

Secrets revealed in this session:

To explore the principles of quantum machine learning models, their parameterisation and optimisation



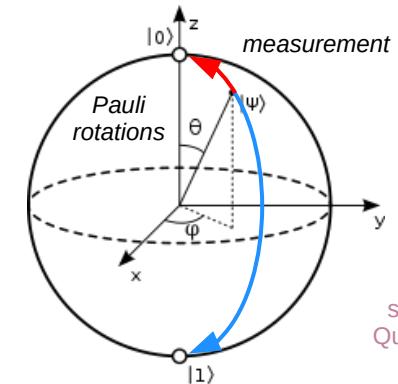
QML workshop
QML team
QML and aims
Parameterised circuits
Data encoding
 Angle encoding
 The good, the bad and the ugly
State measurement
Quantum model training
Parameters optimisation
Model geometry and gradients
QML readings
PennyLane demo
Summary

Quantum Machine Learning

Introduction

Jacob L. Cybulski

Enquantum, Melbourne, Australia



We will assume some knowledge of Quantum Computing ML and Python

Our QML Team



Sebastian Zajac

Assistant Professor
SGH Warsaw School of
Economics
[LinkedIn](#)



Dr Paweł Gora

Coordinator of QPoland and
CEO of Fundacja Quantum AI
[LinkedIn](#)



Tomek Rybotycki

QML Researcher
SRI PAS, NCAC PAS, CEA AGH, KPLabs
[LinkedIn](#)

Jacob Cybulski
Founder, Researcher, Consultant
at Enquanted

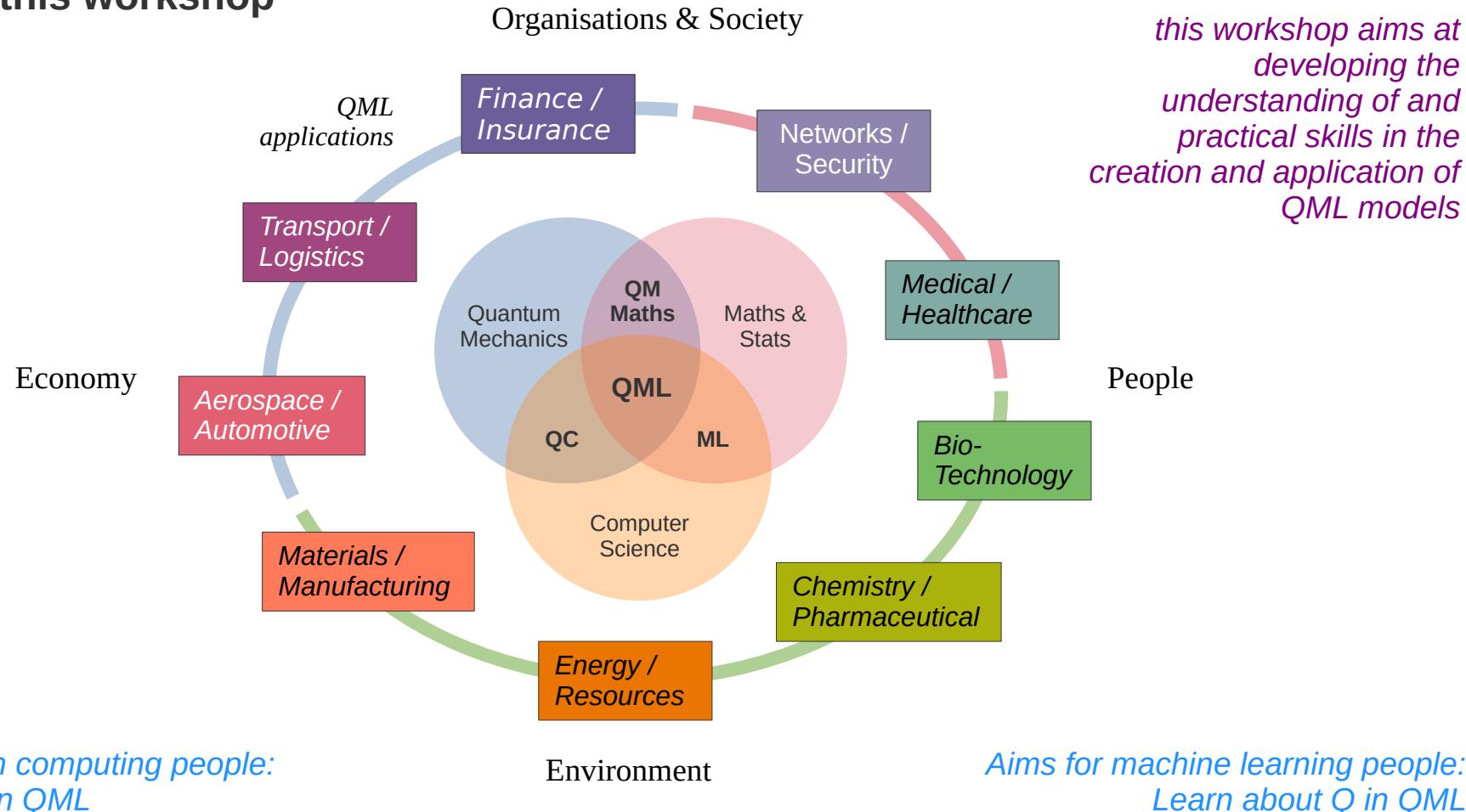
and
Honorary A/Prof
In Quantum Computing
Deakin University
[LinkedIn](#)



