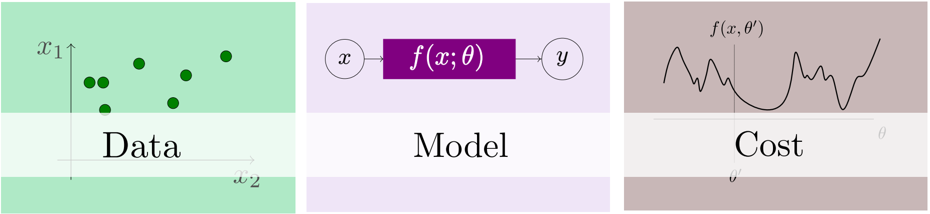
Quantum Machine Learning

Maria Schuld, Xanadu

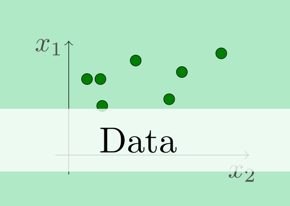
CERN seminar, 3-4th February 2021

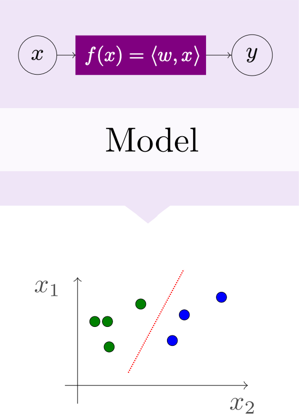
Machine Learning

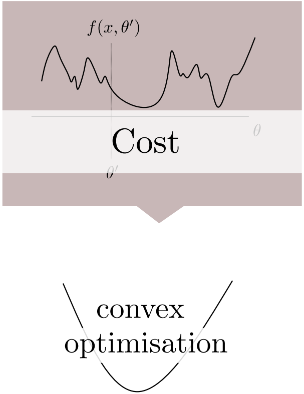


Use data samples
to construct model
that minimises cost
on unseen data.

