psi_{\text{input}}\rangle = \begin{bmatrix} 0.51846 & 0.03377 & 0.05581 & \cdots & 0.00198 \end{bmatrix}^T$$

Interpretation:

- Each entry corresponds to an amplitude for a basis state $|b_3 b_2 b_1 b_0\rangle$.
- Example: $0.51846 \to |0000\rangle$, $0.77504 \to |0100\rangle$.
- Measurement probabilities are $|\alpha_i|^2$ for amplitude α_i.

Understanding the CNOT Gate Mathematically

Definition: A Controlled-NOT (CNOT) gate acts on two qubits:

- Control qubit in $|0\rangle$: target unchanged.
- Control qubit in $|1\rangle$: target flipped (X applied).

Matrix Representation (2-qubit system):

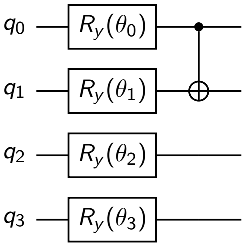
$$\mathrm{CNOT}_{01} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathrm{CNOT}_{10} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Embedding into 4-qubit system (16×16):

$$U_{\mathrm{CNOT}_{01}} = I \otimes I \otimes \mathrm{CNOT}_{01}$$

Application:

$$|\psi_{\text{after}}\rangle = U_{\mathrm{CNOT}_{01}} |\psi_{\text{input}}\rangle$$



CNOT on non-adjacent qubits: control q_c, target q_t

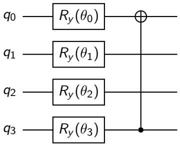
We use the computational basis ordered as $|q_3 q_2 q_1 q_0\rangle$ (16-dimensional Hilbert space).

1. Projector expansion: c=3 and t=0

$$U_{\mathrm{CNOT}(q_3,q_0)} = \big(|0\rangle\langle 0|_{q_3} \otimes I \otimes I \otimes I\big) + \big(|1\rangle\langle 1|_{q_3} \otimes I \otimes I \otimes X\big)$$

2. Block-diagonal representation:

$$U_{\mathrm{CNOT}(q_3,q_0)} = \begin{bmatrix} I_8 & 0 \\ 0 & (I \otimes I \otimes X) \end{bmatrix}_{16\times 16}$$



Quantum circuit: CNOT with control on q_3 and target on q_0

3. Basis-state swaps (examples):

$$|1000\rangle \leftrightarrow |1001\rangle, \quad |1010\rangle \leftrightarrow |1011\rangle, \quad |1110\rangle \leftrightarrow |1111\rangle$$

Application of Ring Entanglement in a 4-Qubit System

Definition: A ring entanglement layer applies CNOT gates sequentially so that each qubit controls the next, forming a closed loop.

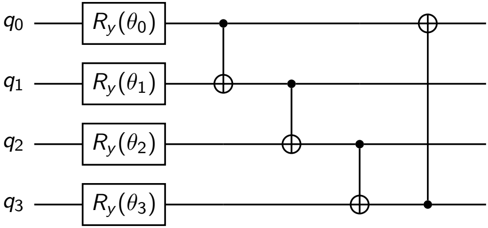
$$\text{Ring} = \{\text{CNOT}_{01},\ \text{CNOT}_{12},\ \text{CNOT}_{23},\ \text{CNOT}_{30}\}$$

Unitary:

$$U_{\text{ring}} = U_{30}\, U_{23}\, U_{12}\, U_{01}, \quad U_{\text{ring}} \in \mathbb{C}^{16\times 16}$$

Action on state:

$$|\psi_{\text{out}}\rangle = U_{\text{ring}}\, |\psi_{\text{in}}\rangle$$



Quantum circuit: R_y layer + ring entanglement

Application of Ring Entanglement Layer: Action on State

Definition:

$$|\psi_{\text{out}}\rangle = U_{\text{ring}}\,|\psi_{\text{in}}\rangle, \qquad U_{\text{ring}} = U_{30}\,U_{23}\,U_{12}\,U_{01}$$

Input state vector (dimension 16×1):

$$|\psi_{\text{in}}\rangle = \begin{bmatrix} 0.51846 & 0.03377 & 0.05581 & \cdots & 0.00198 \end{bmatrix}^T$$

Output state vector (dimension 16×1):

$$|\psi_{\text{out}}\rangle = \begin{bmatrix} \psi'_0 & \psi'_1 & \psi'_2 & \cdots & \psi'_{15} \end{bmatrix}^T$$

$$|\psi_{\text{out}}\rangle = \begin{bmatrix} 0.51846 & 0.00363 & 0.08341 & \cdots & 0.05581 \end{bmatrix}^T$$

Note: Each amplitude ψ'_k is obtained from basis-state swaps induced by the four CNOTs in the ring entanglement pattern.