Quantum ML

aims of this workshop

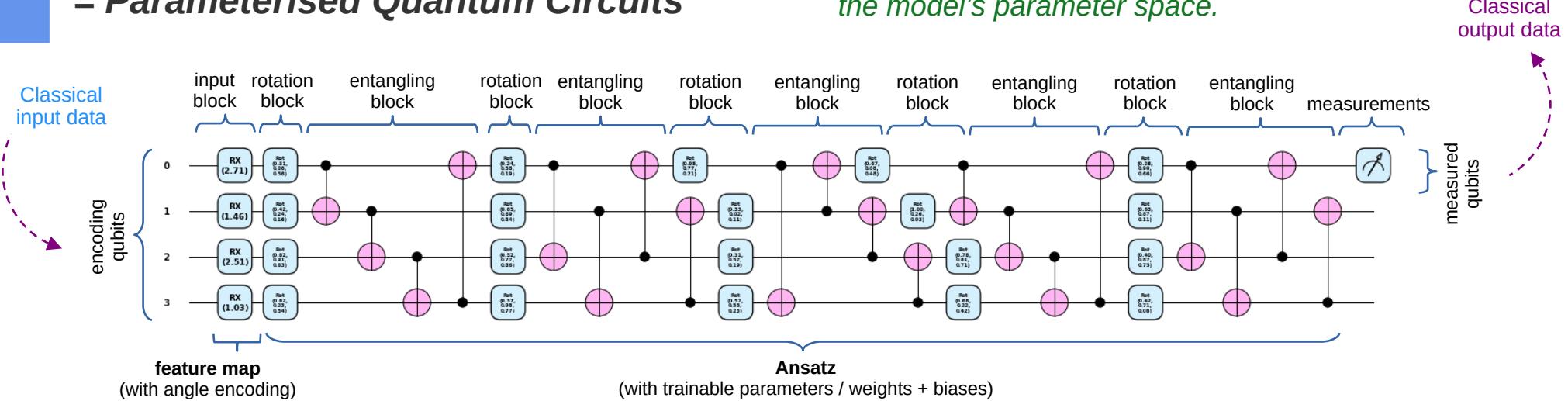
Jacob L. Cybulski, Quantum Business Series (Deakin, RMIT, ACS, Warsaw School of Economics)
Jacob L. Cybulski, Quantum Computing Intro Series (SheQuantum, Assoc of Polish Profs in Australia)
2021-2025



Variational Quantum Models

= Parameterised Quantum Circuits

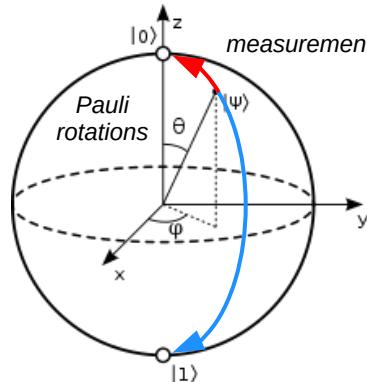
*Ansatz parameters are trainable.
Each parameter defines a dimension in
the model's parameter space.*



We can create a “variational” model = a circuit template with parameterised gates, e.g. **P(a)**, **Ry(a)** or **Rz(a)**, each allowing rotation of a qubit state in x, y or z axis (as per Bloch sphere).

Typically, but now always, the circuit consists of three blocks:

- a **feature map (input)**
- an **ansatz (processing)**
- **measurements (output)**



Classical input data is encoded into the feature map's parameters, setting the model's initial quantum state.

The quantum state is then altered by an ansatz, which consists of parameterised gates (operations), which alter the initial quantum state.

The quantum state of the circuit is then measured and interpreted as the model's output in classical data form, e.g. as binary values, integer or real value, a single event's probability or the probability distribution.

Data encoding strategies

Data encoding

There are many methods of data embedding, such as:
the *basis*, *angle*, *amplitude*, *QRAM*, ... encoding,

In this workshop we will rely on *angle encoding* realised as qubit state rotation by the angle defined by the data.

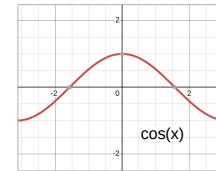
The rotation operators are always available in a quantum platform API (e.g. *Rx*, *Ry*, *Rz* or *Rxyz*).

Typically, the encoding rotation is performed around x or y axis, or both (allowing two values per qubit).

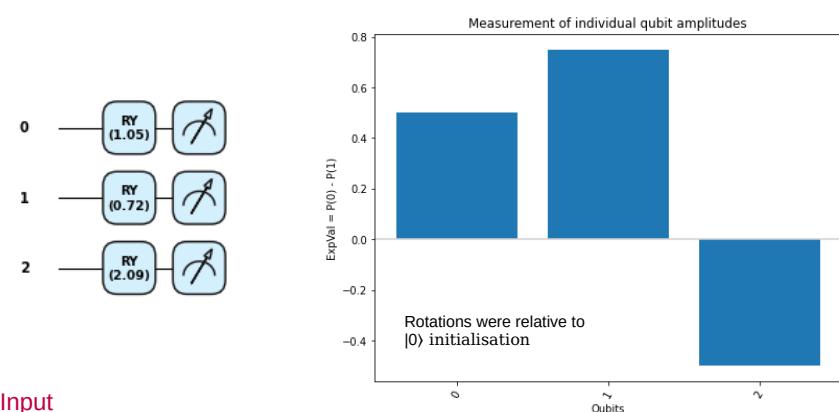
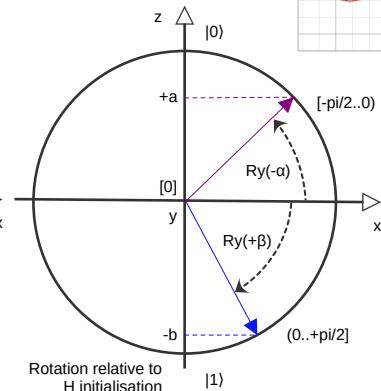
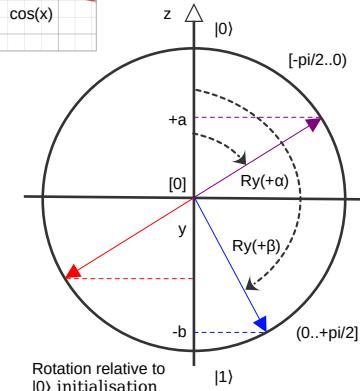
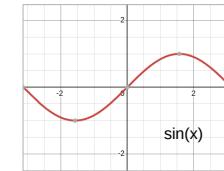
Rotations are *relative to a specific qubit state*, commonly starting at $|0\rangle$ state, or $(|0\rangle+|1\rangle)/\sqrt{2}$, which require qubits to be initialised in these states.

