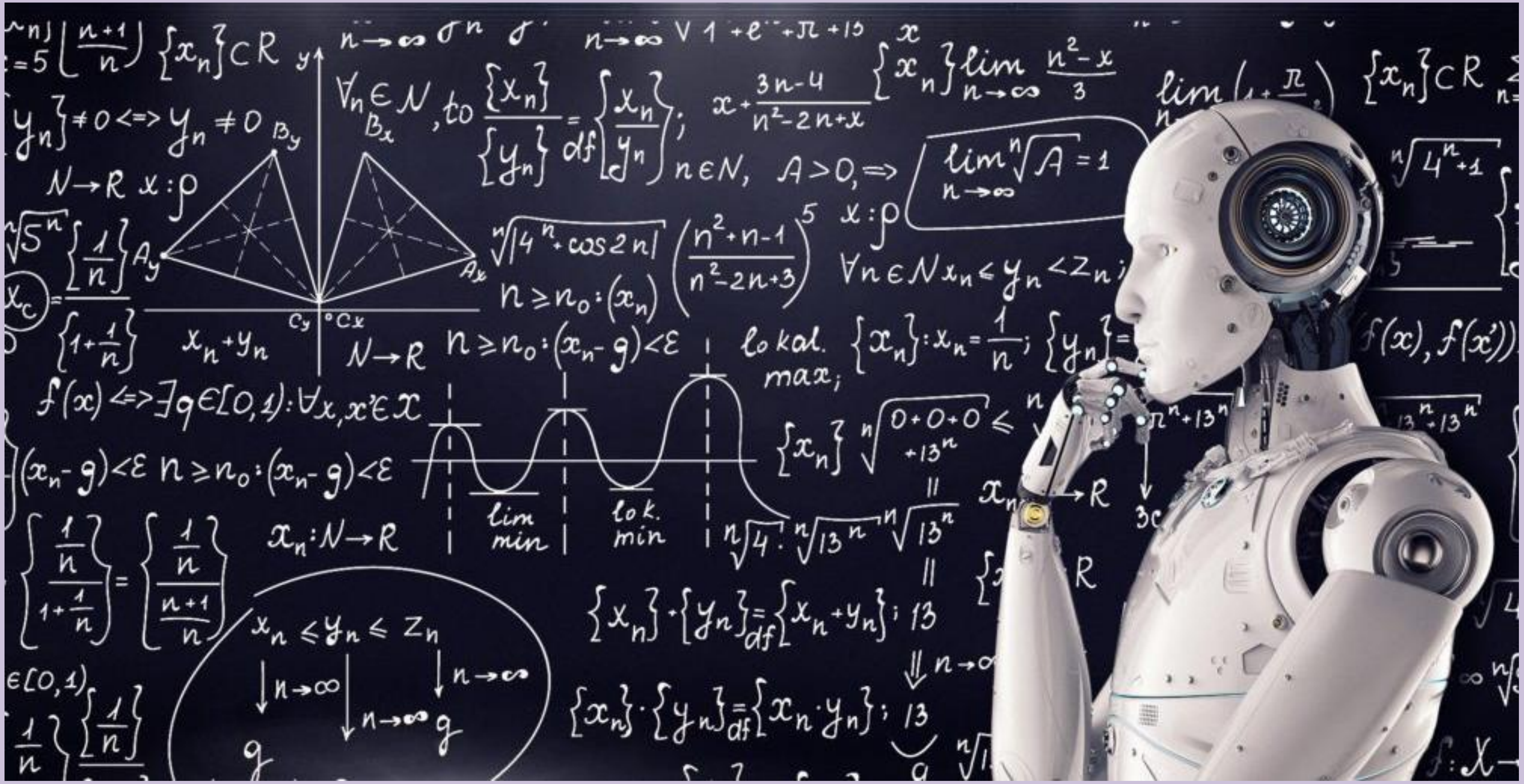
Quantum Machine Learning

Yuewen Hou, University of Michigan

Slides Courtesy: IBM Quantum, Mohammad Aamir Sohail

Machine learning preliminaries



Machine learning overview

Function approximation and optimization

$$g(x)$$

true function

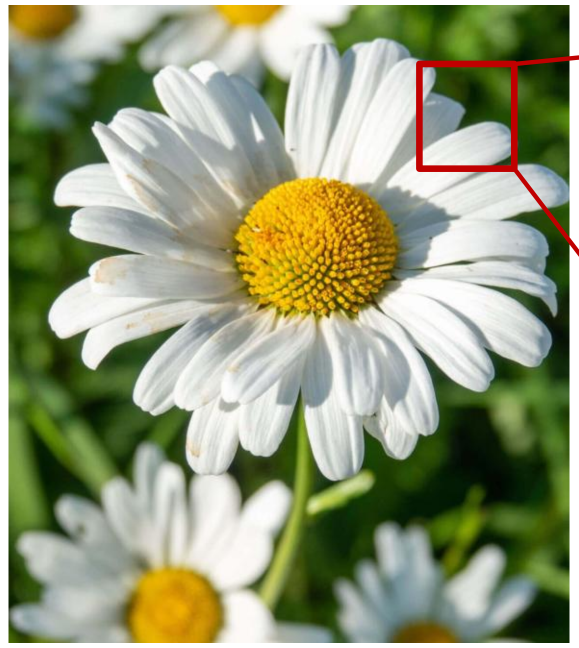
approximate $\longrightarrow$

$$f(\hat{x}, \overrightarrow{\theta})$$

mathematical model

x : data features

e.g. pixel values of an image



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

e.g. $h(x, \overrightarrow{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$

Goal: choose f

train $\overrightarrow{\theta}$

Machine learning types

1

Supervised Learning

- Classification
- Regression

2

Unsupervised Learning

- Dimensionality reduction
- Clustering
- Some generative models like GAN, autoencoder, etc.

