Quantum Machine Learning

Evaluate quantum learning proposals through data, encoding, training and output task descriptions.

Undergraduate 22 min lesson Dr. Charith Jayasekara

By the end, you'll be able to

- Compare angle and amplitude data access.
- Calculate a one-angle classifier probability.
- Separate generalisation, trainability and measurement uncertainty.

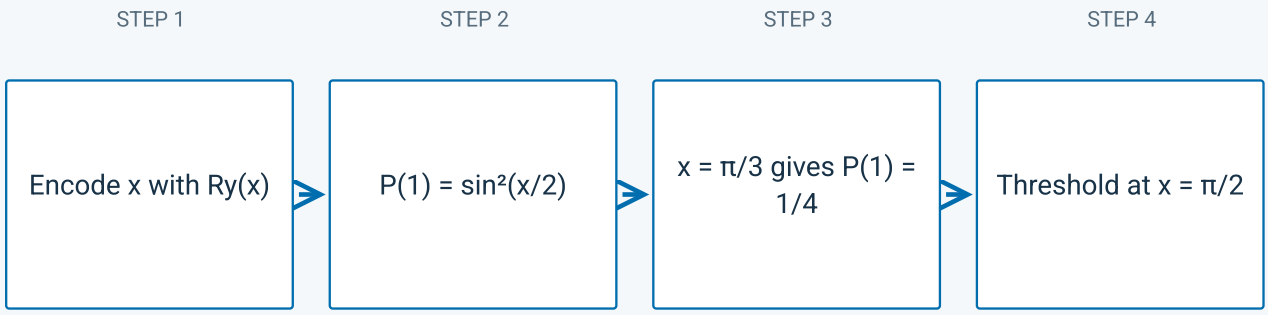
Identify what is quantum in the learning task

Quantum machine learning is a collection of methods combining quantum information processing with learning tasks. The data may be classical or quantum, the model may contain a quantum circuit, and the training loop may be classical. Those choices define different problems. A large Hilbert space alone does not guarantee better prediction, faster training or a useful generalisation advantage.

Prerequisites: measurement, parameterised gates and fair complexity claims. Quantum kernels and variational classifiers are concrete examples, each with its own assumptions. Havlíček and colleagues, Farhi and Neven.

WORKED DIAGRAM · A29

A one-angle learning model



This simple classifier is classically calculable. It illustrates the learning pipeline rather than a computational advantage.

The learning setup

Specify the input examples, labels, distribution and output predictor. State whether examples are ordinary arrays, efficiently prepared quantum states or unknown physical quantum specimens. Define the prediction loss,

train/validation/test split and total training budget. A claim about access to quantum data cannot be transferred automatically to a classical spreadsheet that must first be encoded.

For supervised classification, the aim is accurate predictions on unseen examples, not merely a low training error. For generative modelling, define the target distribution and evaluation statistic. For a quantum linear-algebra subroutine, state whether the result is a state, sample or classical vector. The application's output requirement can determine whether a proposed speedup is relevant.

Worked example: one-angle data encoding

Encode a scalar x between zero and pi by applying a Y rotation of angle x to zero. The resulting amplitudes are cosine and sine of half x. Measuring in Z gives one with probability sine squared of half x. At x zero, the output is zero; at x pi over two, it is evenly random; at x pi, it is one.

Suppose a classifier predicts the positive class whenever that probability exceeds one half. Its boundary is x equal to pi over two. A classical program can implement the same threshold immediately. The quantum circuit is a transparent model of encoding and readout, not evidence of an advantage. A finite-shot implementation adds uncertainty near the boundary, where the probability is close to one half.

Add a trainable Y rotation of angle theta after the encoding. Because the rotations share an axis, they combine into a rotation by x plus theta. The model shifts its decision boundary but does not suddenly become an arbitrary nonlinear classifier. Counting parameters and examining the actual function class is more informative than calling the circuit a neural network.