The encoded value could be represented either by the *angular rotation*, or the *amplitude* of the qubit projective measurement (Z).

In some cases, input data is repeatedly encoded and interspersed with ansatz layers, called *data reuploading*, which improves the model performance.



Note that training will place qubit states in areas $x < 0$ and arbitrarily around the z axis. Measurements of such states cannot distinguish them from "pure" $x > 0$ and $z = 0$.



Input

Values entered:
Ry angles used:

[np.arccos(0.5), np.arccos(0.75), np.pi-np.arccos(0.5)]
[1.047, 0.723, 2.094]

Measurements

Probabilities:
Amplitudes:

[[0.25, 0.75], [0.562, 0.438], [0.25, 0.75]]
[0.5, 0.75, -0.5]

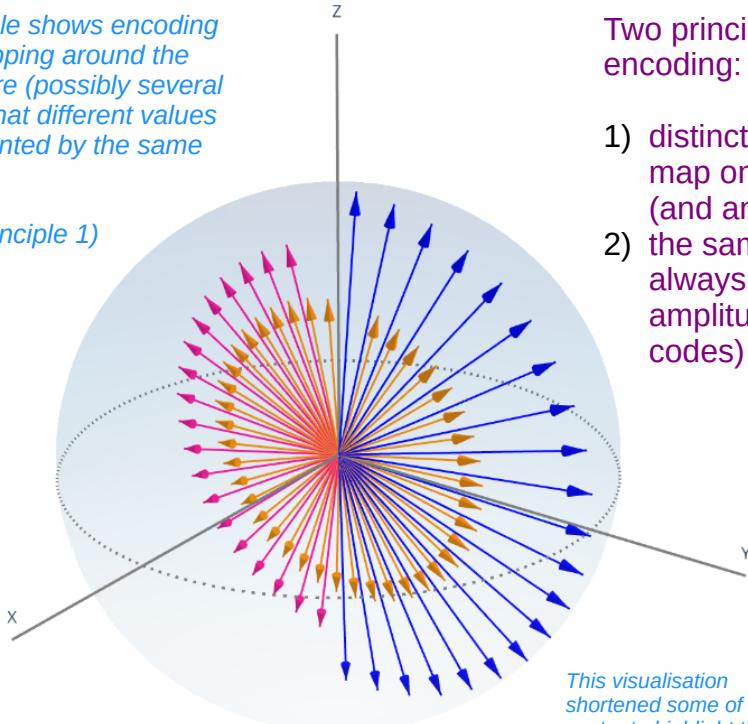
Angle encoding

The Good, the Bad and the Ugly

Angle Range: -6 to 6
 $0..\pi$ = "blue" (long) | $>\pi$ = "deeppink" (medium) | $-\pi..0$ = "darkorange" (short)

This example shows encoding values wrapping around the Bloch sphere (possibly several times), so that different values are represented by the same amplitude.

(violates principle 1)



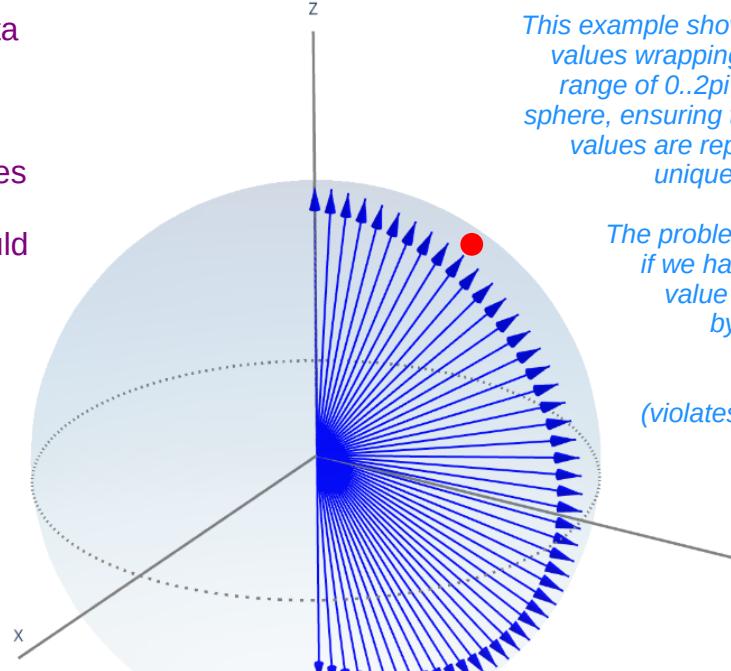
Two principles of quantum data encoding:

- 1) distinct data values should map onto distinct amplitudes (and angular codes)
- 2) the same data values should always map into identical amplitudes (and angular codes)

Angle Range: 0 to 3.141592653589793
 $0..\pi$ = "blue" (long) | $>\pi$ = "deeppink" (medium) | $-\pi..0$ = "darkorange" (short)

This example shows encoding values wrapping around the range of 0.2π of the Bloch sphere, ensuring that different values are represented by unique amplitudes.