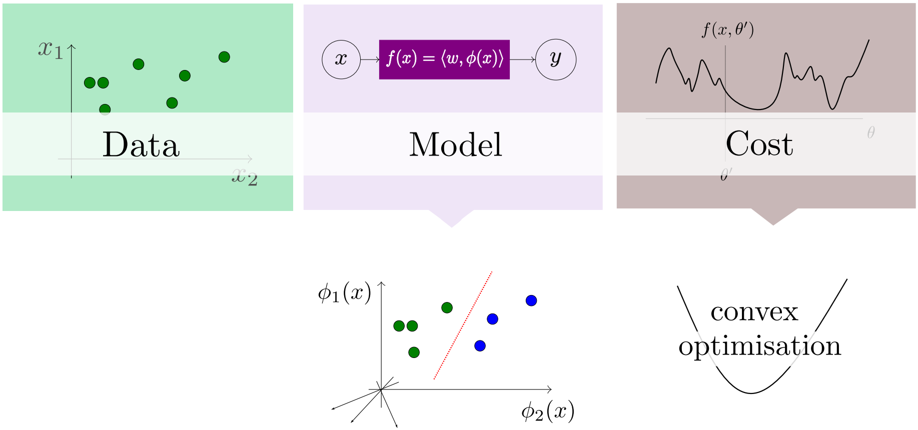
Linear models



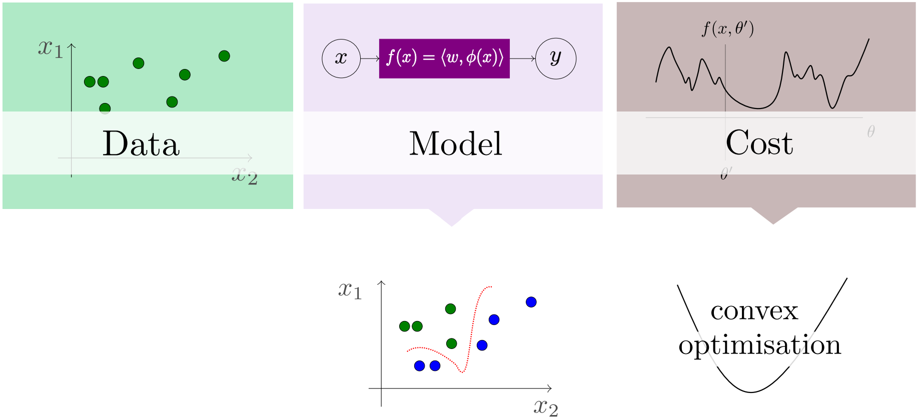




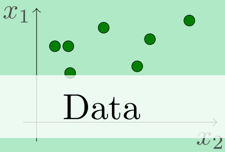
Kernel methods



Kernel methods



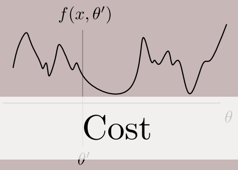
Deep learning



Data

$x \to f(x) = \cdots \phi(\langle w, x \rangle) \cdots \to y$

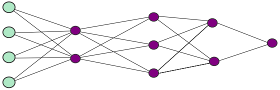
Model



Cost

Big

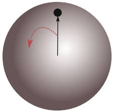
trainable,
composable
& differentiable



nonconvex optimisation

- gradient descent
- high performance hardware
- special purpose software

Quantum computing



PHYSICAL CIRCUIT

$$n \left[\begin{array}{c} |0\rangle \\ \vdots \\ |0\rangle \end{array} \right.$$

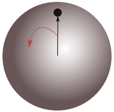
MATHEMATICAL DESCRIPTION

$|1|^2 = p(0...00)$

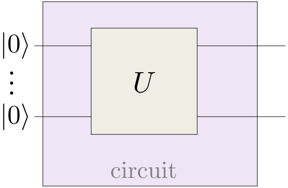
$$2^n \left[\boxed{\begin{array}{c} 1 + 0i \\ 0 + 0i \\ \vdots \end{array}} \right.$$

$|0|^2 = p(0...01)$

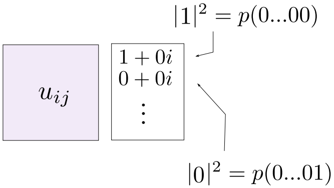
Quantum computing



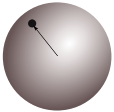
PHYSICAL CIRCUIT



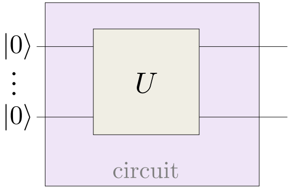
MATHEMATICAL DESCRIPTION



Quantum computing



PHYSICAL CIRCUIT



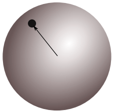
MATHEMATICAL DESCRIPTION

$$|\psi_1|^2 = p(0 \ldots 00)$$

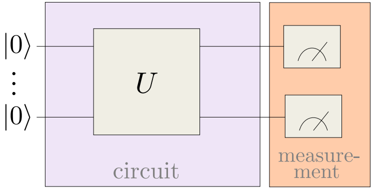
$$\begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \end{bmatrix}$$

$$|\psi_2|^2 = p(0 \ldots 01)$$

Quantum computing



PHYSICAL CIRCUIT



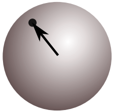
MATHEMATICAL DESCRIPTION

$$\begin{matrix} 10110\ldots1 \\ 11010\ldots0 \\ 00001\ldots0 \\ \vdots \end{matrix} \sim \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \end{pmatrix}$$

