LECTURE 5.1 | By Amira Abbas | 16/07/2021

5

Building a Quantum Classifier

Hilbert space is a big space
— Carlton Caves

> With just 275 qubits, we can represent more states than the number of atoms in the observable universe.
>
> 2^{275}

→ Exponential State Space

→ Interference – Probability Amplitude
– Amplitudes can cancel each other out and we have this interference that really creates a different way to think about probability.

→ Quantum Mechanics provides Generalization of probability theory

→ In classical ML, probability theory is the backbone of interference and statistical modeling.

Quantum Machine Learning

Data generating system

Data Processing Device		
	CC	CQ
	QC	QQ

CC – classical data processed on a classical computer

QC – Quantum data processed on a classical computer

C-Classical Q-Quantum

Typo

DATE

{ In Quantum ML, we focus on

<u>CQ</u> — classical data processed on a Quantum computer }

Training Data X_{train} $\longrightarrow$ Model $f(X_{train}; \vec{\theta}) = \hat{y}$ $\longrightarrow$ COST $C(f(X_{train}; \vec{\theta}), y)$

$\downarrow$

Compute gradients

$$\frac{\partial C(f(X_{train}; \vec{\theta}), y)}{\partial \vec{\theta}}$$

$\longleftarrow$ Update $\vec{\theta}$ $\uparrow$ (back to Model)

What kind of model can be used in Quantum?

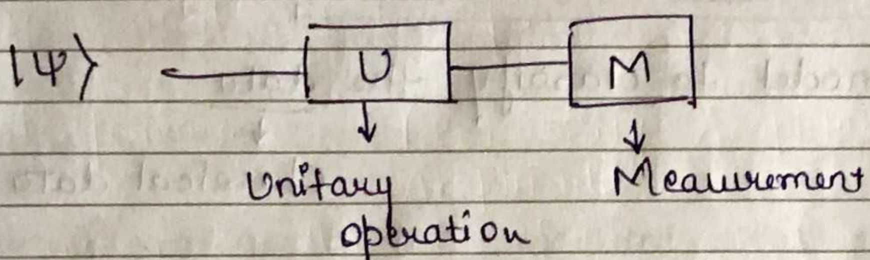
Some form of computation that runs on a Quantum computer.

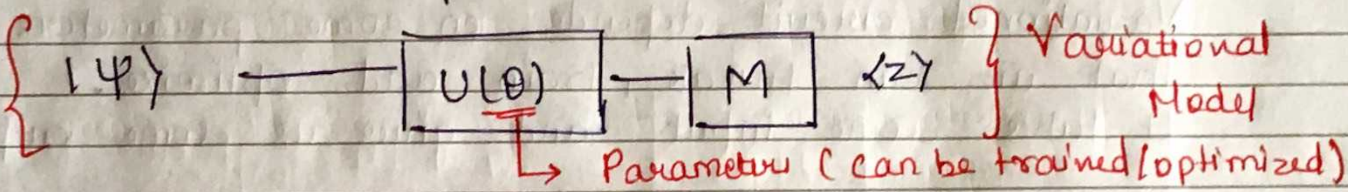
→ Near-term vs Fault-tolerant
↓
Noisy
Lot of errors

→ Noisy, error-prone, small devices

Qubits are no longer available.

→ **VARIATIONAL MODELS**





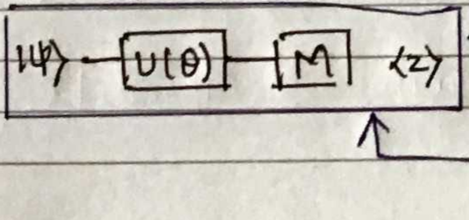
When we measure in quantum system, the output is STOCHASTIC in nature.

We repeat measurements multiple times to get an expectation value of the output.

So we get a probability distribution of possible basis states when we do measurements of a quantum circuit

If we change our parameter sets, we change the distribution of the statistics of our circuit.

Different names
Parametrized circuit / Variational Model / Ansatz



Ansatz :- A circuit that acts as a template and the circuit has parameters in it.