The problem may arise if we have the same value represented by two distinct amplitudes



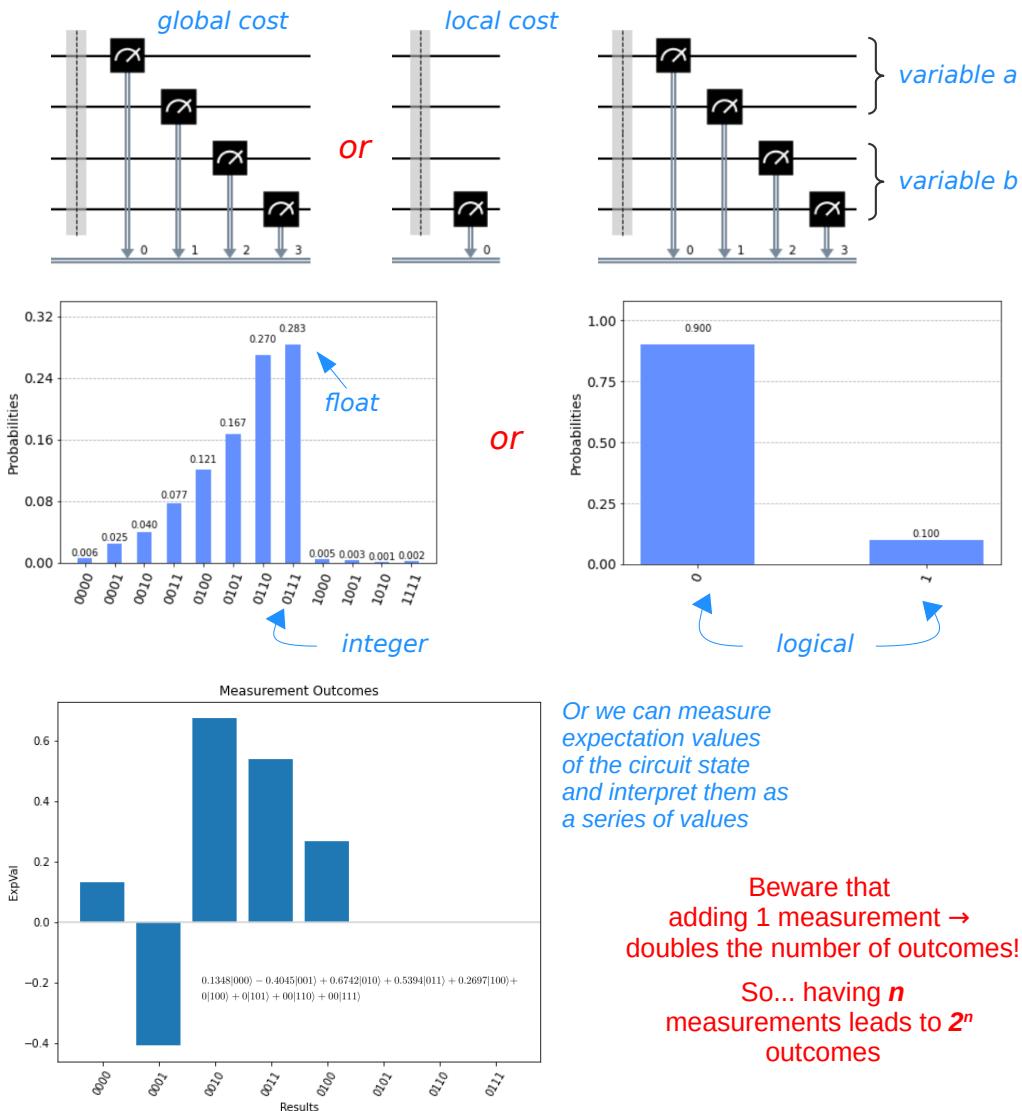
Commonly used measurements and interpretation

There are many ways of obtaining the outcome of a circuit execution, e.g. we can measure:

- all qubits (global cost / measurement)
- a few selected qubits (local cost / measurement)
- groups of qubits (each as a variable value)
- as counts of outcomes (repeated measurements)
- as probabilities of outcomes (e.g. $P(|0111\rangle)$)
- as Pauli expectation values (i.e. of eigenvalues)
- as expectation of interpreted values (e.g. 0 to 15)
- as variance, etc.

Repeated circuit measurement can be interpreted as outcomes of different types, e.g.

- as a probability distribution (as is)
- as a series of values (via expvals)
- as a binary outcome:
single qubit measurement or parity of kets
- as an integer:
most probable ket in multi-qubit measurement
- as a continuous variable:
probability of the selected ket (e.g. $|0^n\rangle$)