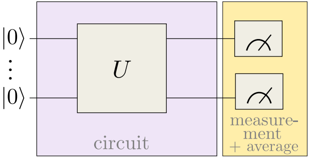
$|\psi_1|^2 = p(0\ldots00)$

$|\psi_2|^2 = p(0\ldots01)$

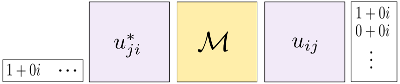
Quantum computing



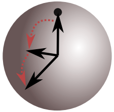
PHYSICAL CIRCUIT



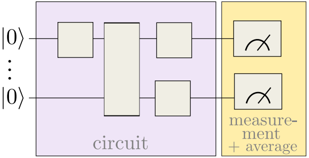
MATHEMATICAL DESCRIPTION



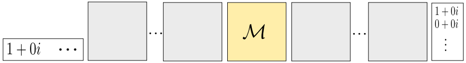
Quantum computing



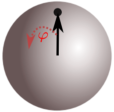
PHYSICAL CIRCUIT



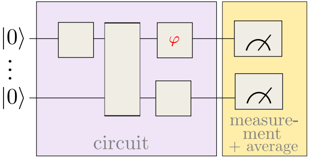
MATHEMATICAL DESCRIPTION



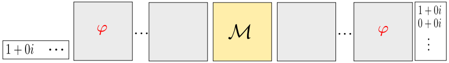
Quantum computing



PHYSICAL CIRCUIT



MATHEMATICAL DESCRIPTION



How could quantum computing help with ML?

- Data
  - speed up sampling from data distributions
  - use fewer data samples (e.g., Arunachalam 1701.06806)

How could quantum computing help with ML?

- Data
  - speed up sampling from data distributions
  - use fewer data samples (e.g., Arunachalam 1701.06806)
- Optimisation
  - speed up optimisation (Wiebe et al. 1204.5242, Rebentrost et al. 1307.0471, Denil & Freitas ~ 2012 `cs.ubc.ca/~nando/`)
  - find better solutions

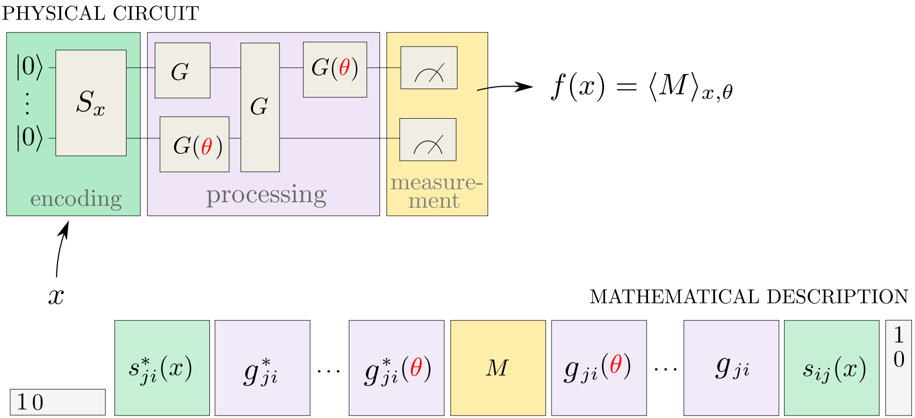
How could quantum computing help with ML?

- Data
  - speed up sampling from data distributions
  - use fewer data samples (e.g., Arunachalam 1701.06806)
- Optimisation
  - speed up optimisation (Wiebe et al. 1204.5242, Rebentrost et al. 1307.0471, Denil & Freitas ~ 2012 `cs.ubc.ca/~nando/`)
  - find better solutions
- Model
  - speed up existing models (Pararo et al. 1401.4997, Low et al. 1402.7359, Allcock et al. 1812.03089)
  - design better models (Amin et al. 1601.02036, Benedetti et al. 1906.07682)

How could quantum computing help with ML?

