Zurich University
of Applied Sciences

Quantum Machine Learning –
The Next Big AI Wave in the Age of Energy Transition?

Prof. Dr. Kurt Stockinger

Datalab

Zurich University of Applied Sciences

Swiss Data Science Conference (SDS 2023)
Zurich, June 23, 2023

Zurich University of Applied Sciences

Tokomak of ITER for Generating Fusion Energy

ITER ("The Way" in Latin) is one of the most ambitious energy projects in the world today

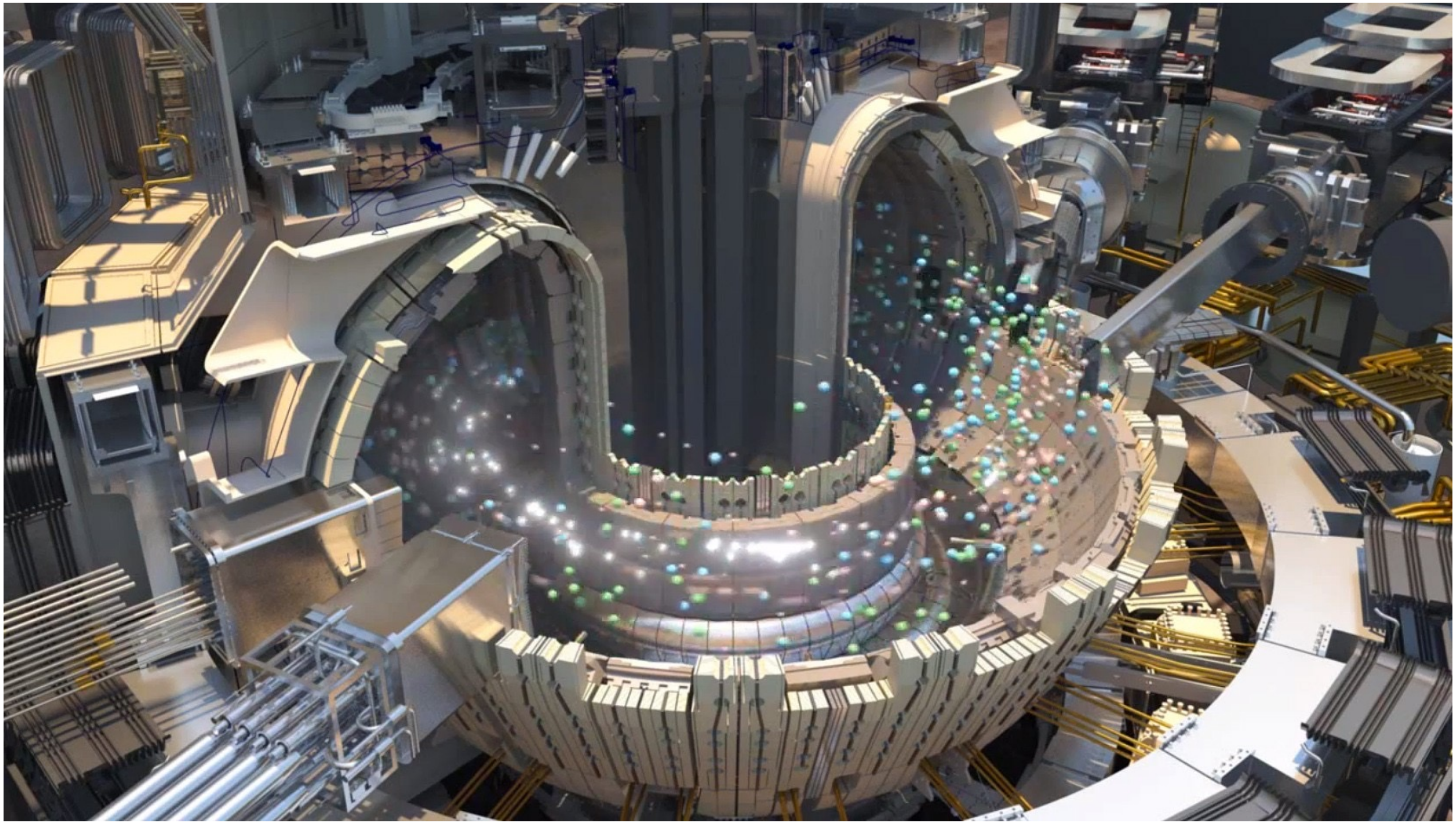
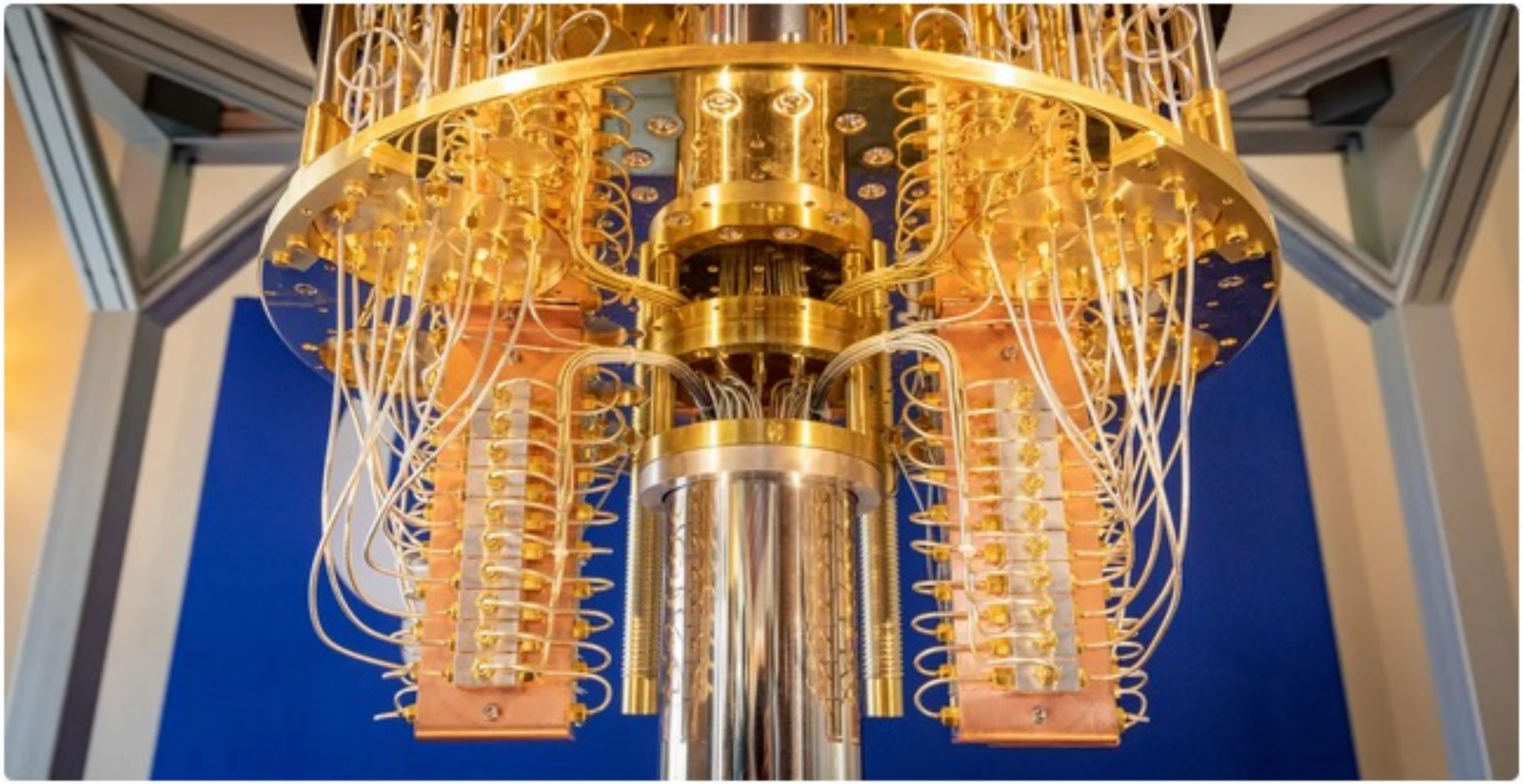