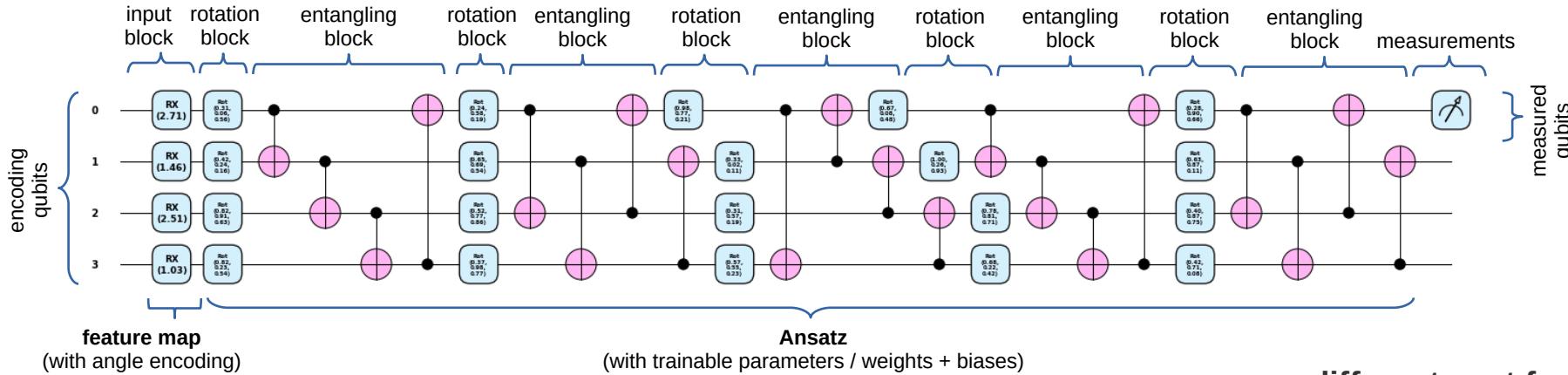


Ansatz design and training

A simple quantum classifier ...

Beware that
adding qubits adds
parameters and entanglements!

The number of states represented by the
circuit grows exponentially with the
number of qubits!



feature maps vary in:

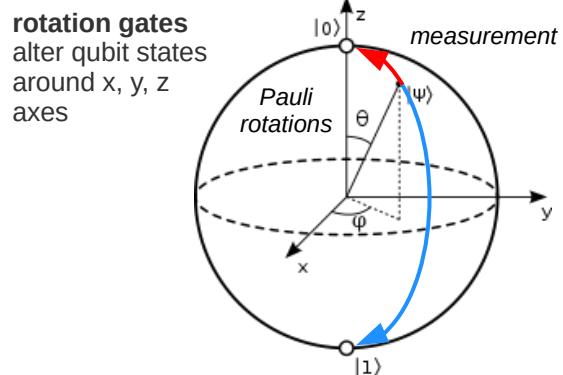
structure and function

ansatze vary in:

- width (qubits #)
- depth (layers #)
- dimensions (param #)
- structure (e.g. funnelling)
- entangling (circular, linear, sca)

ansatz layers consist of:

rotation blocks and entangling blocks
of $R(x, y, z)$ and CNOT gates
(rotation) (entanglement)



To execute a circuit we just apply it to input data
and the optimum parameters

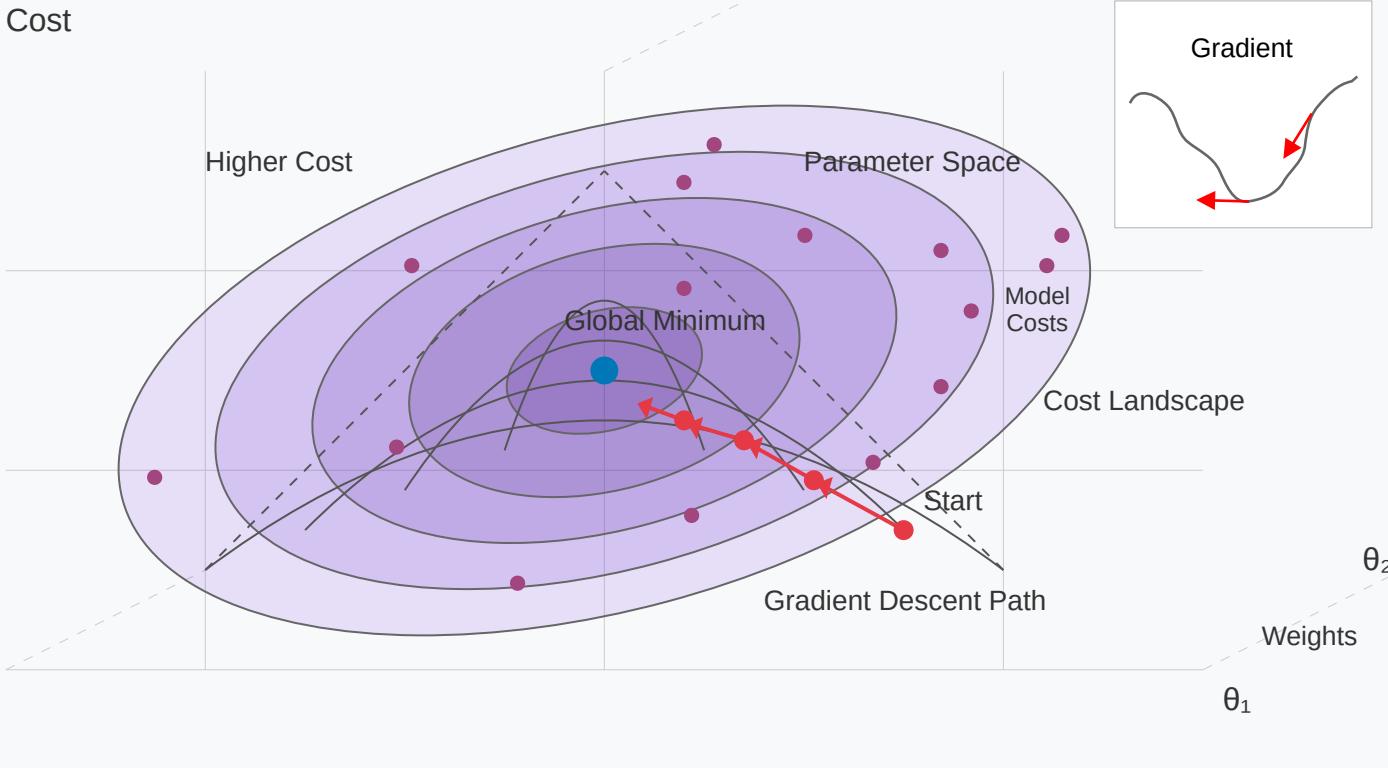
different cost functions:
R2, MAE, MSE, Huber, Poisson, cross-entropy,
hinge-embedding, Kullback-Leibner divergence

different optimisers:
gradient based (Adam, NAdam and SPSA)
linear approximation methods (COBYLA)
non-linear approximation methods (BFGS)
quantum natural gradient optimiser (QNG)

circuit execution on:
simulators (CPUs), accelerators (GPUs) and
real quantum machines (QPUs)