They are all referred to as Variational circuit

A First Attempt : Quantum ML

Varitional circuit as a Classifier

↓

As a model to classify the data

data ↓ Classical data

Variational circuit depends on some parameters that we can tweak, train and optimize.
It takes some input and gives us some output.

Data : x (classical) ⟶ Model : $f(x, \vec{\theta})$ linear model ⟶ Get a prediction/answer/label

$|\psi\rangle$ — $U(\theta)$ — M $\langle Z\rangle$

We can replace the linear model with a variational model.

DATE

$\boxed{\text{TASK}}$:- Train a quantum circuit on labelled samples in order to predict labels for new data

STEP 1 :- Encode the classical data into a quantum state

STEP 2 :- Apply a parameterized Model

STEP 3 :- Measure the circuit to extract labels

STEP 4 :- Use optimization techniques (like gradient descent) to update model parameters

DATA ENCODING :-

→ Basis encoding :- Encoding classical data on a Quantum computer
(No hard & fast rule)

Classical Data is encoded into Basis state or Computational state

Consider 2 qubits in a quantum circuit
↓
Basis states 00, 01, 10 and 11

$|00\rangle$ 0

$|01\rangle$ X_1

$|10\rangle$ 0

$|11\rangle$ X_2

We need to convert our data points x_1 and x_2 into some binary representation

Basis encoding says let's set up a circuit such that we encode our data x_1 and x_2 and associate them with appropriate basis states that they represent.

2 diff data points (In Binary form)

$x_1 \rightarrow 01$

$x_2 \rightarrow 11$

x_1 corresponds to 01

x_2 ——"—— 11

DATE

Amplitude encoding :-

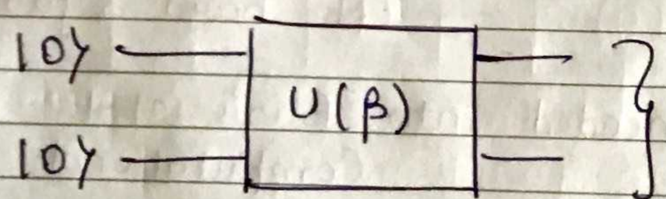
2 qubits $|0\rangle$ —

$|0\rangle$ —

Quantum state : $|0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ 4 enteries

Amplitude encoding says apply some rotations and these rotations must somehow encode your data.

We do some rotations that depends on some angles (β).



$|\psi\rangle = \begin{bmatrix} 0.8 \\ 0.1 \\ 0 \\ 0.1 \end{bmatrix}$ New vector

Data point (vector) $x = \begin{bmatrix} 0.8 \\ 0.1 \\ 0 \\ 0.1 \end{bmatrix}$

↳ they are not normalized (Not make sense)

We are encoding our data vectors into probability amplitudes and hence the name amplitude encoding

Lot of S/W packages have this templates that allow you to do amplitude encoding quite easily.

→ Angle encoding :- Consider we have data that has 2 features

Data is 2-D, classical data

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

We have 2 features so we have 2 qubits

$|0\rangle$ —

$|0\rangle$ —

Angle encoding says that with each qubit we will associate them with each feature value and encode each feature value by rotating the qubits about an angle that depends on the feature value.

It means that we apply an operation to the 1st qubit and this will be a rotation about some axis (z axis) and the angle that we rotate the 1st qubit will depend on the value of the first feature, so it will depend on the value x_1

$|0\rangle$ — $\boxed{R_z(x_1)}$ —

$|0\rangle$ — $\boxed{R_z(x_2)}$ —

We are encoding our classical data into angles of rotations for qubits.

If we increase the features in data point (ex x_3) then we just add 1 qubit to our system.