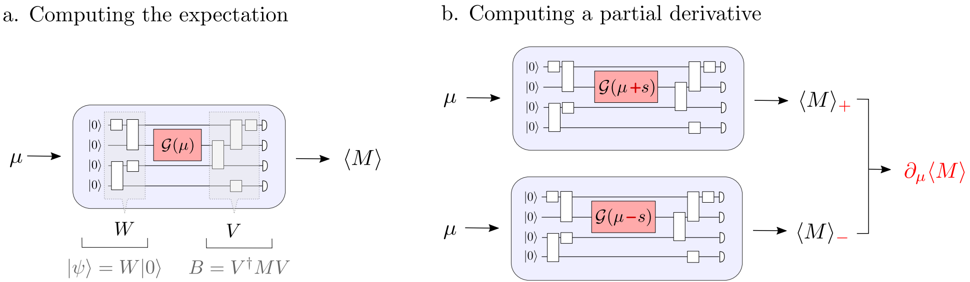
- Data
  - speed up sampling from data distributions
  - use fewer data samples (e.g., Arunachalam 1701.06806)
- Optimisation
  - speed up optimisation (Wiebe et al. 1204.5242, Rebentrost et al. 1307.0471, Denil & Freitas ~ 2012 `cs.ubc.ca/~nando/`)
  - find better solutions
- Model
  - speed up existing models (Pararo et al. 1401.4997, Low et al. 1402.7359, Allcock et al. 1812.03089)
  - design better models (Amin et al. 1601.02036, Benedetti et al. 1906.07682)
- ML for quantum experiments

Variational circuits as composable & differentiable models.



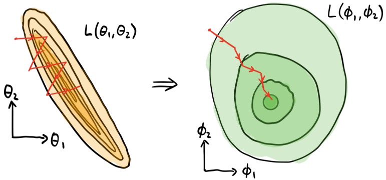
Farhi & Neven 1802.06002, Schuld et al. 1804.00633

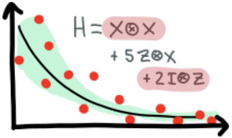
Variational circuits as composable & differentiable models.



Guerreschi & Smelyanskiy 1701.01450, Mitarai et al. 1803.00745, Schuld et al. 1811.11184

Variational circuits as composable & differentiable models.





Stokes et al. 1909.02108, Kübler et al. 1909.09083, Sweke et al. 1910.01155, Ostaszewski et al. 1905.09692, ...

Variational circuits as composable & differentiable models.

Barren plateaus in quantum neural network training landscapes

Jarrod R. McClean,[1,*] Sergio Boixo,[1,†] Vadim N. Smelyanskiy,[1,‡] Ryan Babbush,[1] and Hartmut Neven[1]
[1]*Google Inc., 340 Main Street, Venice, CA 90291, USA*
(Dated: March 30, 2018)

Many experimental proposals for noisy intermediate scale quantum devices involve training a parameterized quantum circuit with a classical optimization loop. Such hybrid quantum-classical algorithms are popular for applications in quantum simulation, optimization, and machine learning. Due to its simplicity and hardware efficiency, random circuits are often proposed as initial guesses for exploring the space of quantum states. We show that the exponential dimension of Hilbert space and the gradient estimation complexity make this choice unsuitable for hybrid quantum-classical algorithms run on more than a few qubits. Specifically, we show that for a wide class of reasonable parameterized quantum circuits, the probability that the gradient along any reasonable direction is non-zero to some fixed precision is exponentially small as a function of the number of qubits. We argue that this is related to the 2-design characteristic of random circuits, and that solutions to this problem must be studied.

Rapid developments in quantum hardware have motivated advances in algorithms to run in the so-called noisy intermediate scale quantum (NISQ) regime [1]. Many of the most promising application-oriented approaches are hybrid quantum-classical algorithms that rely on optimization of a parameterized quantum circuit [2–8]. The resilience of these approaches to certain types of errors and high flexibility with respect to coherence time and gate requirements make them especially attractive for NISQ implementations [3, 9–11].