The actual one-qubit model is a simple trigonometric function of the encoded input.

$$R_y(x)|0\rangle = \cos(x/2)|0\rangle + \sin(x/2)|1\rangle, \qquad p_1(x) = \sin^2(x/2), \qquad p_1(x;\theta) = \sin^2((x+\theta)/2).$$

x is the encoded scalar angle and θ the trainable same-axis rotation added afterward. For $x \in [0, \pi]$ and $\theta = 0$, threshold probability one half occurs at $x = \pi/2$. The formula makes the model's limited function class visible.

Encoding is part of the algorithm

Angle encoding places features in rotation parameters and may require multiple gates per example. Basis encoding maps binary strings to basis states. Amplitude encoding represents a vector through normalised amplitudes, potentially using few qubits but demanding a preparation procedure. The qubit count alone hides that loading work and any loss of the original vector norm.

If data arrives natively as quantum states, different opportunities arise because measuring everything into a classical dataset can itself be costly. Even then, state-preparation assumptions, the number of copies and permissible measurements must be specified. "Quantum data" is an access model, not a synonym for data related to physics.

A complete training workflow

Choose a task and a classical baseline. Fix preprocessing using only training data. Select the encoding, circuit architecture, measurements and loss. Train parameters under a recorded shot and optimisation budget. Use validation data for model selection, then evaluate the selected model on an untouched test set. Report uncertainty from both finite samples and repeated training initialisations.

Count the number of circuit evaluations per example, per gradient and per optimiser update. Parameter-shift gradients may require multiple circuit settings per parameter. Kernel estimation can require many pairwise overlaps. Error mitigation adds further samples. A short inference circuit does not imply a cheap training procedure.

Generalisation and trainability are separate

A circuit family may represent many functions but be difficult to train. A model can fit training labels while generalising poorly. A quantum feature map may be hard to simulate yet produce a kernel that concentrates and becomes uninformative for the chosen data. These are different failure modes and require different tests.

Barren-plateau research identifies settings where gradients become difficult to resolve as systems grow. It does not prove every variational model is untrainable. Architecture, cost locality, initialisation, data structure and noise matter. McClean and colleagues. Treat a trainability claim as a statement about a specified family, not a blanket verdict.

Sampling uncertainty affects a decision boundary

At an input whose true one probability is one half, a classifier based on a finite measured majority can change labels across repeated runs. Increasing shots narrows the probability estimate's uncertainty, but it does not create a confident class distinction when the model itself lies exactly on the threshold.

At probability one quarter, one thousand shots have a count standard deviation of about 13.69 and a frequency standard deviation about 0.01369. That estimate is usually clearly below a one-half threshold. Near the boundary, the same shot budget may be insufficient for a desired confidence. A sensible inference policy can allocate more shots or return an uncertainty indication rather than pretending every measured label has equal reliability.

Distinguish this measurement uncertainty from generalisation error. Even an exactly evaluated model can be wrong on unseen data. More quantum shots improve evaluation of the chosen model, not the suitability of its learned decision rule.

What an advantage claim needs

Match the task and data access, compare with strong tuned classical methods, include loading/training/inference costs and evaluate on unseen examples. Demonstrating nonclassical states or fitting a small dataset is not sufficient. A useful result may be empirical and restricted, or theoretical under assumptions; state which one is supported.

The practical lesson is to ask a sequence of concrete questions: what data, what model, what output, what accuracy and what total resource? That framework lets a beginner evaluate a proposal without assuming that every quantum learning label represents the same algorithm.

Exercises and worked answers

Exercise 1. In the toy encoder, what is the one probability at x equal to pi over three?

> Check the worked answer
>
> It is sine squared of pi over six, equal to one quarter. A measurement-based prediction must estimate that probability from repeated preparations unless the model is evaluated analytically.

Exercise 2. Why is a model's perfect training accuracy insufficient evidence of quantum advantage?

Check the worked answer

It says neither how the model performs on unseen data nor whether a classical method achieves the same result more cheaply. Include test performance, baselines and all data-access and training costs under the same task definition.

Check your understanding

PAUSE & THINK

What is P(1) for the toy encoder at x = π/3?

- [ ] 1/4
- [ ] 1/2
- [ ] 3/4

Answer: 1/4. The half-angle is π/6 and $\sin^2(\pi/6) = 1/4$.

Sources

- Havlíček and colleagues
- Farhi and Neven
- McClean and colleagues