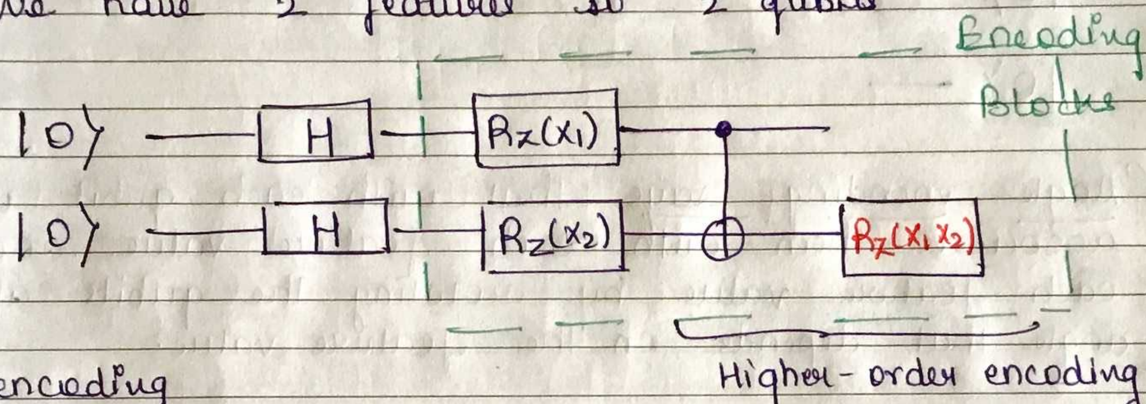
Higher-order Encoding :-

Similar to the idea of angle encoding, but with some more steps.

Consider 2-dimensional data vector, that we want to encode

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

We have 2 features so 2 qubits



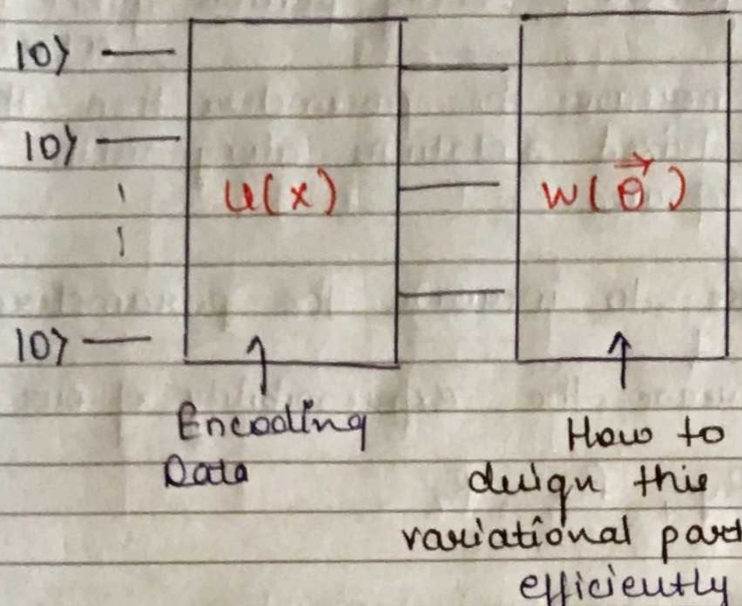
We are encoding higher orders of our data

Higher-order encoding
↓
→ Entanglement
→ Angle is a product of x_1 & x_2

People repeat this encoding strategy (Green) Encoding Blocks can be repeated. So they can be stacked upon each other. And this is referred to as the depth of your feature map.

When you increase the depth or the repetitions of your feature map then we are repeating this blocks, in the feature map structure.