Classical model optimisation

Gradient descent

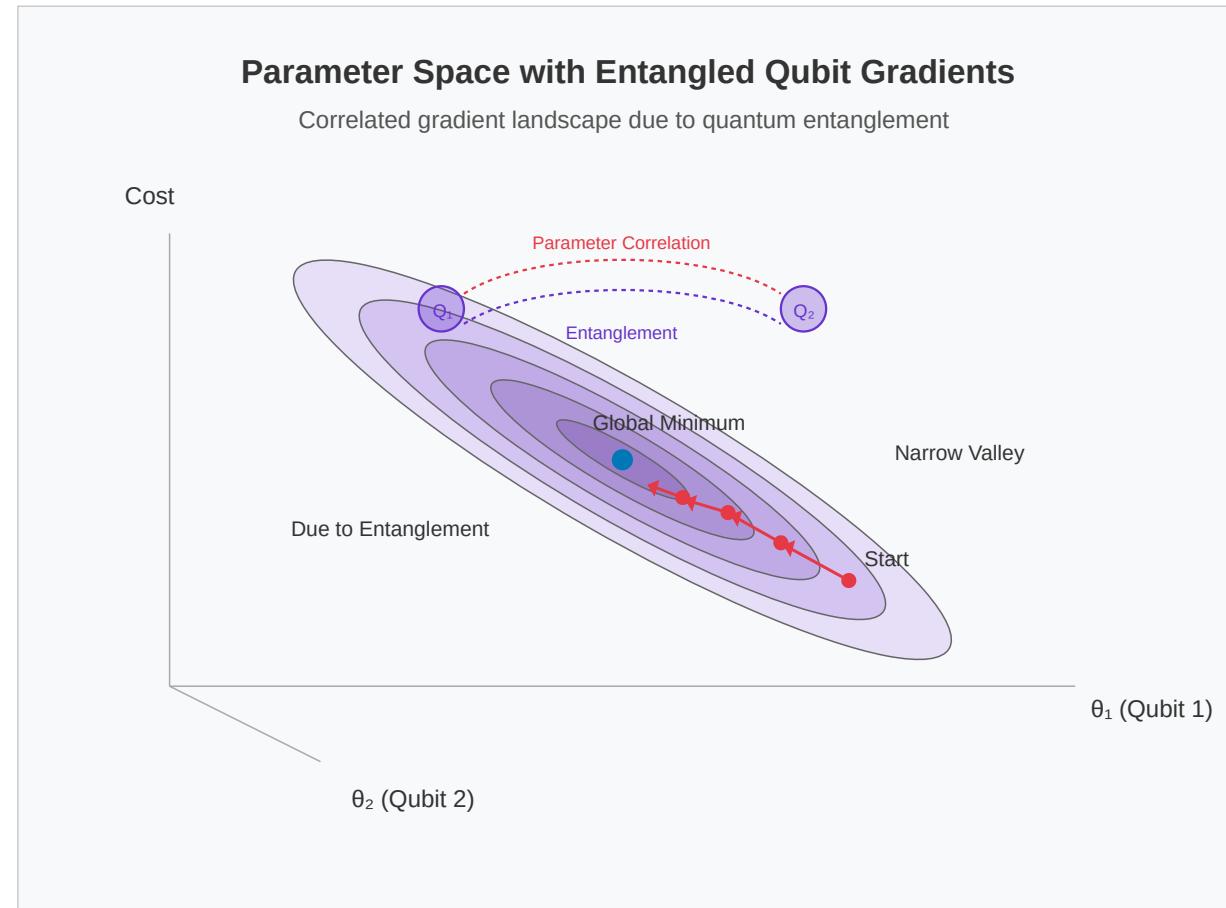


Gradients are local, i.e. their changes influence only their immediate neighbourhood

- A Neural Network, the most commonly used classical ML model, is parameterised with *weights* and *biases*.
- The model quality is linked to the *cost function*, where the lower the cost, the better is the model.
- The costs of all possible model parameterisations, form a multi-dim surface, or the *cost landscape*.
- The *optimisation process* relies on the shape of the landscape, which in turn is reflected in the *gradient* of points on the cost surface.
- *Gradient descent* algorithms can assist in the identification of the model with the *minimum cost*.
- *Backpropagation* can also be used to very efficiently re-calculate NN weights.