Image source: https://www.iter.org/proj/inafewlines

Zurich University
of Applied Sciences

Quantum Computer

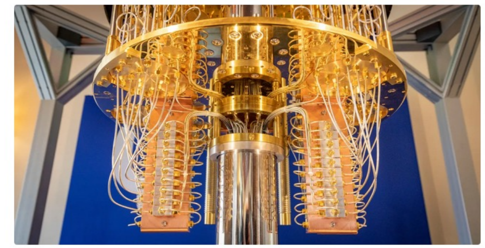


A close-up view of an IBM quantum computer. The processor is in the silver-colored cylinder.

Stephen Shankland/CNET

Contents

- What is quantum computing?
- How do we write a quantum program?
- How do we implement quantum machine learning?



A close-up view of an IBM quantum computer. The processor is in the silver-colored cylinder.
Stephen Shankland/CNET

A Major Leap in Quantum Computing

Zurich University of Applied Sciences

nature

Explore our content ⌄ Journal information ⌄

nature > articles > article

Article | Published: 23 October 2019

Quantum supremacy using a programmable superconducting processor

Frank Arute, Kunal Arya, [...] John M. Martinis

Nature **574**, 505–510(2019) | Cite this article

Abstract

The promise of quantum computers is that certain computational tasks might be executed exponentially faster on a quantum processor than on a classical processor[1]. A fundamental challenge is to build a high-fidelity processor capable of running quantum algorithms in an exponentially large computational space. Here we report the use of a processor with programmable superconducting qubits[2,3,4,5,6,7] to create quantum states on 53 qubits, corresponding to a computational state-space of dimension 2^{53} (about 10^{16}). Measurements from repeated experiments sample the resulting probability distribution, which we verify using classical simulations. Our Sycamore processor takes about 200 seconds to sample one instance of a quantum circuit a million times—our benchmarks currently indicate that the equivalent task for a state-of-the-art classical supercomputer would take approximately 10,000 years. This dramatic increase in speed compared to all known classical algorithms is an experimental realization of quantum supremacy[8,9,10,11,12,13,14] for this specific computational task, heralding a much-anticipated computing paradigm.

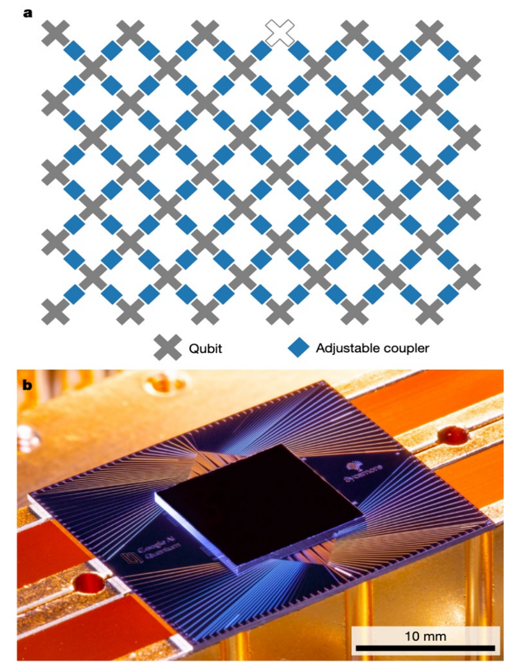


Fig. 1 | The Sycamore processor. a, Layout of processor, showing a rectangular array of 54 qubits (grey), each connected to its four nearest neighbours with couplers (blue). The inoperable qubit is outlined. **b,** Photograph of the Sycamore chip.

The Most Powerful Quantum Computer Currently

Zurich University of Applied Sciences

REUTERS® World Business Markets Sustainability Legal Breakingviews Technology Investigations

Disrupted

IBM launches its most powerful quantum computer with 433 qubits

By Jane Lee

November 9, 2022 3:07 PM GMT+1 · Updated 7 months ago

[1/2] A computer rendering shows IBM's 433-qubits Osprey quantum processor, with more than three times the qubits of the IBM Eagle processor unveiled in 2021, in this undated handout image. Connie Zhou for IBM/Handout via REUTERS

Nov 9 (Reuters) - International Business Machines Corp on Wednesday said it launched its most powerful quantum computer to date called the Osprey, a 433-qubit machine that has three times the number of qubits than its Eagle machine announced last year.