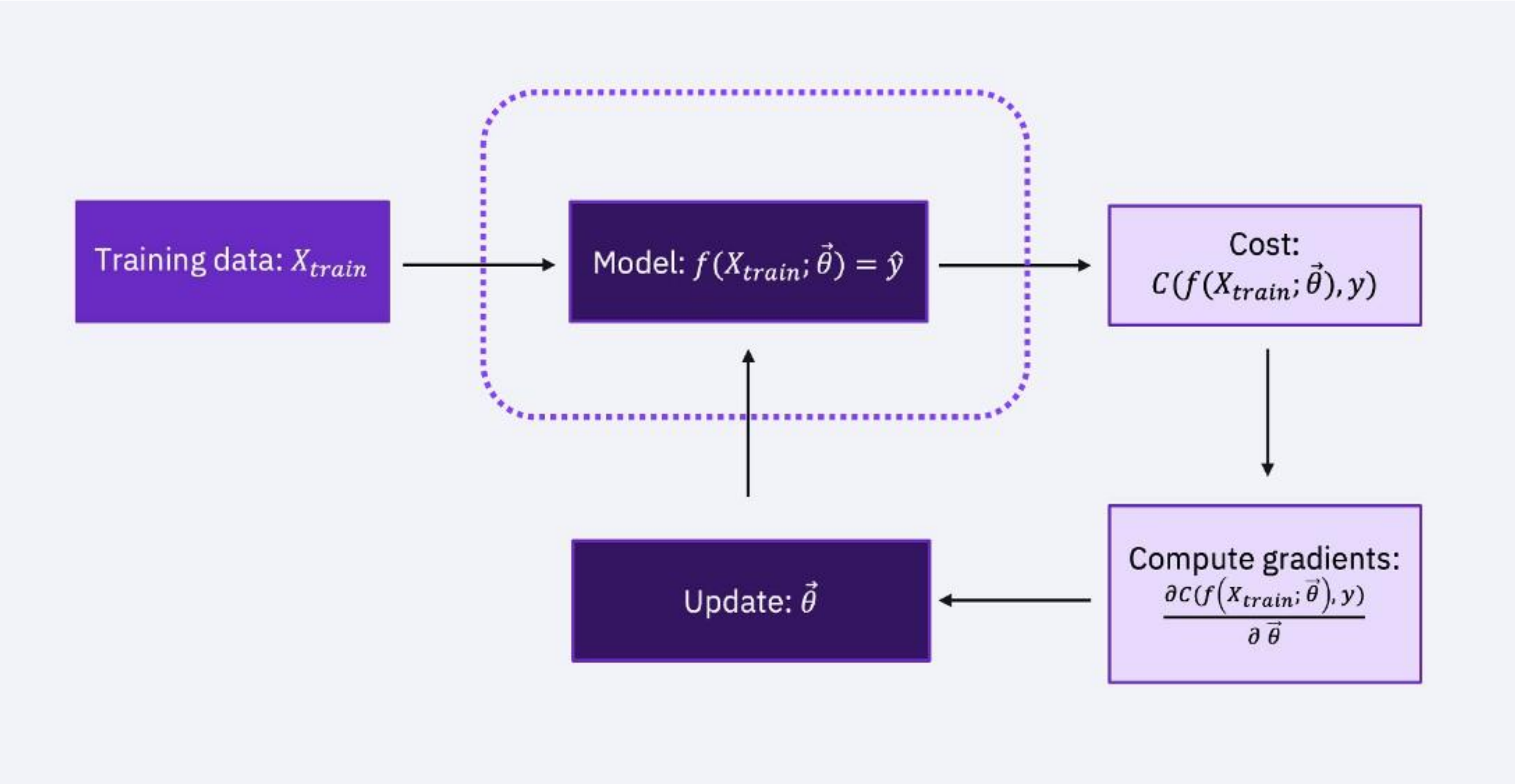
3

Reinforcement Learning

Agent maximizing rewards in an environment

Source: https://www.ibm.com/blog/machine-learning-types/

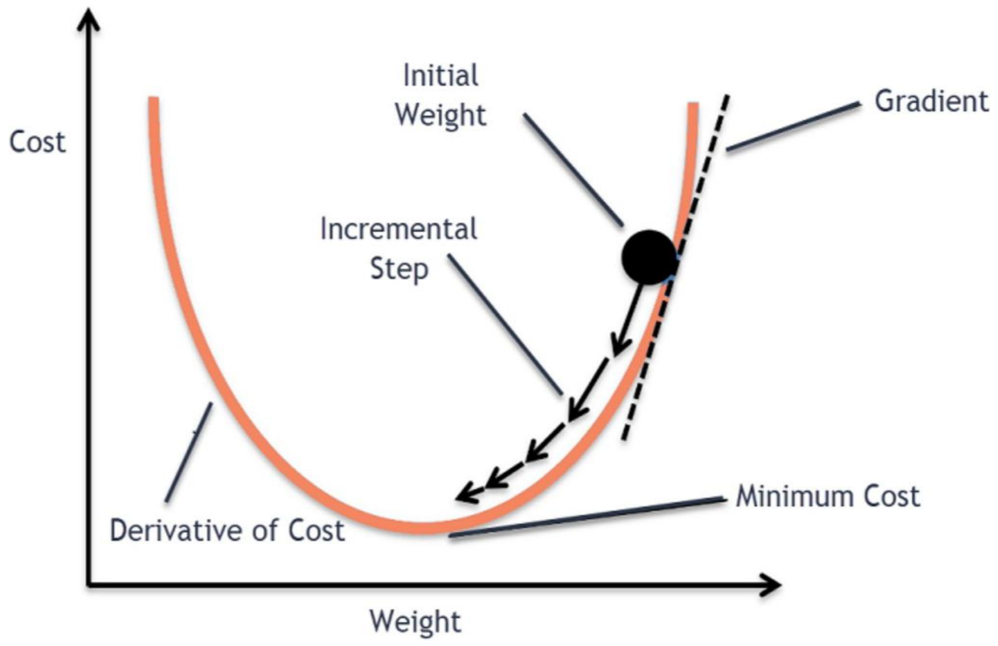
Supervised learning workflow



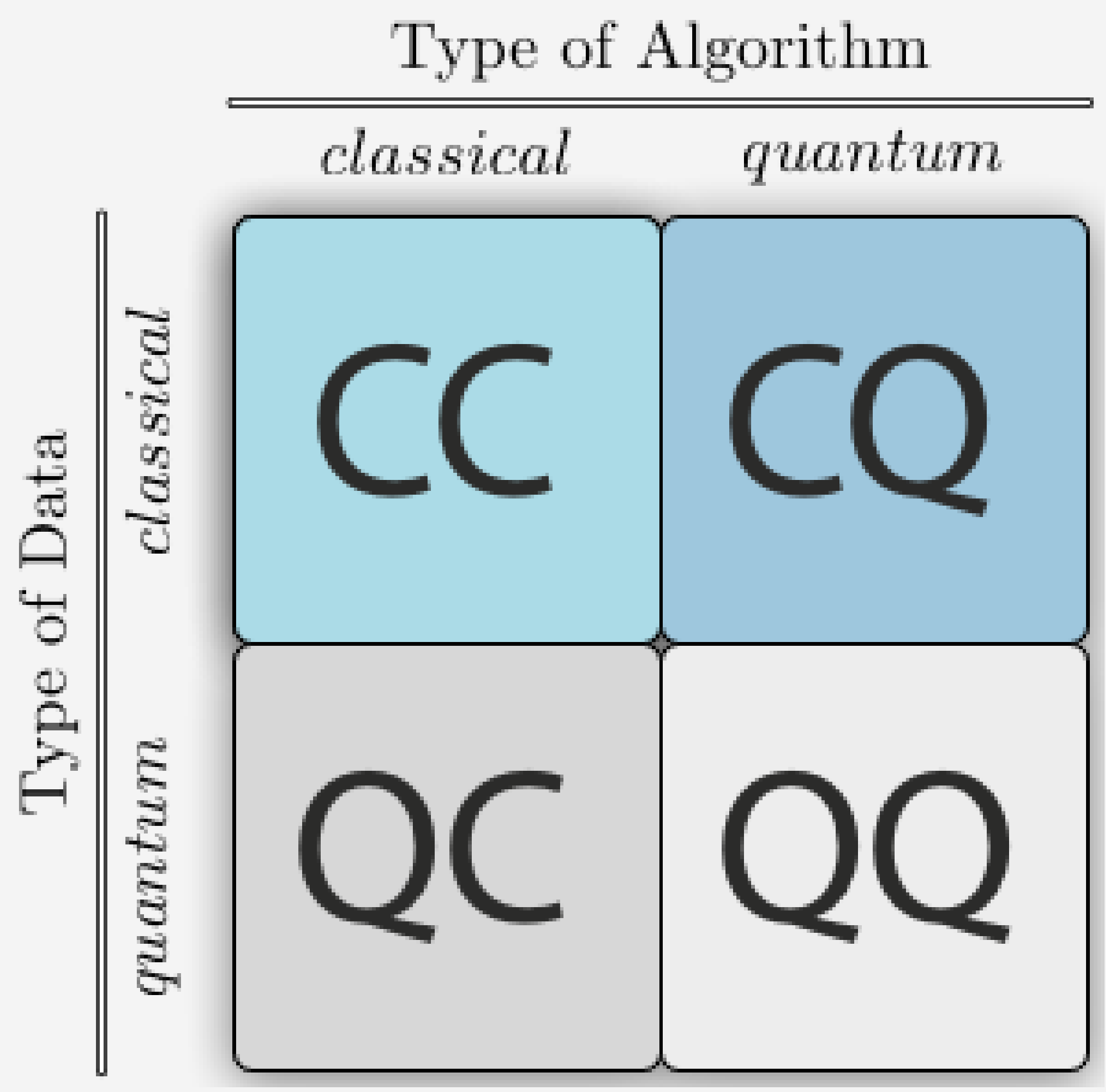
e.g. Mean squared error

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2.$$

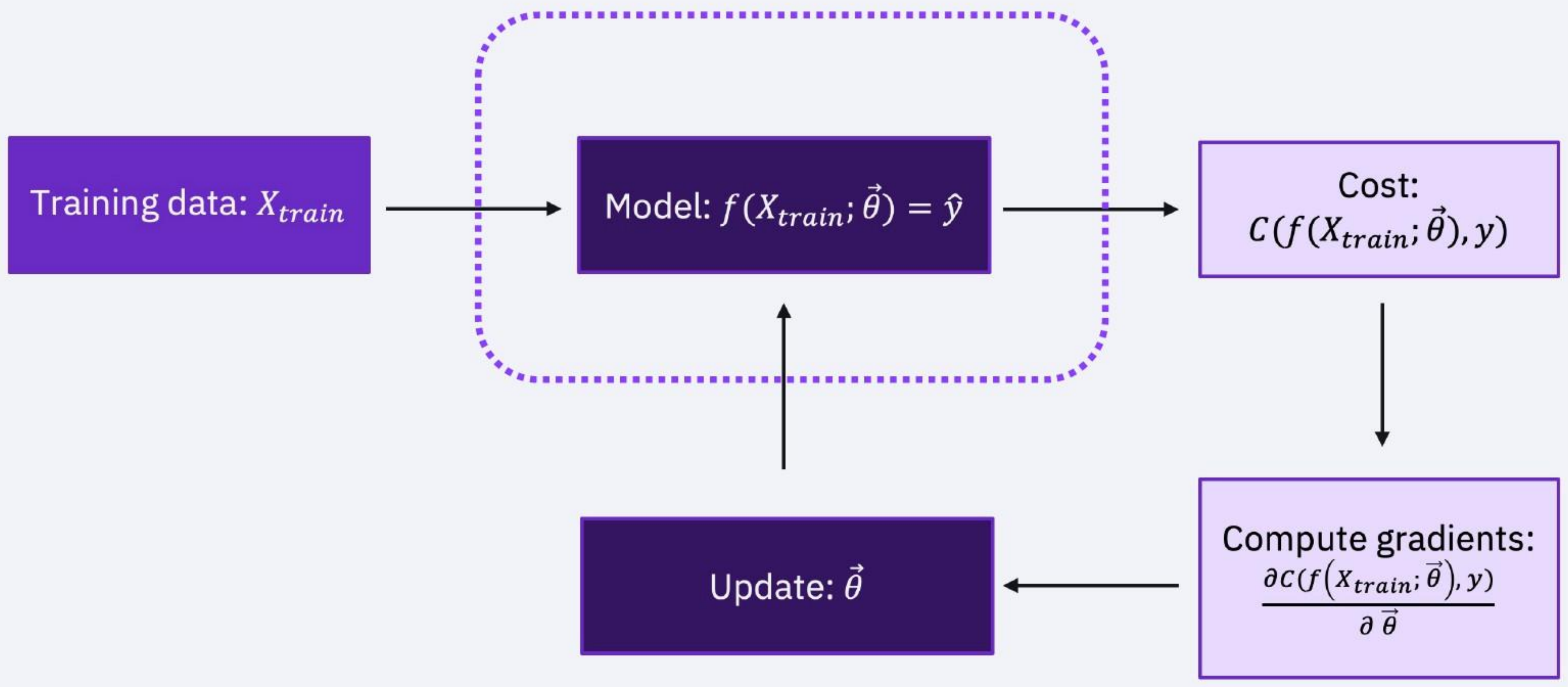
e.g. Gradient descent