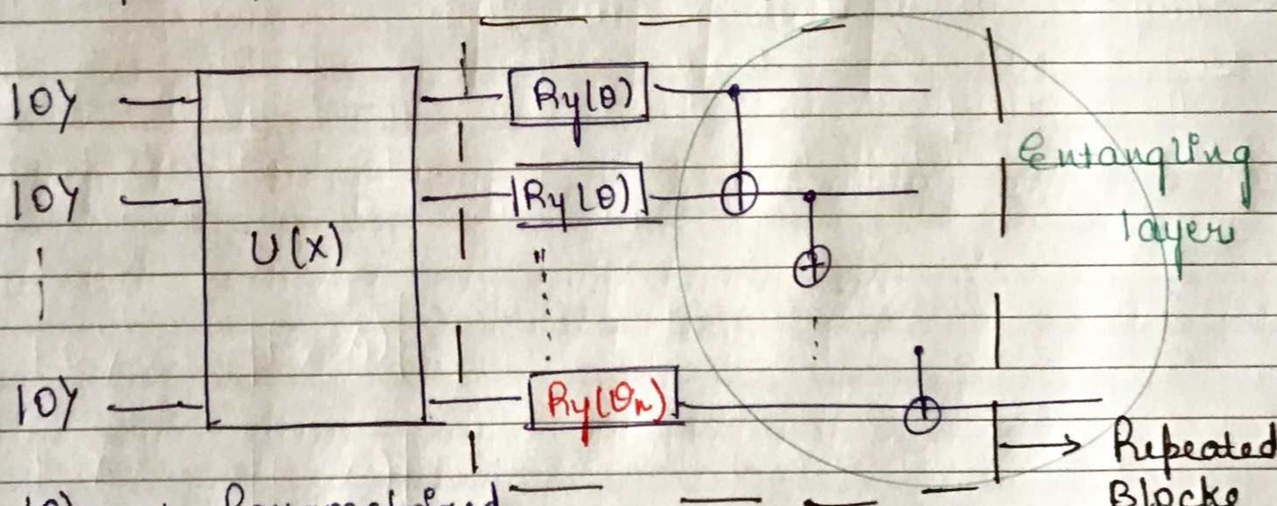
APPLYING A VARIATIONAL MODEL



Ref :- One of the research paper answers this question.

" Expressibility and Entangling capability of parametrized quantum circuits for hybrid quantum-classical algorithms ".

Ex :-



Ry(θ) → Parametrized

In the variational part, we have some rotations, that are applied to each qubit and these rotations are about a certain axis (eg y-axis) but this time these rotations are parameterized by trainable parameters (θ).

So the structure of these variational models are such that we do a set of rotations on every qubit that depends on a different angle.

DATE

- Pairwise Entanglement — Nearest Neighbours
- All-to-all Entanglement — Every possible qubit is entangled

If we want to increase the parameters then the block of parametrized rotations along with the entanglement is repeated.

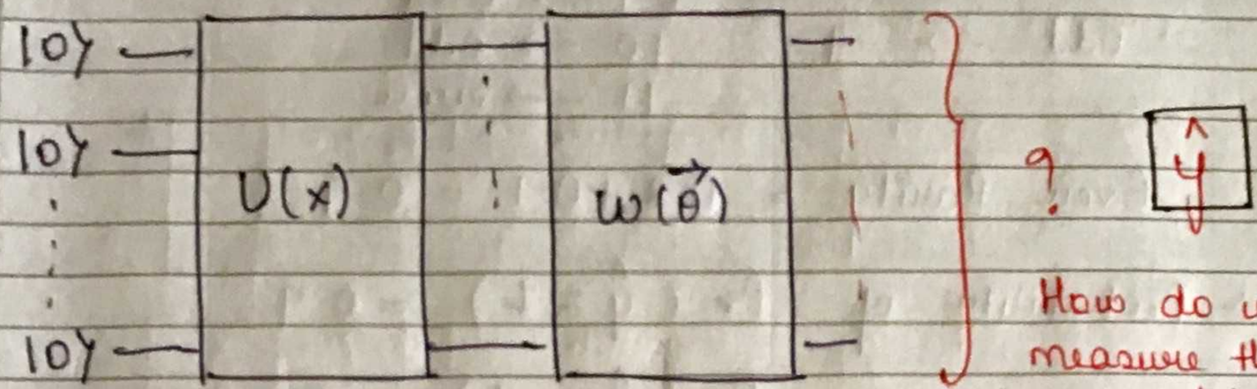
- Repeat the Blocks to increase the parameters

↓

which would increase the expressibility of our model.

Rotations : R_y, R_x, R_z

EXTRACTING LABELS

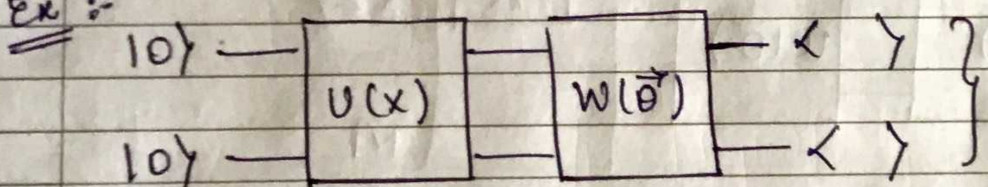


How do we measure this quantum model such that we can interpret as a label for our machine learning task?