The number of qubits, or quantum bits, are an indication of the power of the quantum computer which uses quantum mechanics, although different quantum computer companies make different claims about the power of their qubits which can be created many different ways.

Major Concepts of Quantum Computing

- Qubits
- Superposition
- Entanglement
- Quantum circuits

Classical Bit vs. Quantum Bit (Qubit) #1

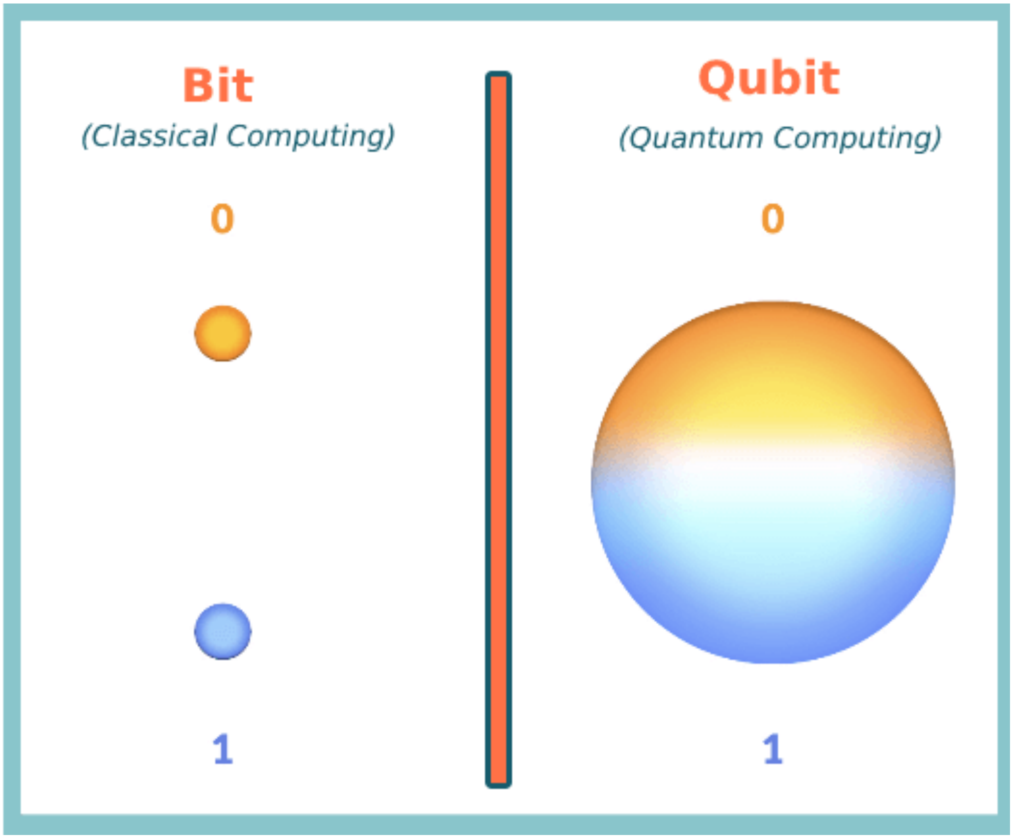
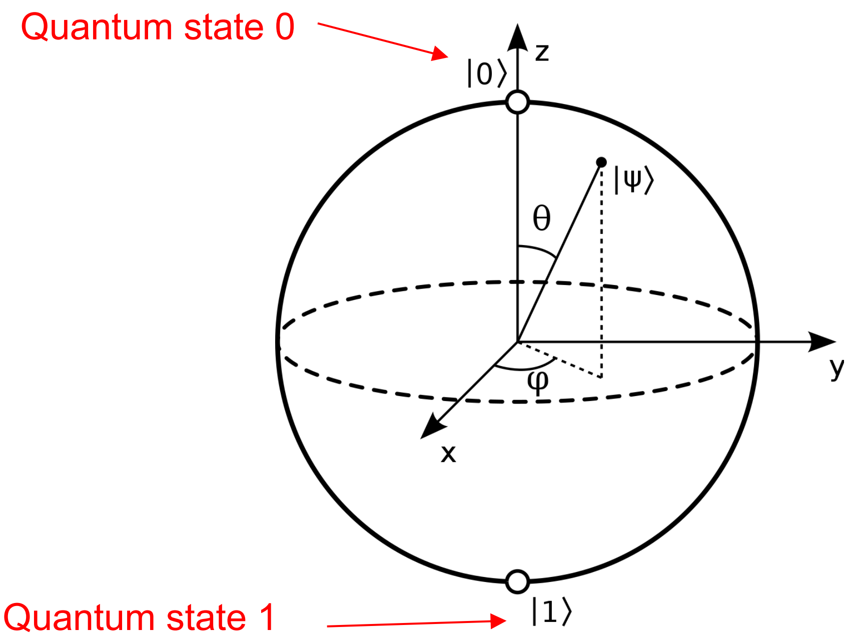


Image source: https://cheapsslsecurity.com/blog/quantum-computing-vs-encryption-a-battle-to-watch-out-for/

Classical Bit vs. Quantum Bit (Qubit) #2
Superposition on the Bloch Sphere



Vectors show the state |ψ> of a quantum system

Superposition = weighted sum of two states, i.e. a linear combination of 0 and 1 (quantum randomness)

Entanglement #1

State 1:

Entanglement #2

State 1:

rotation

State 2:

?