Quantum machine learning



Schuld, Maria, and Francesco Petruccione. Supervised learning with quantum computers. Vol. 17. Berlin: Springer, 2018.



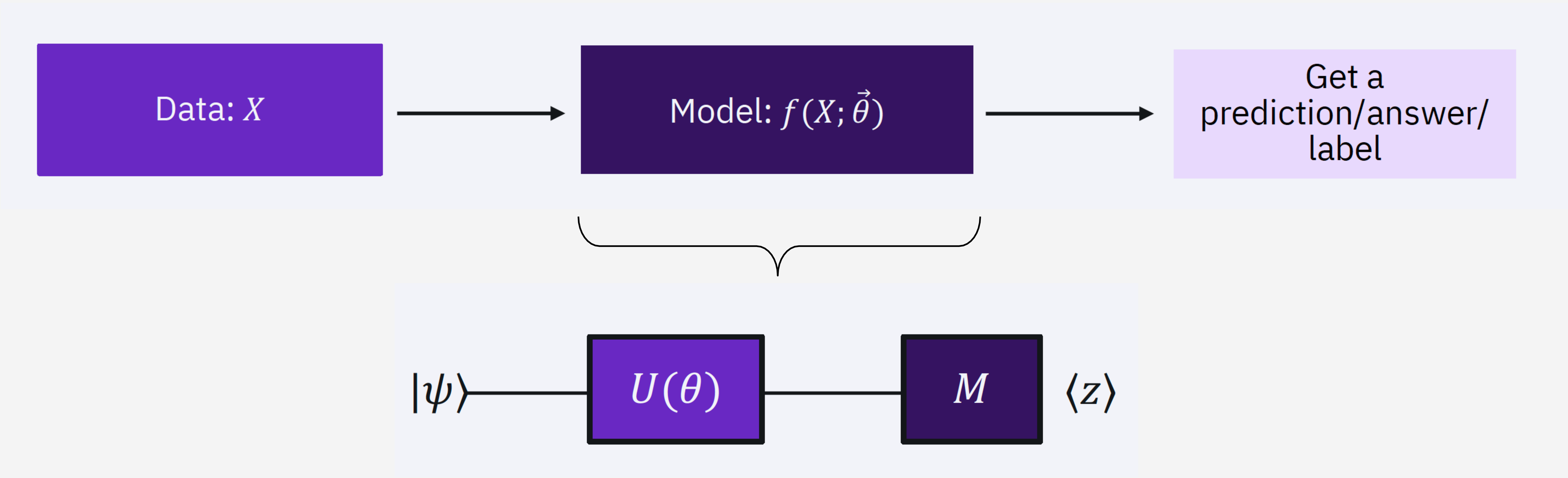
Also: **near-term** vs fault-tolerant

- Quantum SVM
- Quantum NNs

- HHL algorithm
- Quantum PCA

Harrow, Aram W., Avinatan Hassidim, and Seth Lloyd. "Quantum algorithm for linear systems of equations." Physical review letters 103.15 (2009): 150502.