The first implementation of such algorithms was de-

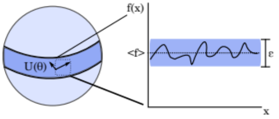
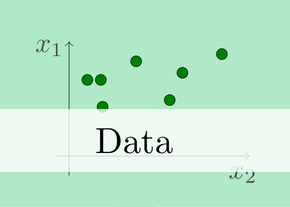
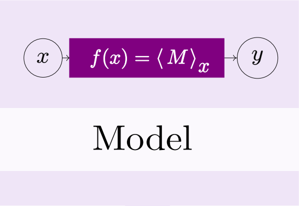


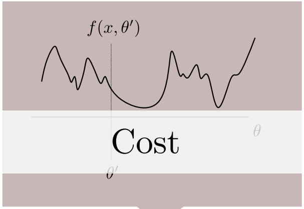
FIG. 1. A cartoon of the general geometric results from this work. The sphere depicts the phenomenon of concentration of

McClean et al. 1803.11173

We can train quantum circuits like neural nets.

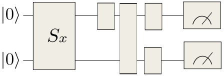






Big

trainable,
composable
& differentiable



- gradient descent
- high performance hardware
- special purpose software

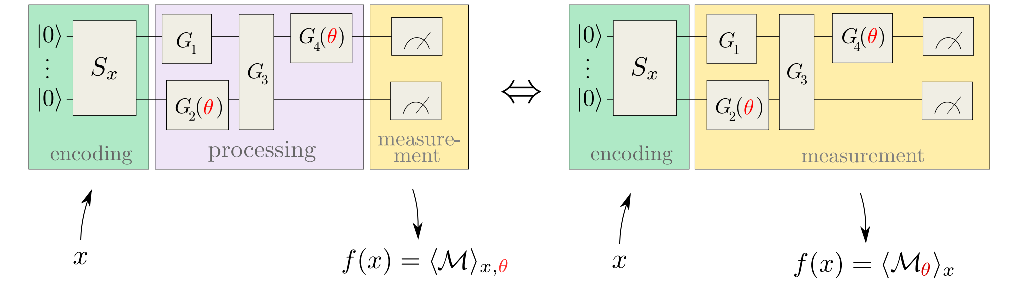
We can train quantum circuits like neural nets.

```python
import torch
from torch.autograd import Variable

data = torch.tensor([(0., 0.), (0.1, 0.1), (0.2, 0.2)])


def model(phi, x=None):
    return x*phi


def loss(a, b):
    return torch.abs(a - b) ** 2

def av_loss(phi):
    c = 0
    for x, y in data:
        c += loss(model(phi, x=x), y)
    return c

phi_ = Variable(torch.tensor(0.1), requires_grad=True)
opt = torch.optim.Adam([phi_], lr=0.02)

for i in range(5):
    l = av_loss(phi_)
    l.backward()
    opt.step()
```

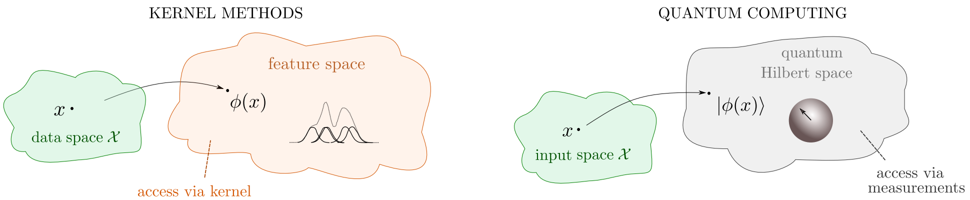
```python
from pennylane import *
import torch
from torch.autograd import Variable

data = [(0., 0.), (0.1, 0.1), (0.2, 0.2)]

dev = device('default.qubit', wires=2)

@qnode(dev, interface='torch')
def circuit(phi, x=None):
    templates.AngleEmbedding(features=[x], wires=[0])
    templates.BasicEntanglerLayers(weights=phi, wires=[0, 1])
    return expval(PauliZ(wires=[1]))

def loss(a, b):
    return torch.abs(a - b) ** 2

def av_loss(phi):
    c = 0
    for x, y in data:
        c += loss(circuit(phi, x=x), y)
    return c

phi_ = Variable(torch.tensor([[0.1, 0.2],[-0.5, 0.1]]), requires_grad=True)
opt = torch.optim.Adam([phi_], lr=0.02)

for i in range(5):
    l = av_loss(phi_)
    l.backward()
    opt.step()
```

pennylane.ai

Quantum models are (similar to) kernel methods.