Entanglement #3

State 1:

State 2:

A Quantum Circuit Consists of Quantum Gates

- Pauli-Gates: Rotation gates
- Hadamard-Gate: Creates superposition
- Controlled-Gates: Create entanglement

Contents

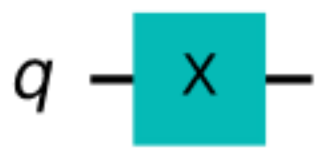
Quantum Programming with Qiskit

- Qiskit = Quantum Information Science Kit
- Released by IBM in 2017

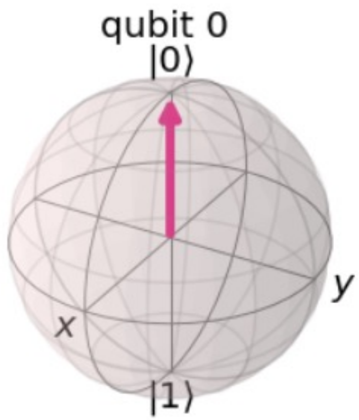
Working with Single Qubit Gates

- Pauli Gates:
  - X-Gate: NOT-gate, rotation around x-axis
  - Y-Gate: rotation around y-axis
  - Z-Gate: rotation around z-axis (phase flip, + becomes -)

```
# Let's do an X-gate on a |0> qubit
qc = QuantumCircuit(1)
qc.x(0)
qc.draw()
```

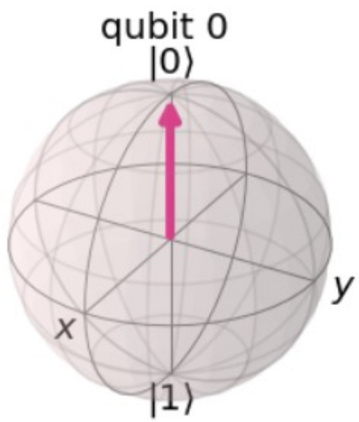


Showing the Result on the Bloch Sphere #1



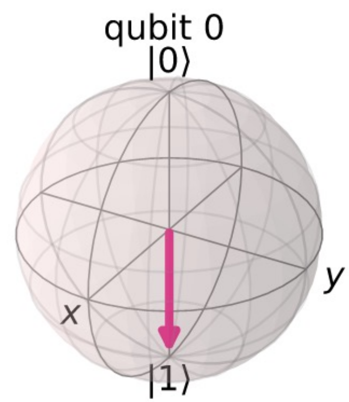
Initial state

Showing the Result on the Bloch Sphere #2



Initial state

apply X-gate →



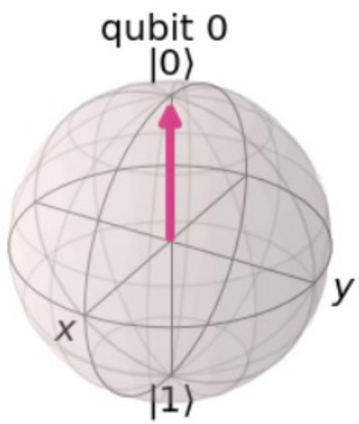
State after applying X-gate

Applying Y-Gate on Qubit 0 #1

Zurich University
of Applied Sciences

```
qc = QuantumCircuit(1)
qc.y(0)
qc.draw()
```

How does the state on the Bloch sphere look like?

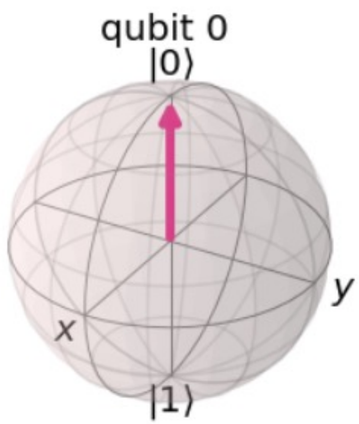


Initial state

Applying Y-Gate on Qubit 0 #2

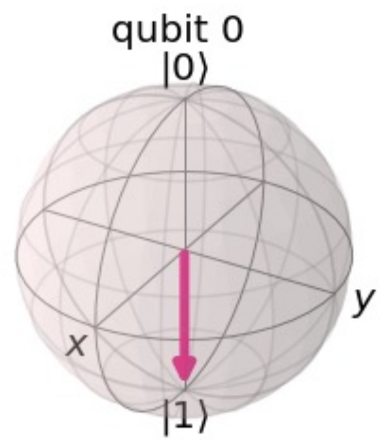
```
qc = QuantumCircuit(1)
qc.y(0)
qc.draw()
```

How does the state on the Bloch sphere look like?



Initial state

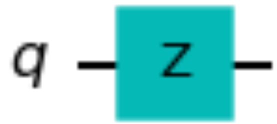
apply Y-gate



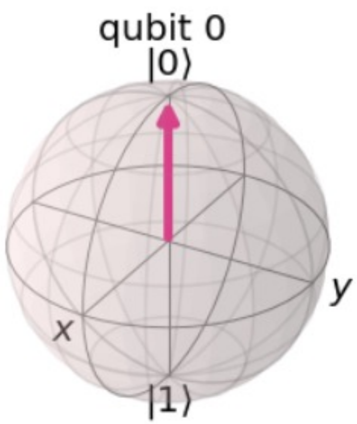
Same result as applying X-gate

Applying Z-Gate on Qubit 0 #1

```
qc = QuantumCircuit(1)
qc.z(0)
qc.draw()
```



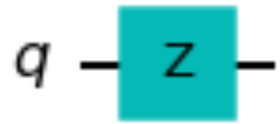
How does the state on the Bloch sphere look like?



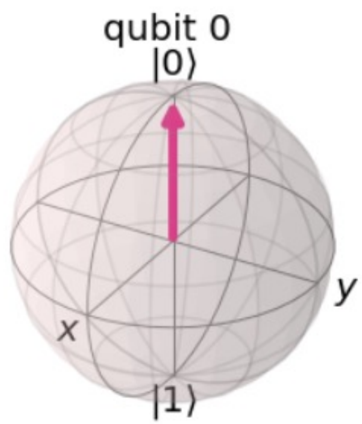
Initial state

Applying Z-Gate on Qubit 0 #2

```
qc = QuantumCircuit(1)
qc.z(0)
qc.draw()
```

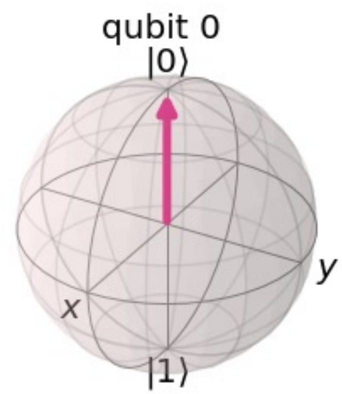


How does the state on the Bloch sphere look like?