Lloyd, Seth, Masoud Mohseni, and Patrick Rebentrost. "Quantum principal component analysis." Nature Physics 10.9 (2014): 631-633.

Variational circuit as a classifier



Variational circuit

Parametrized quantum circuit (PQC)

Ansatz

Variational circuit as a classifier

Task: Supervised learning (suppose binary classification, {1, -1})

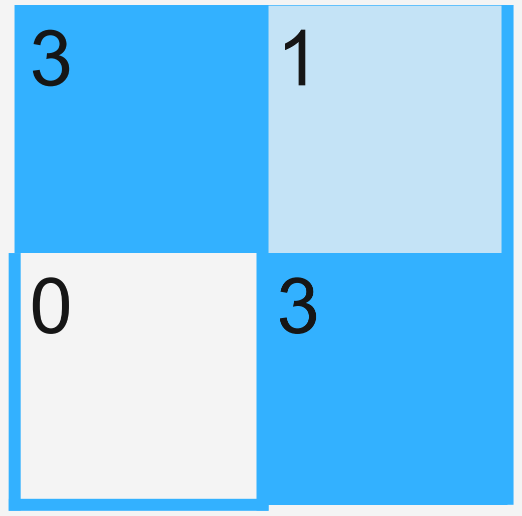
Step 1: Encode the classical data into a quantum state