In PennyLane, `qml.Rot` is the most general single-qubit rotation, while `qml.Rx`, `qml.Ry`, `qml.Rz` are special cases.

General Rotation:

$$\texttt{qml.Rot}(\phi,\theta,\omega) \;\equiv\; R(\phi,\theta,\omega) \;=\; R_z(\omega)\,R_y(\theta)\,R_z(\phi)$$

$$R(\phi,\theta,\omega) = \begin{bmatrix} \cos\frac{\theta}{2}\,e^{-i(\phi+\omega)/2} & -\sin\frac{\theta}{2}\,e^{-i(\phi-\omega)/2} \\ \sin\frac{\theta}{2}\,e^{+i(\phi-\omega)/2} & \cos\frac{\theta}{2}\,e^{+i(\phi+\omega)/2} \end{bmatrix}$$

Special Cases:

$$R_x(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}, \quad R_y(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix},$$

$$R_z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{+i\theta/2} \end{bmatrix}.$$

Interpretation:

Applying $R_y(\theta)$ to qubit q_0 in a 4-qubit system

We use the basis ordering $|q_3 q_2 q_1 q_0\rangle$ so operators act left-to-right in tensor products.

Single-qubit rotation:

$$R_y(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}, \qquad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Full 4-qubit operator acting on q_0:

$$U_{R_y}^{(q_0)}(\theta) \; = \; I \otimes I \otimes I \otimes R_y(\theta) \qquad \in \mathbb{C}^{16\times 16}.$$

Stepwise view (block-diagonal):

$$I_8 = I \otimes I \otimes I, \qquad U_{R_y}^{(q_0)}(\theta) = I_8 \otimes R_y(\theta) = \begin{bmatrix} R_y(\theta) & 0 & \cdots & 0 \\ 0 & R_y(\theta) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_y(\theta) \end{bmatrix}_{16\times 16}$$

Applying an $R_y(\theta)$ Layer to All 4 Qubits

Full 4-qubit operator (layer on all qubits):

$$U^{\text{layer}}_{R_y}(\theta) \;=\; R_y(\theta) \,\otimes\, R_y(\theta) \,\otimes\, R_y(\theta) \,\otimes\, R_y(\theta) \qquad \in \mathbb{C}^{16\times 16}.$$

Structure: This is no longer block-diagonal with copies of R_y; instead, it is a full Kronecker product of four 2×2 matrices, giving a dense 16×16 operator.
For example:

$$R_y^{\otimes 2}(\theta) = \begin{bmatrix} \cos^2\frac{\theta}{2} & -\cos\frac{\theta}{2}\sin\frac{\theta}{2} & -\cos\frac{\theta}{2}\sin\frac{\theta}{2} & \sin^2\frac{\theta}{2} \\ \cos\frac{\theta}{2}\sin\frac{\theta}{2} & \cos^2\frac{\theta}{2} & -\sin^2\frac{\theta}{2} & -\cos\frac{\theta}{2}\sin\frac{\theta}{2} \\ \cos\frac{\theta}{2}\sin\frac{\theta}{2} & -\sin^2\frac{\theta}{2} & \cos^2\frac{\theta}{2} & -\cos\frac{\theta}{2}\sin\frac{\theta}{2} \\ \sin^2\frac{\theta}{2} & \cos\frac{\theta}{2}\sin\frac{\theta}{2} & \cos\frac{\theta}{2}\sin\frac{\theta}{2} & \cos^2\frac{\theta}{2} \end{bmatrix}.$$

Action on a state:

$$|\psi_{\text{out}}\rangle \;=\; U^{\text{layer}}_{R_y}(\theta)\,|\psi_{\text{in}}\rangle$$

Rot Gate Unitary on a 4-Qubit System

To apply it to a specific qubit (say q_0) in a 4-qubit system, we use the tensor product with identity matrices:

$$U_{\text{Rot},q_0} = I_{q_3} \otimes I_{q_2} \otimes I_{q_1} \otimes \text{Rot}_{q_0}$$

Layer of Rot gates on all qubits:

$$U_{\text{Rot Layer}} = \text{Rot}_{q_3} \otimes \text{Rot}_{q_2} \otimes \text{Rot}_{q_1} \otimes \text{Rot}_{q_0}$$

Remarks:

- Each qubit rotation is independent.
- Total parameters for a layer $= 3 \times 4 = 12$ (since each Rot uses 3 angles).
- Rot can be decomposed as $R_z(\phi)R_y(\theta)R_z(\lambda)$.