Ref :- "Supervised learning with quantum-enchanced feature spaces."

→ Parity - Post processing

Ex :-



Basis states

00 → 0.8
01 → 0.1
10 → 0
11 → 0.1

Prob

⇒ Quantum measurements are stochastic in nature.

⇒ So we have to repeat the experiment, the measurements over and over, we do multiple shots.

These shots-based measurements gives us a probability distribution or an expectation for achieving or measuring each of the above basis states.

Each of the basis states will have a probability associated with it.

DATE

Parity

even → +1

odd → −1

00 → even
01 → odd
10 → odd
11 → even

Even Parity = 0.8 + 0.1 = 0.9

⇒ Probability of $Pr(\hat{y} = 1) = 0.9$

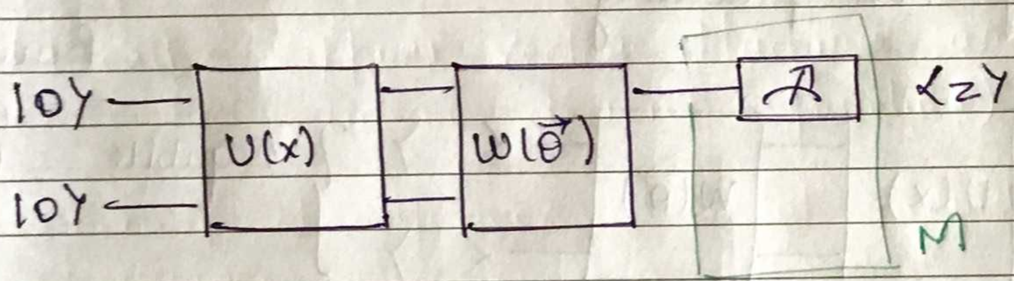
$Pr(\hat{y} = -1) = 0.1$

Measuring the first qubit :-

Consider Binary classification problem

↓

Two classes → class +1, class −1



We just measured the first qubit and ignore others.

⟨Z⟩ this would gives us a value that lies between −1 and +1.

We can threshold the value with some rule.

if $z < 0 \rightarrow -1$

$z \geq 0 \rightarrow +1$

M is the observable where we just measure the first qubit and ignore everything else.

DATE

Complicated Measurement strategy (M^c) can be decomposed in some rotations that happen before an easy measurement strategy (M).

M^c = rotations before M

We don't need complicated variational observable.

OPTIMIZATION :-

→ How could we compute gradients efficiently w.r.t to parameters ?

→ We have a model that depends on a quantum computation on a quantum circuit.

→ Can we compute the gradient of our model.

⇒ Can we compute the gradient of a quantum circuit

⇒ Yes we can.

Parameter Shift Rule (Some reference in the paper)

Gradient Descent

We can compute gradient on Quantum circuits in Qiskit.

Gradient =

$|0\rangle^{\otimes n}$ — $\boxed{U(\theta+s)}$ — $\boxed{\nearrow}$ = $\hat{y}_{\theta+s}$ −

$|0\rangle^{\otimes n}$ — $\boxed{U(\theta-s)}$ — $\boxed{\nearrow}$ = $\hat{y}_{\theta-s}$

Difference between the 2 circuits

We shift the parameters up (+s)

—— 11 —————— down (−s)

Typo

DATE

Advantages :-

Data Encoding :

$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ n qubits $\rightarrow |\phi_x\rangle = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$ $1 \ldots 2^n$

2 qubits $\rightarrow |\phi_x\rangle = \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \end{bmatrix}$

Transformation of encoding data from one space to a higher dimensional space.

They are linear classifier in a feature space (Recall Support vector Machines and their linear decision function)

Quantum classification does not provide any advantage.
We are doing linear classification of some feature map data.

RECAP :- Data Encoding → Basis
→ Amplitude
→ Angle
→ Higher order

Variational Model → Ry
→ Entanglement

Optimization → gradients of circuits

Quantum classifier as a first Attempt