Step 2: Apply a parameterized model

Step 3: Measure the circuit to extract labels

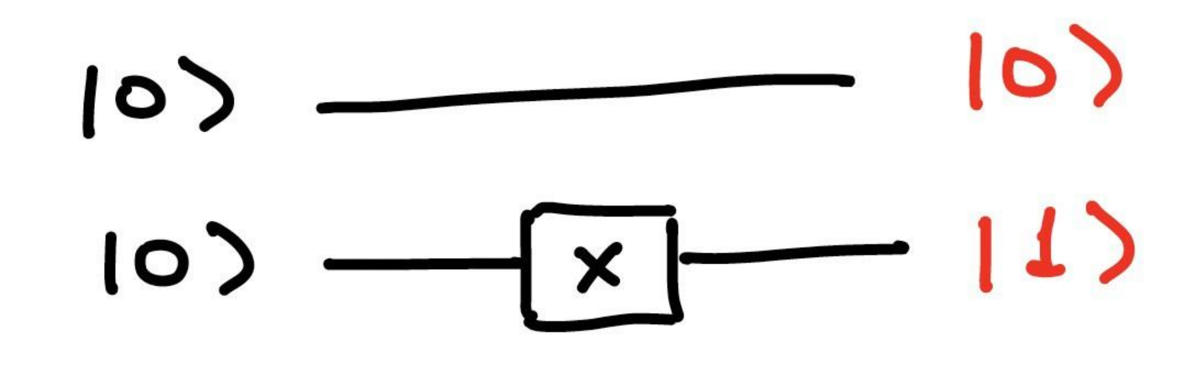
Step 4: Use optimization techniques (like gradient descent) to update model parameters

Data encoding



Basis encoding: Encode each *n*-bit feature into *n* qubits

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 0 \\ 3 \end{bmatrix} = \begin{bmatrix} 11 \\ 01 \\ 00 \\ 11 \end{bmatrix} = \begin{bmatrix} |11\rangle \\ |01\rangle \\ |00\rangle \\ |11\rangle \end{bmatrix}$$



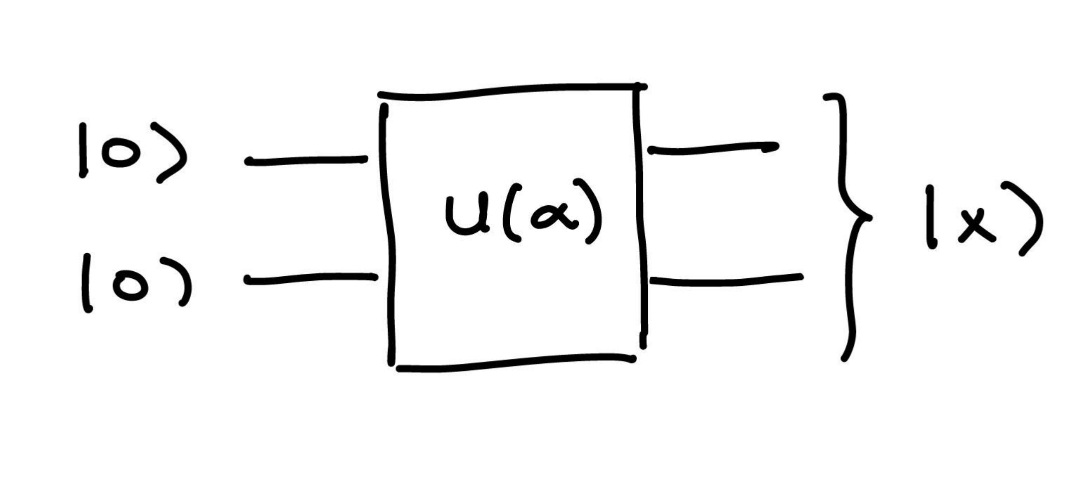
One of the computational basis states of 8 qubits

Data encoding

3	1
0	3

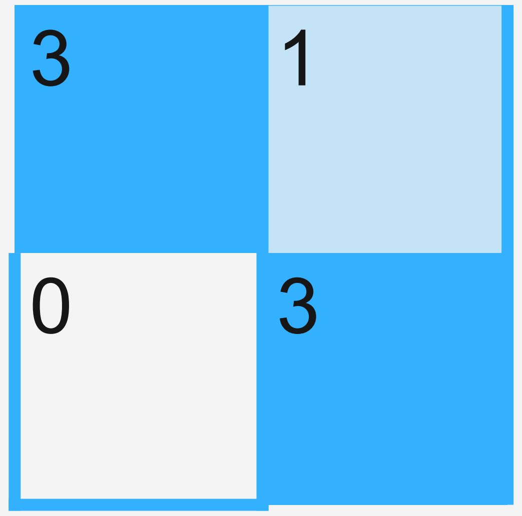
Amplitude encoding: Encode into quantum state amplitudes $|\psi_x\rangle = \sum_{i=1}^{N} x_i |i\rangle$

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3/\sqrt{19} \\ 1/\sqrt{19} \\ 0/\sqrt{19} \\ 3/\sqrt{19} \end{bmatrix}$$