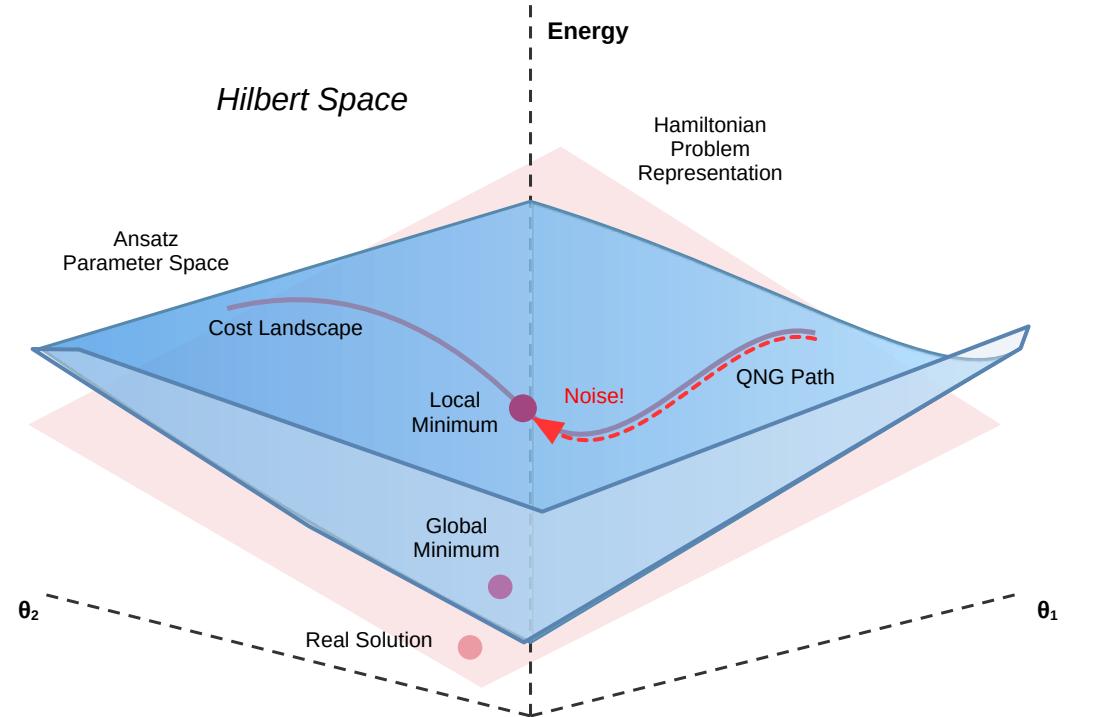
- Optimisation of quantum model needs unique approaches due to the emergence of *non-local gradients*
- *Entangled qubits* result in *correlated parameters and gradients*, so the changes to one are reflected in the distant others
- The cost landscape of highly entangled circuits commonly features *narrow valleys*
- Also, *backpropagation* cannot be used directly in training quantum circuits, as their state is not directly accessible and the measurement collapses the state
- *Gradient descend* can still be used with *global gradients*, i.e. those derived from the geometry of the cost landscape
- *Stochastic optimisation techniques* are highly effective when the cost landscape is smooth (no quantum noise)
- Other techniques are also available, such as *particle swarm optimisers*, these however are applicable to smaller models

Quantum model optimisation



Problem-solving with quantum models

- The abstract mathematical model represents the problem to be solved, e.g. in the form of the *Hamiltonian*
- The Hamiltonian defines some geometry in *Hilbert space* where the optimum solution can be associated with the optimum energy
- The *ansatz* of a quantum circuit approximates the Hamiltonian and therefore the problem
- The parameters of the ansatz define a *parameter space* that intersects the Hamiltonian problem space
- Our search for the problem solution is hence restricted to the ansatz parameter space
- Ideally, the selected *optimiser* and the *cost function* should understand the principles of quantum models, e.g. Quantum Natural Gradient (QNG) optimiser
- The QNG optimiser gradients then define the *cost landscape* (or the manifold), which spans the ansatz geometry
- The QNG optimiser will then identify a *local optimum* solution for this ansatz
- Noise can prevent finding the local optimum



It is far more complex than problem-solving with classical models

Recommended reading on QML

EXPERT INSIGHT

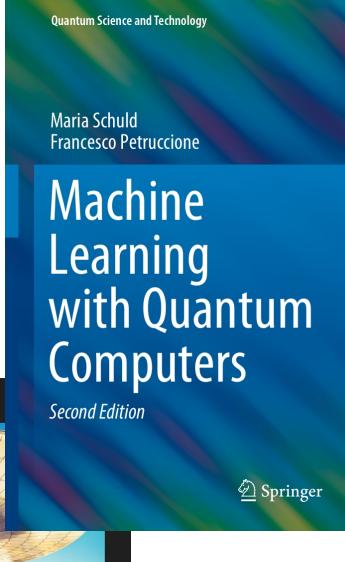
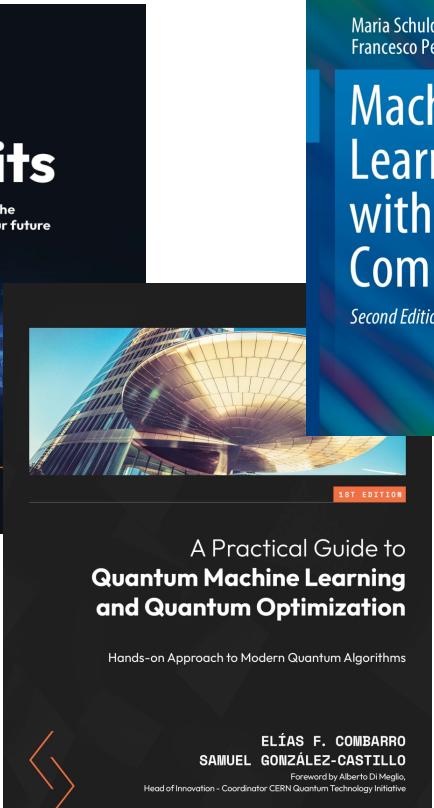
Dancing with Qubits

From qubits to algorithms, embark on the quantum computing journey shaping our future