Action on a state:

$$|\psi_{\text{after}}\rangle \; = \; U_{RotLayer} \; |\psi_{\text{before}}\rangle,$$

Measurement of Qubit q_0 in a 4-Qubit System (Qiskit)

Let the final quantum state be:

$$|\psi_{\text{final}}\rangle = \left[c_{0000}, c_{0001}, c_{0010}, \ldots, c_{1110}, c_{1111}\right]^T, \quad \sum |c_{b_3 b_2 b_1 b_0}|^2 = 1$$

Step 1: Probability of measuring q_0

$$P(q_0 = 0) = |c_{0000}|^2 + |c_{0010}|^2 + |c_{0100}|^2 + \cdots + |c_{1110}|^2$$
$$P(q_0 = 1) = |c_{0001}|^2 + |c_{0011}|^2 + |c_{0101}|^2 + \cdots + |c_{1111}|^2$$

Step 2: Post-measurement (state collapse)

- If $q_0 = 0$:

$$|\psi_0'\rangle = \left[c_{0000}, 0, c_{0010}, \ldots, c_{1110}, 0\right]^T, \quad |\psi_{\text{post}}^{(0)}\rangle = \frac{|\psi_0'\rangle}{\sqrt{P(q_0 = 0)}}$$

- If $q_0 = 1$:

$$|\psi_1'\rangle = \left[0, c_{0001}, 0, \ldots, 0, c_{1111}\right]^T, \quad |\psi_{\text{post}}^{(1)}\rangle = \frac{|\psi_1'\rangle}{\sqrt{P(q_0 = 1)}}$$

Pauli-Z Expectation Value

Pauli-Z operator for qubit i:

$$Z_i = I \otimes \cdots \otimes Z \otimes \cdots \otimes I$$

Expectation value of Z_i:

$$\langle Z_i \rangle = \langle \psi | Z_i | \psi \rangle$$

- **Method 1: Using marginals (probabilities of $b_i = 0$ and $b_i = 1$)**

$$\langle Z_i \rangle = \sum_{\text{bitstrings } b_i = 0} p_{b_0 \dots b_{n-1}} - \sum_{\text{bitstrings } b_i = 1} p_{b_0 \dots b_{n-1}}$$

- **Method 2: Direct sum over all basis states**

$$\langle Z_i \rangle = \sum_{\text{all bitstrings}} (-1)^{b_i} \, p_{b_0 \dots b_{n-1}}$$

Note: $\langle Z_i \rangle \in [-1, 1]$; this is the classical feature fed into dense layers in hybrid QML models.

Pauli-Z Expectation Value with Example (4-Qubit System)

Method 1: Using marginals (for qubit 0)

$$\langle Z_0 \rangle = \sum_{b_0=0} p_{b_3 b_2 b_1 0} - \sum_{b_0=1} p_{b_3 b_2 b_1 1}$$

$$\approx 0.99578 - 0.00422 \approx 0.9916$$

Method 2: Direct sum over all basis states

$$\langle Z_0 \rangle = \sum_{\text{all bitstrings}} (-1)^{b_0} p_{b_3 b_2 b_1 b_0}$$

4-Qubit Probability Vector (partial)

Bitstring	Probability
0000	0.2687
0001	0.00114
•	
(rows 3 to 14)	
•	
1111	0.0000039

$$\approx (+1)\cdot(0.2687 + \cdots + 0.00092) + (-1)\cdot(0.00114 + \cdots + 0.0000039) \approx 0.9916$$

Note

Both methods give the same result. $\langle Z_0 \rangle \in [-1, 1]$ and can be used as a classical feature for dense layers in a hybrid QML model.

Pauli-Z Expectation Values: 4-Qubit System

Expectation Values for each qubit (Q0-Q3):

$$\langle Z_0 \rangle \approx 0.9916$$
$$\langle Z_1 \rangle \approx 0.8765$$
$$\langle Z_2 \rangle \approx 0.4321$$
$$\langle Z_3 \rangle \approx -0.1234$$