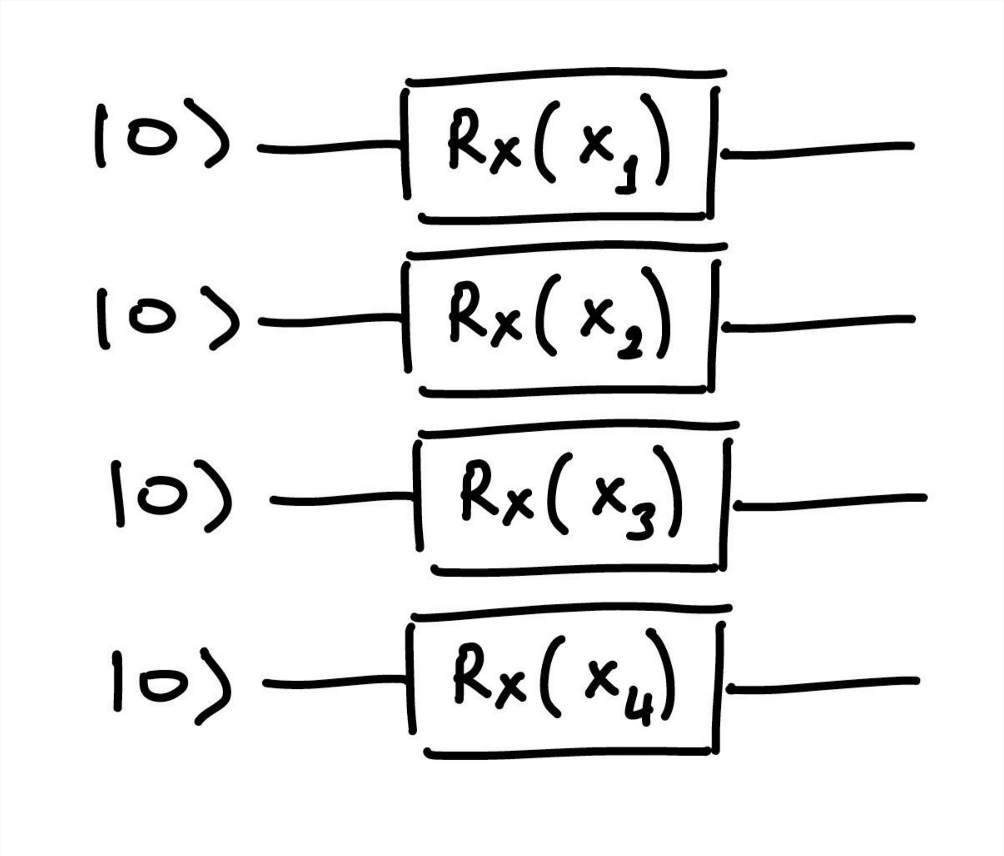
Amplitudes of 2 qubits

Data encoding



Angle encoding: Encode values into qubit rotation angles

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3/\sqrt{19} \\ 1/\sqrt{19} \\ 0/\sqrt{19} \\ 3/\sqrt{19} \end{bmatrix}$$



$$|x\rangle = \bigotimes_{i=1}^{N} \cos(x_i)|0\rangle + \sin(x_i)|1\rangle$$

angle encoding

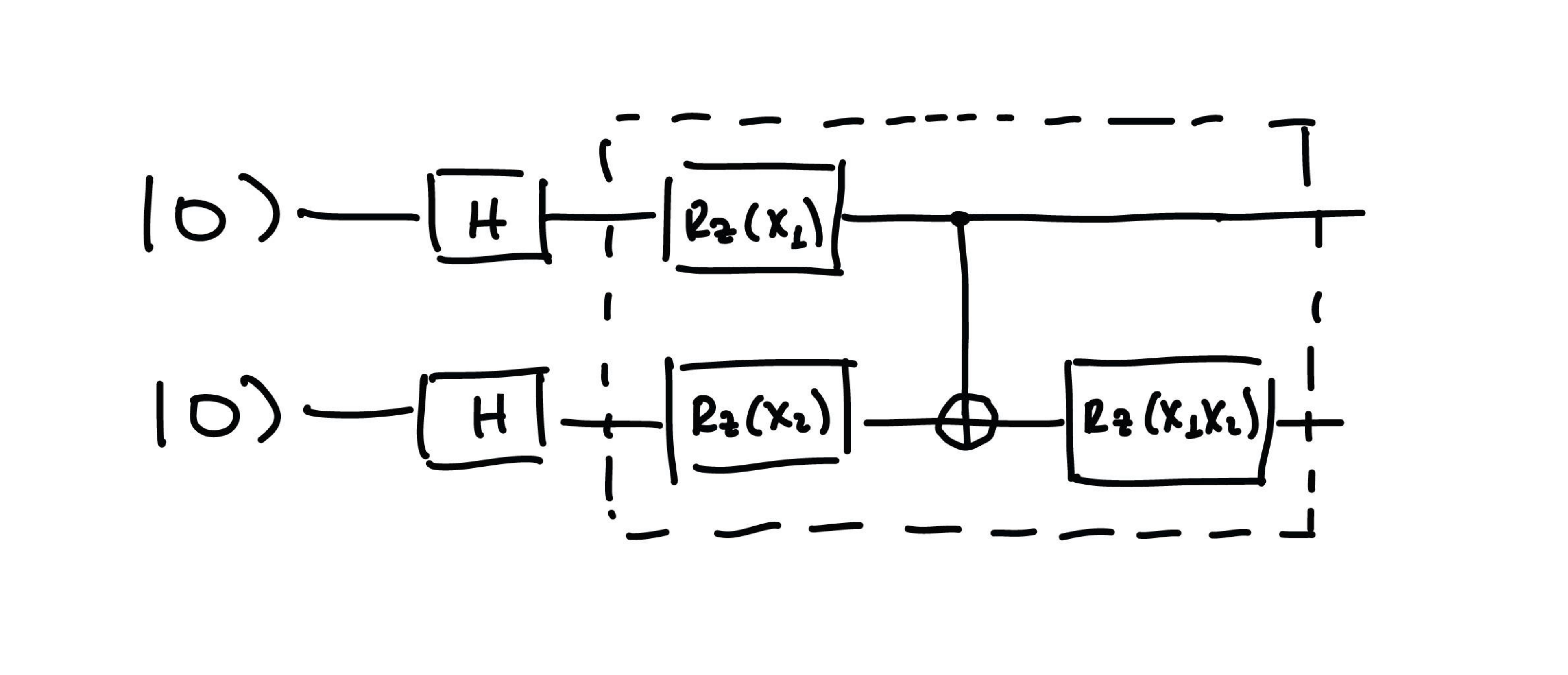
$$|x\rangle = \bigotimes_{i=1}^{n} \cos(x_{2i-1})|0\rangle + e^{ix_{2i}} \sin(x_{2i-1})|1\rangle$$

dense angle encoding

Data encoding

Higher order encoding: Feature maps

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



blocks can be repeated

Havlíček, Vojtěch, et al. "Supervised learning with quantum-enhanced feature spaces." Nature 567.7747 (2019): 209-212.