Initial state

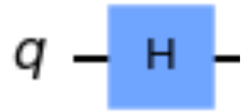
apply Z-gate



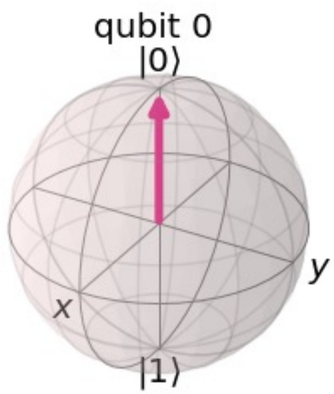
No change

Create Superpositions with Hadamard-Gate #1

```
qc = QuantumCircuit(1)
qc.h(0)
qc.draw()
```

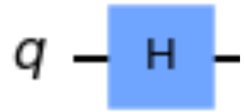


How does the state on the Bloch sphere look like?

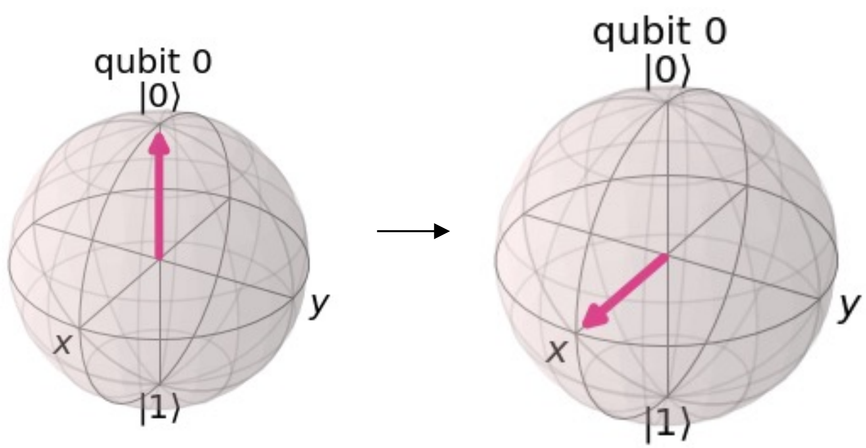


Create Superpositions with Hadamard-Gate #2

```
qc = QuantumCircuit(1)
qc.h(0)
qc.draw()
```



How does the state on the Bloch sphere look like?



State is in a superposition

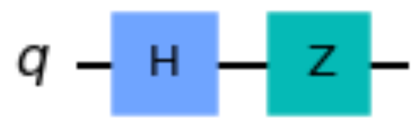
between |0> and |1>

(similar to a coin flip the probability

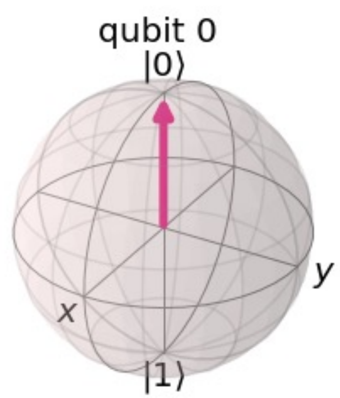
of 0 and 1 is 50/50)

Combination of H-Gate and Z-Gate #1

```
qc = QuantumCircuit(1)
qc.h(0)
qc.z(0)
qc.draw()
```

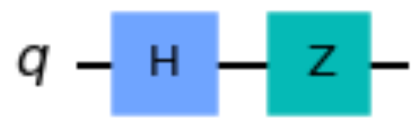


How does the state on the Bloch sphere look like?

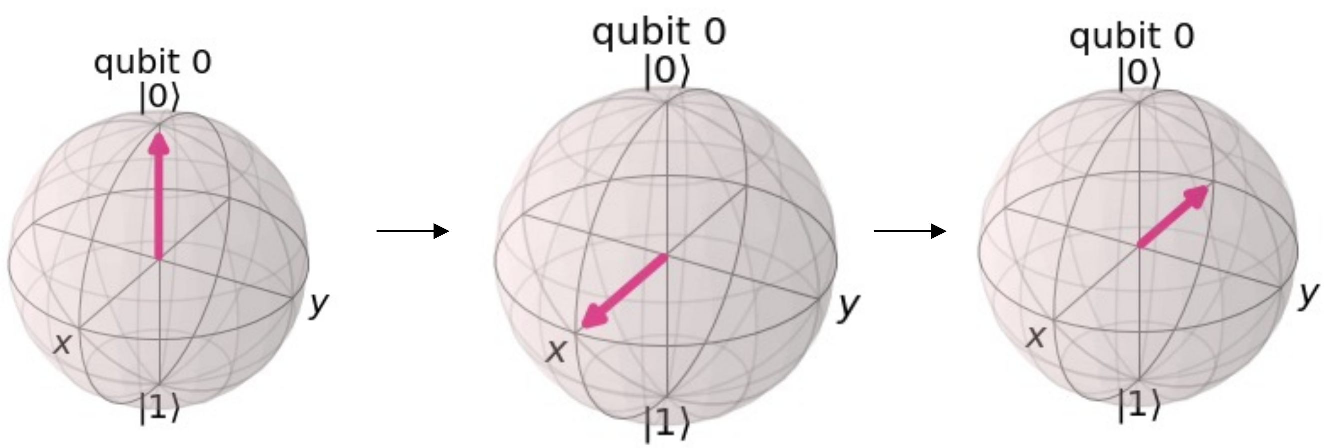


Combination of H-Gate and Z-Gate #2

```
qc = QuantumCircuit(1)
qc.h(0)
qc.z(0) # phase rotation by pi
qc.draw()
```



How does the state on the Bloch sphere look like?