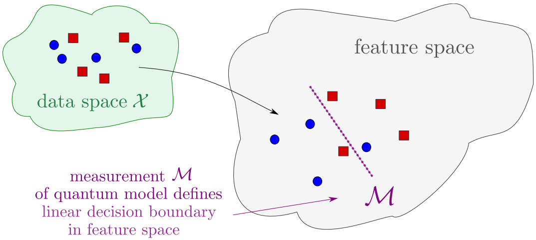
Quantum models are (similar to) kernel methods.



Schuld & Killoran 1803.07128, Havlicek et al. 1804.11326, Schuld 2101.11020

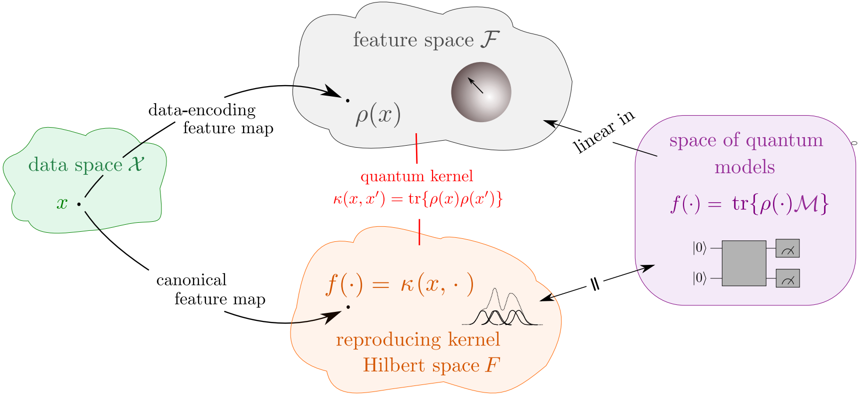
Quantum models are (similar to) kernel methods.

$$f(x) \quad = \quad \langle M_\theta \rangle_x \quad = \quad \mathrm{tr}\{\rho(x) M_\theta\}$$



Schuld 2101.11020

Quantum models are (similar to) kernel methods.



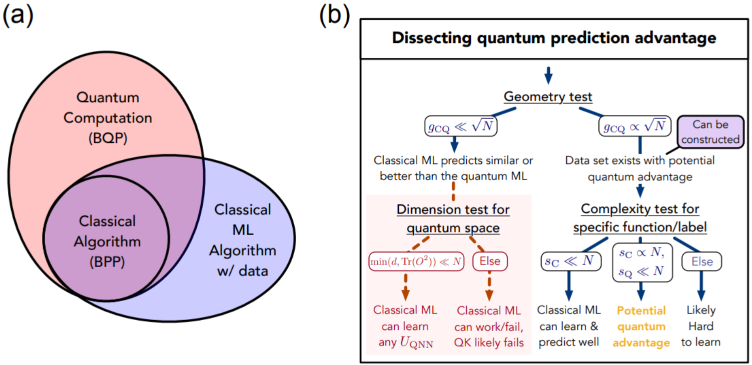
Schuld 2101.11020

Quantum models are (similar to) kernel methods.

- In many situations we can replace a variational QML model by an SVM with a “quantum kernel”.
- We are guaranteed that the best measurement can be found in linear time in the number of training data.
- Compared to current variational training, we need fewer (and shorter) circuit evaluations.