Data encoding

Basis Encoding

Encode each *n*-bit feature into *n* qubits

$$x = (x_{n-1}, \cdots, x_1, x_0) \rightarrow |x\rangle = |x_{n-1} \ \cdot \cdots x_1 x_0\rangle$$

Amplitude Encoding

Encode into quantum state amplitudes

$$x = \begin{pmatrix} x_0 \\ \vdots \\ x_{n-1} \end{pmatrix} \rightarrow |x\rangle = \sum_{j=0}^{n-1} x_j |\mathrm{j}\rangle$$

Amplitude Encoding

Encode values into qubit rotation angles

$$|x\rangle = \bigotimes_j \cos(x_i)|0\rangle + \sin(x_i)|1\rangle$$

Arbitrary Encoding (Feature Map)

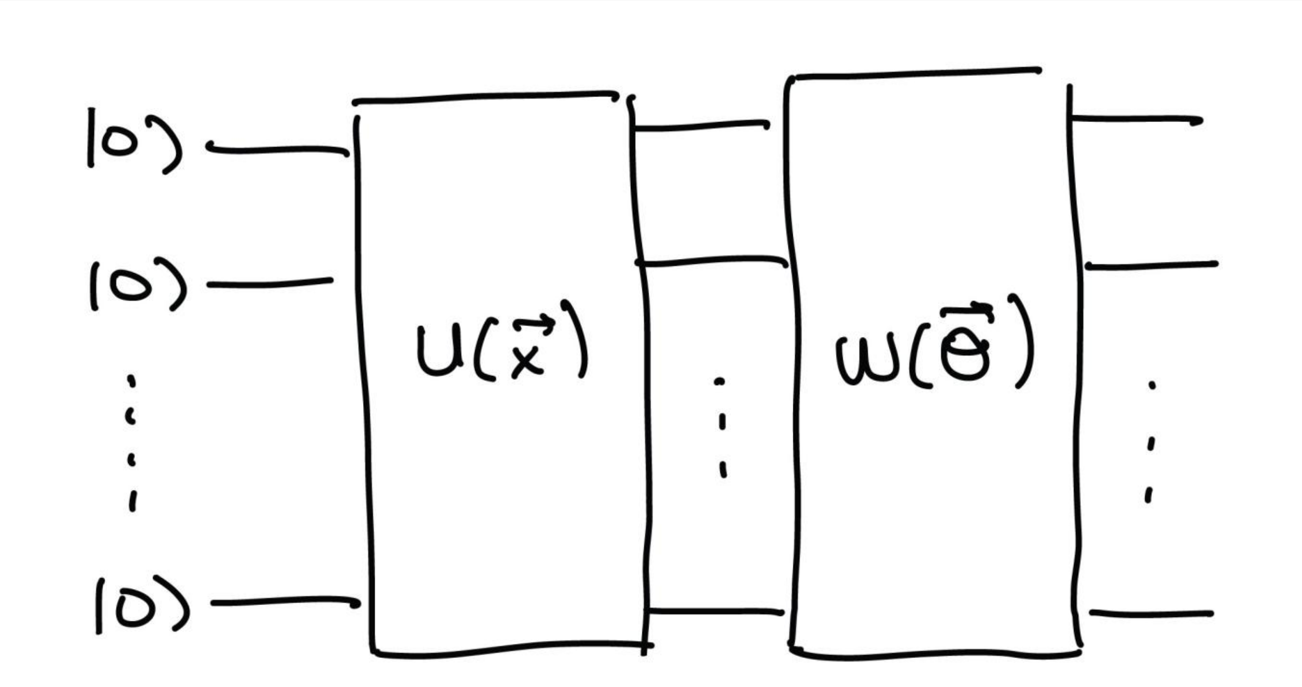
Encode N features on N rotation gates in constant-depth circuit with n qubits

$$x = \begin{pmatrix} x_0 \\ \vdots \\ x_{n-1} \end{pmatrix} \rightarrow |\psi_x\rangle = U_{\Phi(x)}|0\rangle$$

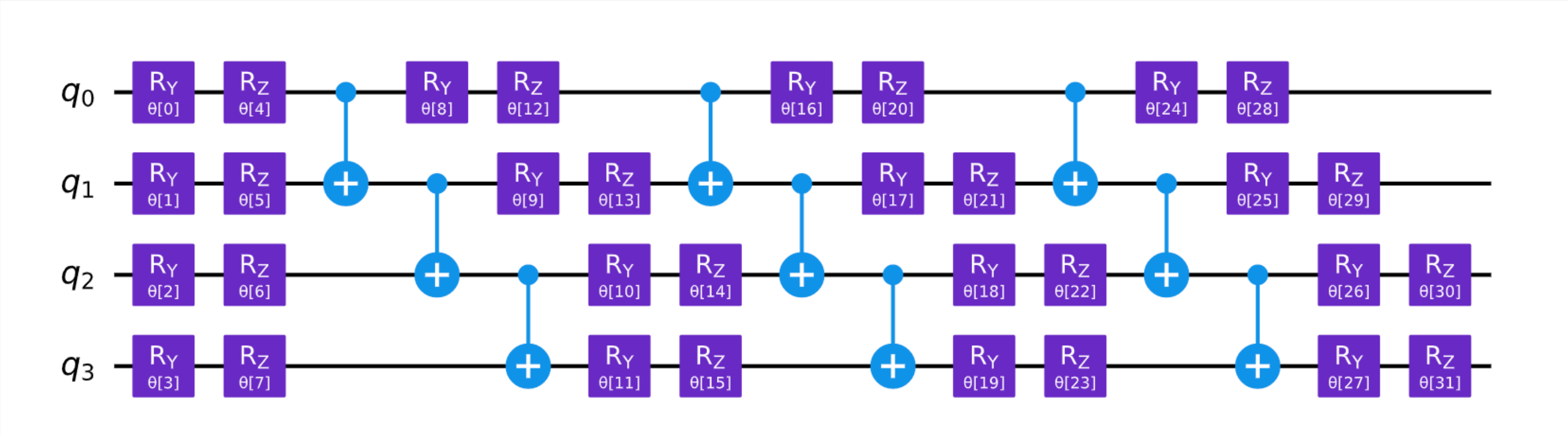
Encoding	# Qubits	State prep runtime
Basis	$n\,N$	$O\,(\,N\,)$
Amplitude	$\log\,(N)$	$\frac{O(N)}{O(\log(N))}$
Angle	N	$O\,(\,N\,)$
Arbitrary	n	$O\,(\,N\,)$