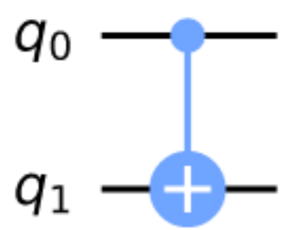
Multi-Qubit Gates Create Entanglement

- CNOT-gate:
  - Conditional gate
  - Performs X-gate (NOT) on second qubit (target),
    if state of first qubit (control) is 1

```
qc = QuantumCircuit(2)

# Apply CNOT
qc.cx(0,1)

# See the circuit:
qc.draw()
```



Assumption: qubits not in superposition

Input (t,c)	Output (t,c)
00	00
01	11
10	10
11	01

First qubit (least significant)

Second qubit

Amplitudes of 01 and 11 are swapped (|+>)

Contents

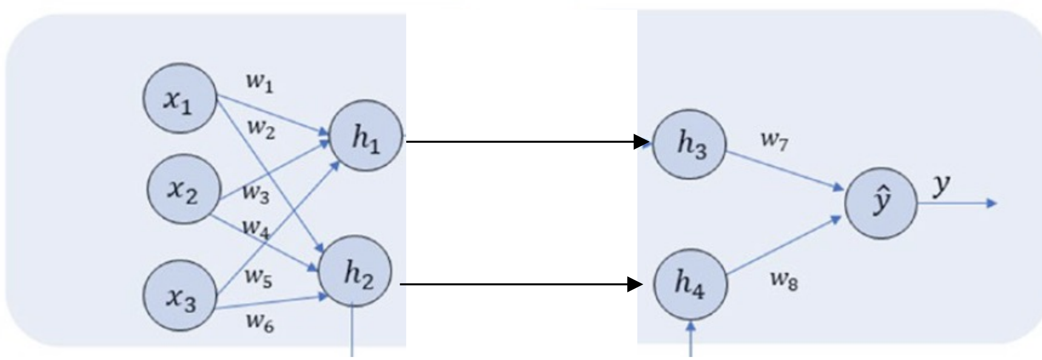
- What is quantum computing?
- How do we write a quantum program?
- How do we implement quantum machine learning?

Quantum Machine Learning

- It is not clear how to best implement neural networks on a quantum computer: open research question
- The field is still in its infancy
- Most approaches are theoretical based on quantum simulators or experimental quantum hardware
- However, there are promising approaches for small problems

