

# 1) Quantum computing basics (QML-এর জন্য A-Z)

## 1.1 Qubit কী

ক্লাসিক্যাল বিট: শুধু **0** বা **1**

কোয়ান্টাম **qubit**: একই সাথে **0** ও **1**-এর মিশ্র অবস্থা হতে পারে।

গাণিতিকভাবে qubit state:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

- $\alpha, \beta$  = complex number (গুরুতে complex নিয়ে ভয় নেই)
  - Probability:
    - 0 পাওয়ার সম্ভাবনা =  $|\alpha|^2$
    - 1 পাওয়ার সম্ভাবনা =  $|\beta|^2$
- এবং  $|\alpha|^2 + |\beta|^2 = 1$

## 1.2 Superposition (একসাথে **0** ও **1**)

Superposition মানে state হলো mixture, কিন্তু **measurement** করলে একটাই ফল আসে (0 বা 1)।

সবচেয়ে জনপ্রিয় gate: **Hadamard (H)**

এটা  $|0\rangle$  কে করে:

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

মানে 0/1 দুইটাই 50%-50%

QML-এ কেন দরকার?

কারণ superposition থেকে “feature processing” parallel মতো হয় (intuition)

---

## 1.3 Entanglement (QML-এ খুব গুরুত্বপূর্ণ)

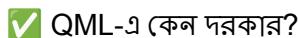
Entanglement হলো—দুই qubit এমনভাবে যুক্ত যে একটার ফল অন্যটার উপর নির্ভরশীল (classical correlation এর চেয়ে শক্তিশালী)।

উদাহরণ Bell state:

$$|\Phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

এখানে মাপলে:

- প্রথমটা 0 হলে দ্বিতীয়টাও 0
- প্রথমটা 1 হলে দ্বিতীয়টাও 1

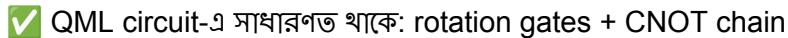


কারণ entanglement দিয়ে features-এর মধ্যে “interaction/relationship” ধরতে সুবিধা হয়।

---

## 1.4 Quantum gates (X, H, CNOT) — সহজ ব্যাখ্যা

- **X gate** = classical NOT  
 $|0\rangle \rightarrow |1\rangle, |1\rangle \rightarrow |0\rangle$
- **H gate** = superposition তৈরি করে
- **CNOT gate** = entanglement তৈরিতে সবচেয়ে বেশি ব্যবহৃত
  - control qubit = 1 হলে target flip হয়



## 1.5 Measurement (সবচেয়ে critical)

Measurement করলে state “collapse” করে এবং probability অনুযায়ী 0/1 আসে।

QML-এ আমরা সাধারণত measurement থেকে নিই:

- Probability of 1
- অথবা **Expectation value** (খুব কমন):  $\langle Z \rangle \in [-1, 1]$

এই ( $\langle \rangle$ ) পরে model output / prediction হিসেবে ব্যবহার হয়।

---

## 2) Data Encoding (সবচেয়ে গুরুত্বপূর্ণ) — A-Z

QML-এ সবচেয়ে বড় প্রশ্ন:

আপনার **classical data (x)** কীভাবে **quantum circuit**-এ চুকবে?

ধরেন আপনার input vector:  $x=(x_1, x_2, \dots, x_d)$

## 2.1 Angle Encoding (Beginner-friendly + popular)

এখানে feature গুলোকে rotation angle বানানো হয়।

একটা qubit-এ rotation gate:

- $R_x(\theta), R_y(\theta), R_z(\theta)$

উদাহরণ:

$$|0\rangle \xrightarrow{R_y(x_1)} \text{state changes}$$

যদি 2 feature হয় ( $x_1, x_2$ ), 2 qubit-এ:

- qubit0: ( $R_y(x_1)$ )
- qubit1: ( $R_y(x_2)$ )

তারপর entanglement gate (CNOT) দিয়ে interaction.

✓ Pros:

- implement করা সহজ
- ছোট dataset-এ ভালো

✓ Cons:

- dimension বেশি হলে qubit বাড়ে
- scaling issue

---

## 2.2 Amplitude Encoding (Powerful কিন্তু কঠিন)

Amplitude encoding-এ পুরো vector কে state amplitude বানানো হয়:

$$|x\rangle = \sum_i x_i |i\rangle$$

## ✓ Pros:

- (d) feature encode করা যায় মাত্র ( $\log_2 d$ ) qubit দিয়ে
- ✓ Cons:
- state preparation খরচ বেশি (গবেষণায় আলোচনা হয়)
- বাস্তবে circuit গভীর হয়ে যায়

Beginner research-এর জন্য মনে রাখবেন:

→ **Angle encoding** দিয়ে শুরু করাই **best**

---

## 3) Variational Quantum Circuit (VQC) / Quantum Neural Network — A-Z

### 3.1 VQC আসলে কী?

VQC হলো একটা trainable quantum model।  
এখানে কিছু gate-এর parameter থাকে ( $\theta$ ), যেগুলো শেখানো হয়।

Model:  $f(x; \theta) = \text{measurement output after circuit}$

### 3.2 VQC-এর ৩টা অংশ

#### Part A: Encoding layer

Data টোকানো: ( $x$ )

#### Part B: Trainable layer (ansatz)

এটা parameterized rotation gate:

- $R_y(\theta_1), R_z(\theta_2), \dots$
- এবং entanglement (CNOT)

#### Part C: Measurement layer

Output হিসেবে:

- probability বা expectation value
-

### 3.3 Output থেকে prediction কীভাবে হয়?

Binary classification এ খুব common:

$$\hat{y} = \sigma(\langle Z \rangle)$$

যেখানে  $\sigma$  = sigmoid

অথবা:

- $\langle Z \rangle > 0$  হলে class 1
  - $\langle Z \rangle \leq 0$  হলে class 0
- 

### 3.4 Loss function (কী minimize করবেন)

Binary classification এ:

- Cross entropy loss

$$\mathcal{L} = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

Regression এ:

- MSE

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

### 3.5 Training হয় কীভাবে (Hybrid training)

Quantum circuit gradient বের করা হয় parameter-shift rule দিয়ে (মনে রাখার মতো):

$$\frac{\partial f}{\partial \theta} = \frac{f(\theta + \pi/2) - f(\theta - \pi/2)}{2}$$

তারপর classical optimizer (Adam/SGD) দিয়ে update:

$$\theta \leftarrow \theta - \eta \nabla_{\theta} \mathcal{L}$$

- ✓ QML বলার কারণ: "quantum forward pass", "classical update"