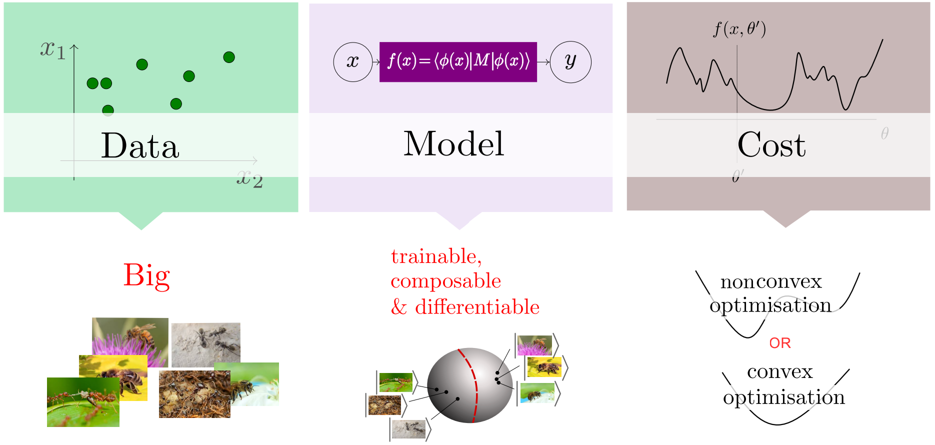
Schuld 2101.11020

Quantum models are (similar to) kernel methods.

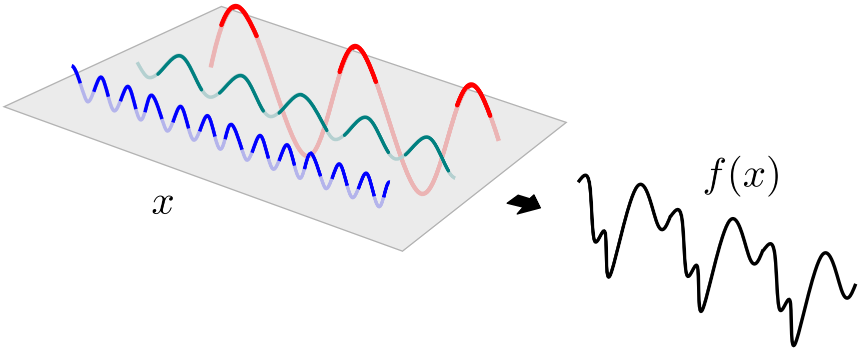


Huang et al. 2011.01938, also: Liu et al. 2010.02174

Quantum models are (similar to) kernel methods.



Quantum models have a Fourier representation.



Javier Vidal & Dirk Theis 1901.11434; Schuld, Sweke & Meyer 2008.08605

Thoughts on QML & HEP.

FERMILAB-PUB-20-184-QIS

2v1 [quant-ph] 18 May 2020

Quantum Machine Learning in High Energy Physics

Wen Guan, Gabriel Perdue, Arthur Pesah, Maria Schuld, Koji Terashi, Sofia Vallecorsa, Jean-Roch Vlimant

E-mail: jvlimant@caltech.edu

May 2020

Abstract. Machine learning has been used in high energy physics since a long time, primarily at the analysis level with supervised classification. Quantum computing was postulated in the early 1980s as way to perform computations that would not be tractable with a classical computer. With the advent of noisy intermediate-scale quantum computing devices, more quantum algorithms are being developed with the aim at exploiting the capacity of the hardware for machine learning applications. An interesting question is whether there are ways to combine quantum machine learning with High Energy Physics. This paper reviews the first generation of ideas that use quantum machine learning on problems in high energy physics and provide an outlook on future applications.

Guan, Perdue, Pesah, Schuld, Terashi, Vallecorsa, Vlimant, *Quantum Machine Learning in High Energy Physics*, arxiv:2005.08582

Thoughts on QML & HEP.

- Can we use specific properties of QML to study HEP data?
- Learning with "quantum data".
- Let $S(x)$ be an experimental protocol that prepares a quantum system in state $\rho(x)$. Can we distinguish between different classes of $\rho(x)$?

Thank you.

www.pennylane.ai
www.xanadu.ai
@XanaduAI