Approach 1:
Hybrid Classical-Quantum Neural Network #1

Classical neural network



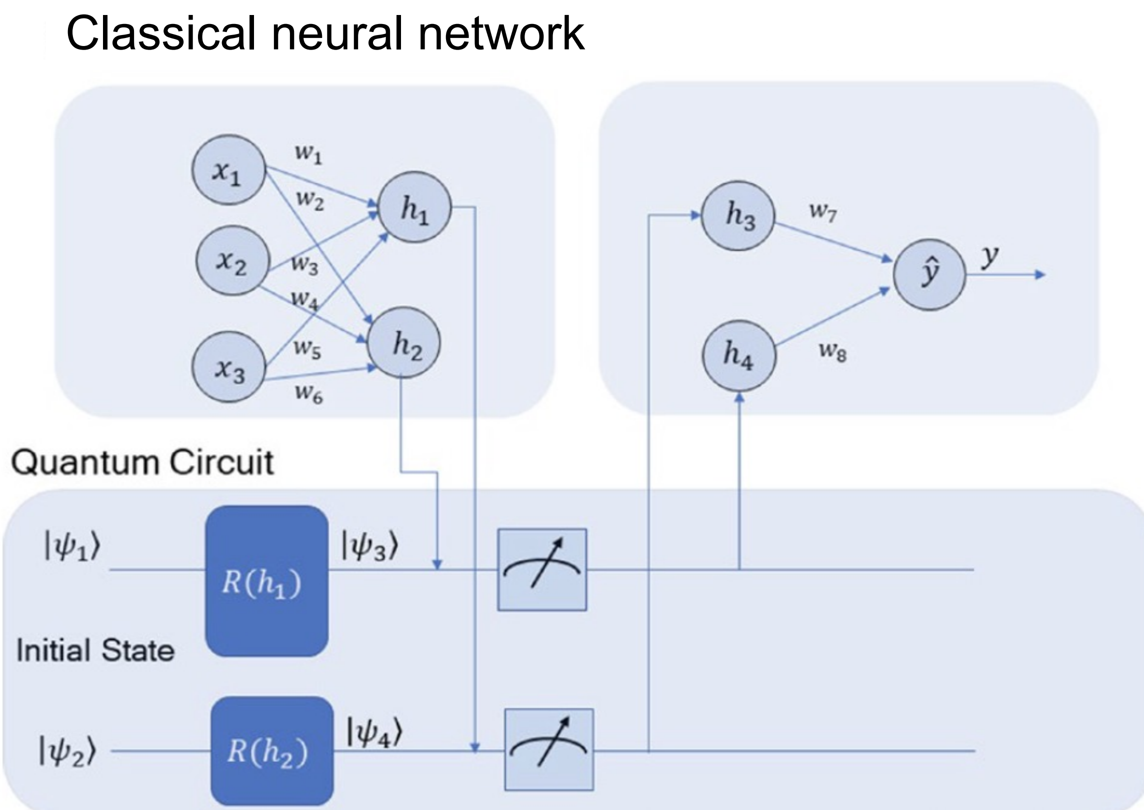
x … input

w … weights

h … hidden layers

y … output

Approach 1:
Hybrid Classical-Quantum Neural Network #2



x … input

w … weights

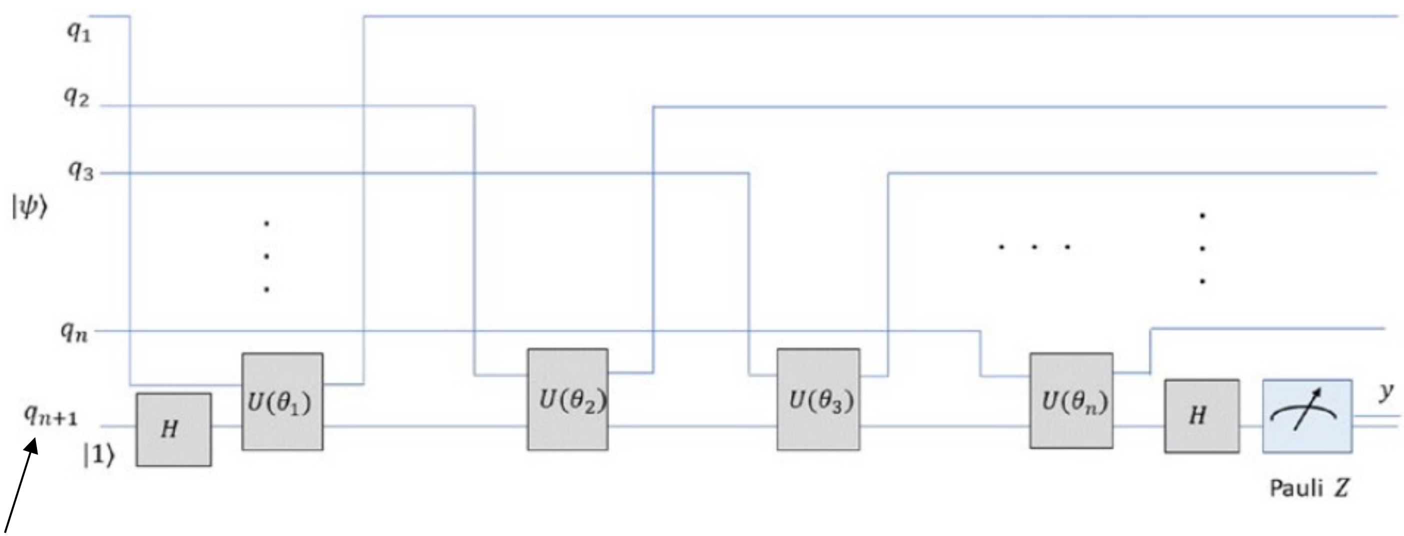
h … hidden layers

y … output

ψ… quantum state

Approach 2:
Quantum Neural Network with Unitary Layers

- The whole network is implemented as a parameterized quantum circuit
- QNN with i layers: $U(\theta) = U_l(\theta_l)U_{l-1}(\theta_{l-1})\ldots U_1(\theta_1)$
  - U ... unitary transformation
  - $\theta = [\theta_L, \theta_{L-1}, \ldots . \theta_1]^T$ set of parameters for the QNN



Readout qubit: After applying i unitary transformations, the state of q_{n+1} should correspond to the real label

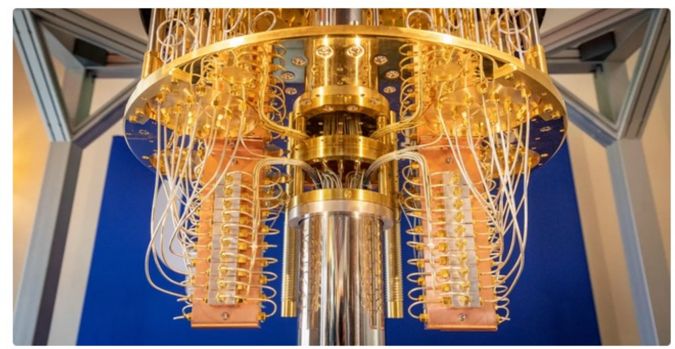
Evaluation of Quantum Machine Learning - Datasets

Dataset	#Features	#Records	#Classes
Iris	4	100	2
Rain	5	100	2
Vlds	5	100	2
Custom	2	100	2
Adhoc	3	100	2

Software and Hardware

- Qiskit:
  - Python library for quantum computing by IBM
- Quantum simulator:
  - By IBM
  - Can be installed locally or run on in cloud
- Quantum computer:
  - By IBM
  - Publicly available via cloud

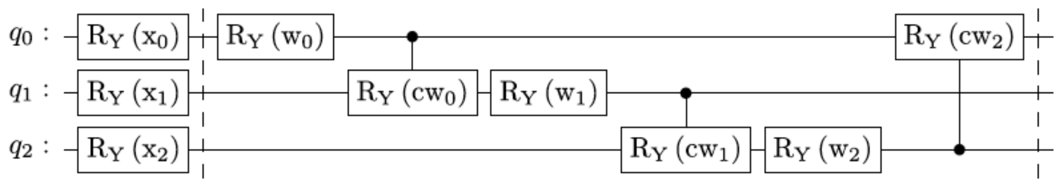
https://qiskit.org/



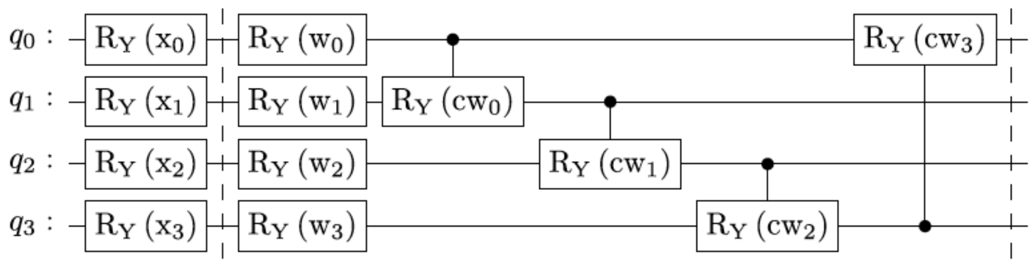
A close-up view of an IBM quantum computer. The processor is in the silver-colored cylinder.
Stephen Shankland/CNET