N features each

Variational model



Data encoding Ansatz

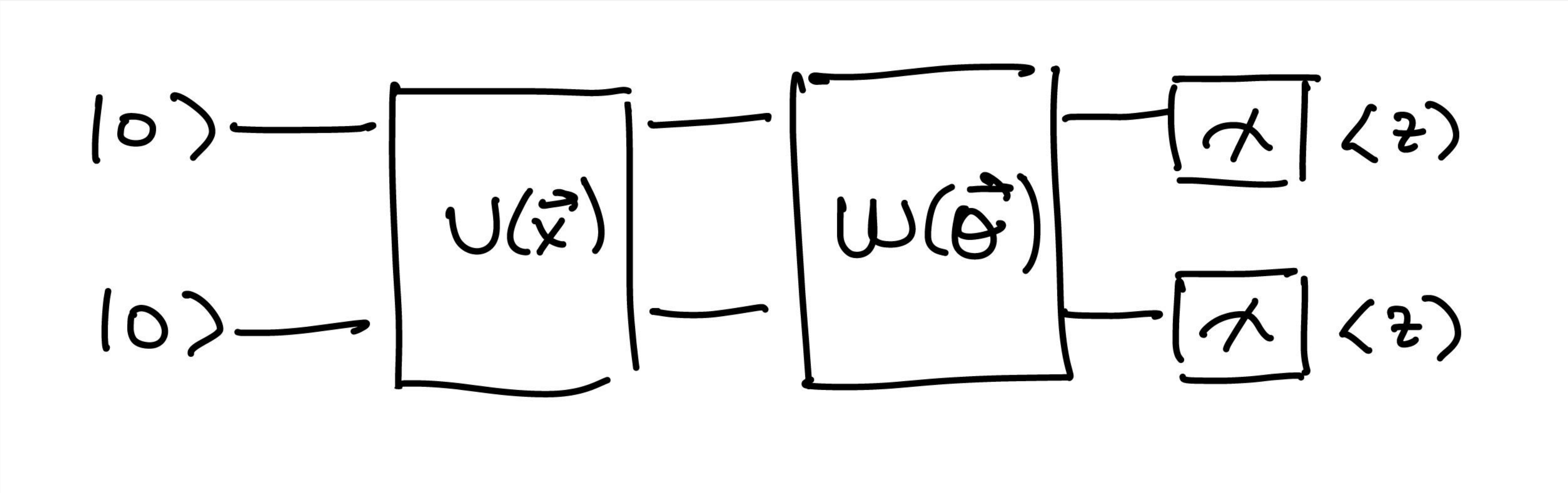


Goal: designing a hardware-efficient ansatz

expressivity and depth

Leone, Lorenzo, et al. "On the practical usefulness of the hardware efficient ansatz." arXiv preprint arXiv:2211.01477 (2022).

Extracting labels



measurement outcomes → labels → cost function → classical optimizer

Binary classification {1,-1}:

1. Parity post-processing (00, 01, 10, 11)
2. Measure only 1 qubit (<Z> >= 0, otherwise)

Qiskit

sampler

estimator

Optimization: parameter update

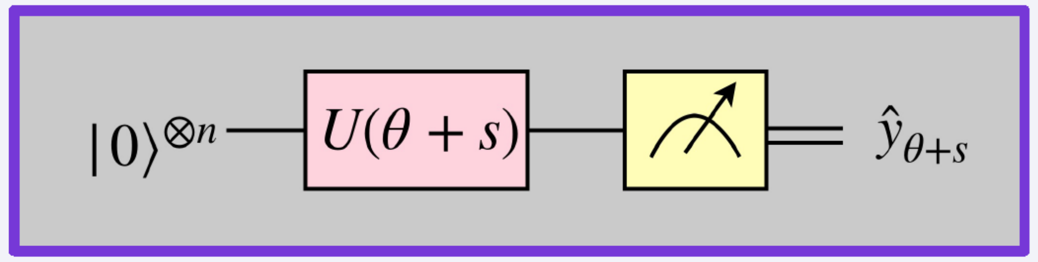
e.g. Mean squared error

Cost: $$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2.$$

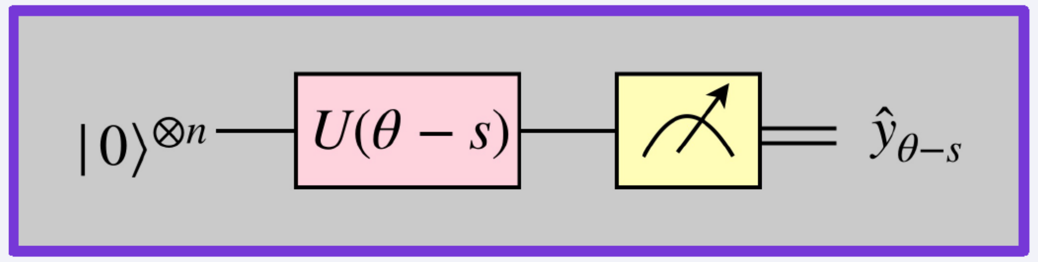
If optimizer needs: $\partial_{\theta_i} f(\boldsymbol{\theta})$

Parameter-shift rule

Gradient =



−



/ 2

$s = \pi/2$

Schuld, Maria, et al. "Circuit-centric quantum classifiers." Physical Review A 101.3 (2020): 032308.