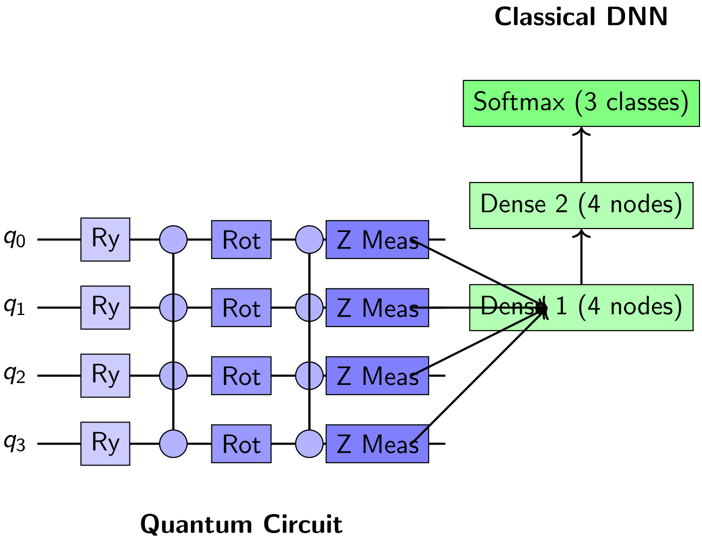
Expectation values $\langle Z_i \rangle \in [-1, 1]$ for $i = 0, 1, 2, 3$ are collected into a classica feature vector:

$$\mathrm{x} = [\langle Z_0 \rangle, \langle Z_1 \rangle, \langle Z_2 \rangle, \langle Z_3 \rangle]$$

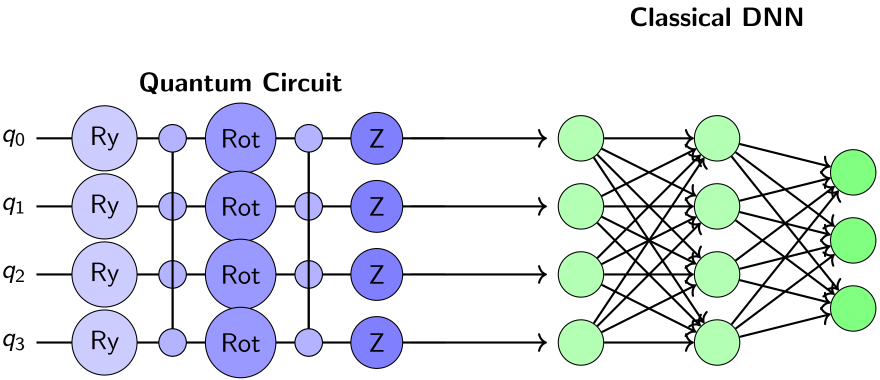
This vector is fed as input to the **classical Deep Neural Network** for pre- diction.

Note

Hybrid Quantum-Classical Neural Network



Hybrid Quantum-Classical Neural Network (4-Qubit System)



Gradients in Classical Deep Neural Networks

Backpropagation:

- Loss function: $\mathcal{L}(y, \hat{y})$
- Each layer l has weights $W^{(l)}$, biases $b^{(l)}$, and activation $\sigma(\cdot)$

$$a^{(l)} = \sigma\left(W^{(l)} a^{(l-1)} + b^{(l)}\right)$$

- Gradients computed using chain rule:

$$\frac{\partial \mathcal{L}}{\partial W^{(l)}} = \delta^{(l)} (a^{(l-1)})^T, \quad \delta^{(l)} = \frac{\partial \mathcal{L}}{\partial a^{(l)}} \odot \sigma'(z^{(l)})$$

- Update rule (SGD):

$$W^{(l)} \leftarrow W^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial W^{(l)}}$$

Gradients in Quantum Neural Networks (QNN)

Parameter-shift rule:

- Circuit outputs an expectation value:

$$f(\boldsymbol{\theta}) = \langle 0 | U^\dagger(\boldsymbol{\theta})\, O\, U(\boldsymbol{\theta}) | 0 \rangle$$

- For a gate $U(\theta) = e^{-i\theta G/2}$ with $G^2 = I$:

$$\frac{\partial f}{\partial \theta} = \tfrac{1}{2}\left[f(\theta + \tfrac{\pi}{2}) - f(\theta - \tfrac{\pi}{2}) \right]$$

- Requires two additional circuit evaluations per parameter.
- Gradient vector:

$$\nabla_{\boldsymbol{\theta}} f = \left(\frac{\partial f}{\partial \theta_1}, \frac{\partial f}{\partial \theta_2}, \dots \right)$$

Hybrid Gradient Descent (Classical + Quantum)

- Loss $\mathcal{L}(\mathrm{x}, \mathrm{y}; \boldsymbol{\theta}, W)$ depends on:
  - Quantum circuit parameters $\boldsymbol{\theta}$
  - Classical DNN weights W
- Gradient update:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}, \quad W \leftarrow W - \eta \nabla_{W} \mathcal{L}$$

- **Key point:**
  - Classical gradients via backpropagation (efficient).
  - Quantum gradients via parameter-shift (extra circuit calls).
- Both updates happen in the same optimizer loop.