Zurich University
of Applied Sciences

Evaluation of Different Quantum Neural Networks

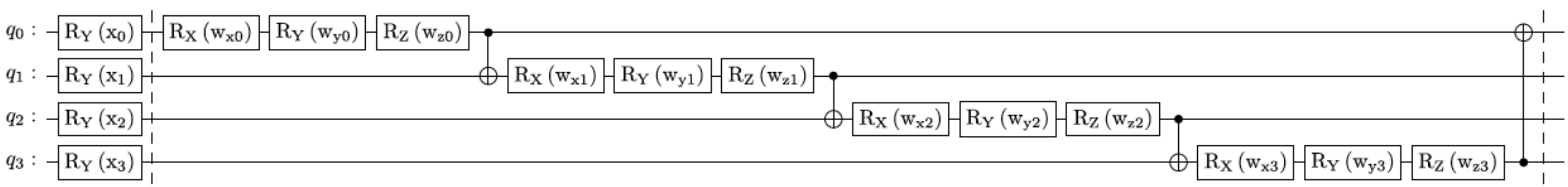


Circuit 1



Circuit 2

Circuit 5



Zurich University of Applied Sciences

zh aw

Experimental Results #1

Metric = accuracy (between 0 and 1): higher is better

Dataset	Classical NN	QNN (Quantum Simulator)	QNN (Quantum Computer)
Iris	1.00	1.00	1.00
Rain	0.70	0.83	0.79
Vlds	0.94	0.93	0.95
Custom	0.64	0.74	0.75
Adhoc	0.61	0.80	0.75
Average	**0.78**	**0.86**	**0.85**

Quantum neural network (QNN) outperforms classical neural network (NN) on specific datasets

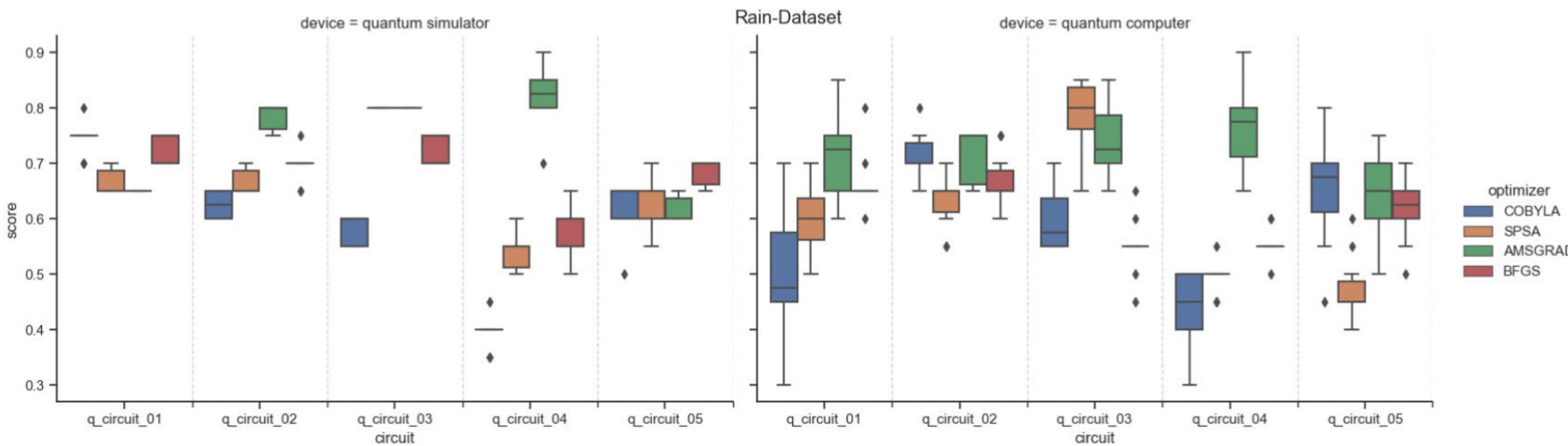
R. D. M. Simões, P. Huber, N. Meier, N. Smailov, R. M. Füchslin and K. Stockinger, "**Experimental Evaluation of Quantum Machine Learning Algorithms**," in *IEEE Access*, vol. 11, pp. 6197-6208, 2023, doi: 10.1109/ACCESS.2023.3236409.

Experimental Results #2
Details on the Rain Dataset

Zurich University of Applied Sciences

zh aw

Comparison of 5 different quantum circuits on
quantum simulator (left) and quantum computer (right)



We can observe a high fluctuation of the results.

score = accuracy (higher is better)

Conclusions

- Quantum machine learning is still in its infancy
- Currently we can only solve small problems
- Quantum hardware needs to mature and become more fault-tolerant
- There is a steep learning curve to get into the topic
- First results are very promising
- Early movers have an advantage

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 December 2022, accepted 26 December 2022, date of publication 12 January 2023, date of current version 20 January 2023.
Digital Object Identifier 10.1109/ACCESS.2023.3236409

APPLIED RESEARCH

Experimental Evaluation of Quantum Machine Learning Algorithms

RICARDO DANIEL MONTEIRO SIMÕES[1], PATRICK HUBER[1], NICOLA MEIER[1], NIKITA SMAILOV[1], RUDOLF M. FÜCHSLIN[1,2], AND KURT STOCKINGER[1]
[1]School of Engineering, ZHAW Zurich University of Applied Sciences, 8401 Winterthur, Switzerland
[2]European Centre for Living Technology, 30123 Venice, Italy
Corresponding author: Kurt Stockinger (Kurt.Stockinger@zhaw.ch)

ABSTRACT Machine learning and quantum computing are both areas with considerable progress in recent years. The combination of these disciplines holds great promise for both research and practical applications. Recently there have also been many theoretical contributions of quantum machine learning algorithms with experiments performed on quantum simulators. However, most questions concerning the potential of machine learning on quantum computers are still unanswered such as *How well do current quantum machine learning algorithms work in practice? How do they compare with classical approaches?* Moreover, most experiments use different datasets and hence it is currently not possible to systematically compare different approaches. In this paper we analyze how quantum machine learning can be used for solving small, yet practical problems. In particular, we perform an experimental analysis of kernel-based quantum support vector machines and quantum neural networks. We evaluate these algorithm on 5 different datasets using different combinations of quantum feature maps. Our experimental results show that quantum support vector machines outperform their classical counterparts on average by 3 to 4% in accuracy both on a quantum simulator as well as on a real quantum computer. Moreover, quantum neural networks executed on a quantum computer further outperform quantum support vector machines on average by up to 5% and classical neural networks by 7%.

https://ieeexplore.ieee.org/document/10015720