Second Edition



Robert S. Sutor



PennyLane: Automatic differentiation of hybrid quantum-classical computations

Ville Bergholm,¹ Josh Izaac,¹ Maria Schulz,¹ Christian Gogolin,¹ M. Sohaib Alam,² Shahnaz Ahmad,³ Juan Miguel Arrazola,¹ Carsten Blank,⁴ Alain Delgado,¹ Soran Jahangiri,¹ Keri McKiernan,² Johannes Jakob Meyer,⁵ Zeyue Niu,¹ Antal Száva,¹ and Nathan Killoran¹

¹Xanadu, 777 Bay Street, Toronto, Canada

²Rigetti Computing, 2019 Seventh Street, Berkeley, CA 94710

³Wüllenberg Centre for Quantum Technology, Department of Microtechnology and Nanoscience, Chalmers University of Technology, 412 96 Gothenburg, Sweden

⁴Data Cybernetics, Martin-Kolbengser-Str 26, 68899 Landsberg, Germany

⁵Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, 14195 Berlin, Germany

14 Feb 2020

The framework for optimization and machine learning of quantum and hybrid quantum provides a unified architecture for near-term quantum computing devices, supporting the paradigm. PennyLane's core feature is the ability to compute gradients of variational compatible with classical techniques such as backpropagation. PennyLane thus extends optimizers common in optimization and machine learning to include quantum and hybrid tasks the framework compatible with any gate-based quantum simulator or hardware. We fields, Rigetti Forest, Qiskit, Cirq, and ProjectQ, allowing PennyLane optimizations to be used by devices provided by Rigetti and IBM Q. On the classical front, PennyLane interfaces with libraries such as TensorFlow, PyTorch, and autograd. PennyLane can be used for the eigenvalues, quantum approximate optimization, quantum machine learning models,

Modern applications of machine learning in quantum sciences

Anna Dawid^{1,2*}, Julian Arnold^{3,7}, Borja Requejo^{2,8}, Alexander Gresch⁹, Marcin Płodzień², Kaelan Donatella⁵, Kim A. Nicolaï^{6,7}, Paolo Stornai², Rouven Koch⁸, Miriam Blitner⁹, Robert Okula^{10,11}, Gorika Muñoz-Gill¹², Rodrigo A. Vargas-Hernández^{13,14}, Alba Cervera-Lierta¹⁵, Juan Carrasquilla¹⁴, Vedran Dunjko¹⁶, Marylou Gabrié¹⁷, Patrick Huembeli¹⁸, Evert van Nieuwenburg^{16,20}, Filippo Vicentini¹⁸, Lei Wang^{11,22}, Sebastian J. Wetzel²³, Giuseppe Carleo²⁴, Michal Gokh²⁵, Roman Krems²⁶, Florian Marquardt^{26,27}, Michał Tomza¹, Maciej Lewenstein²⁸, and Alexandre Dauphin^{29*}

¹ Faculty of Physics, University of Warsaw, Poland

² ICFO - Institut de Ciències Fotòniques, The Barcelona Institute of Science and Technology, 08860 Castelldefels (Barcelona), Spain

³ Department of Physics, University of Basel, Switzerland

⁴ Quantum Technologies Institute, University of Würzburg, Germany

⁵ Université de Paris, CNRS, Laboratoire Matière et Phénomènes Quantiques, France

⁶ Machine Learning Group, Technische Universität Berlin, Germany

⁷ BIFOLD, Berlin Institute for Foundations of Learning and Data, 10587 Berlin, Germany

⁸ Department of Applied Physics, Aalto University, Espoo, Finland

⁹ Institute of Physics, Albert-Einstein University of Freiburg, Germany

¹⁰ International Centre for Theoretical Quantum Physics, University of Gdańsk, Poland

¹¹ Department of Algorithms and System Modeling, Faculty of Electronics, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Poland

¹² Institute for Theoretical Physics, University of Innsbruck, Austria

¹³ Department of Chemistry, University of Toronto, Canada

¹⁴ Vector Institute for Artificial Intelligence, Toronto, Canada

¹⁵ Barcelona Supercomputing Center, Spain

¹⁶ IJCLab, Leiden University, The Netherlands

¹⁷ CMAP École Polytechnique, France

¹⁸ Institute of Physics, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

¹⁹ Mentor AI, Inc., Palo Alto, California, United States of America

²⁰ Beijing National Lab for Condensed Matter Physics

and Institute of Physics, Chinese Academy of Sciences, Beijing, China

²² Songshan Lake Materials Laboratory, Dongguan, China

²³ Institute for Theoretical Physics, University of Cologne, Germany

²⁴ Kavli Institute of Nanoscience, Delft University of Technology, NL-2600 GA Delft, The Netherlands

²⁵ Department of Chemistry, University of British Columbia, Vancouver, Canada

²⁶ Max Planck Institute for the Science of Light, Erlangen, Germany

²⁷ Department of Physics, Friedrich-Alexander University Erlangen-Nürnberg, Germany

²⁸ ICREA, Pg. Lluís Companys 23, 08010 Barcelona, Spain

* These authors contributed equally.

* Anna.Dawid@fuw.edu.pl, Alexandre.Dauphin@icfo.eu

June 23, 2022

Abstract

In these Lecture Notes, we provide a comprehensive introduction to the most recent advances in the application of machine learning methods in quantum sciences. We cover the use of deep learning and kernel methods in supervised, unsupervised, and reinforcement learning algorithms for phase classification, representation of many-body quantum states, quantum feedback control, and quantum circuits optimization. Moreover, we introduce and discuss more specialized topics such as differentiable programming, generative models, statistical approach to machine learning, and quantum machine learning.

Thank you!

Any questions?



This presentation has been released under the Creative Commons CC BY-NC-ND license, i.e.

BY: credit must be given to the creator.

NC: Only noncommercial uses of the work are permitted.

ND: No derivatives or adaptations of the work are permitted.