Title: Why supervised learning with quantum circuits reduces to kernel methods

Speakers: Maria Schuld

Series: Perimeter Institute Quantum Discussions

Date: May 19, 2021 - 10:00 AM

URL: http://pirsa.org/21050018

Abstract: With the race for quantum computers in full swing, researchers became interested in the question of what happens if we replace a supervised machine learning model with a quantum circuit. While such "supervised quantum models" are sometimes called "quantum neural networks", their mathematical structure reveals that they are in fact kernel methods with kernels that measure the distance between data embedded into quantum states. This talk gives an informal overview of the link, and discusses the far-reaching consequences for quantum machine learning.

Why supervised learning with quantum circuits reduces to kernel methods

Maria Schuld, Xanadu

Perimeter Quantum Information Seminar, April 2021

XANADU

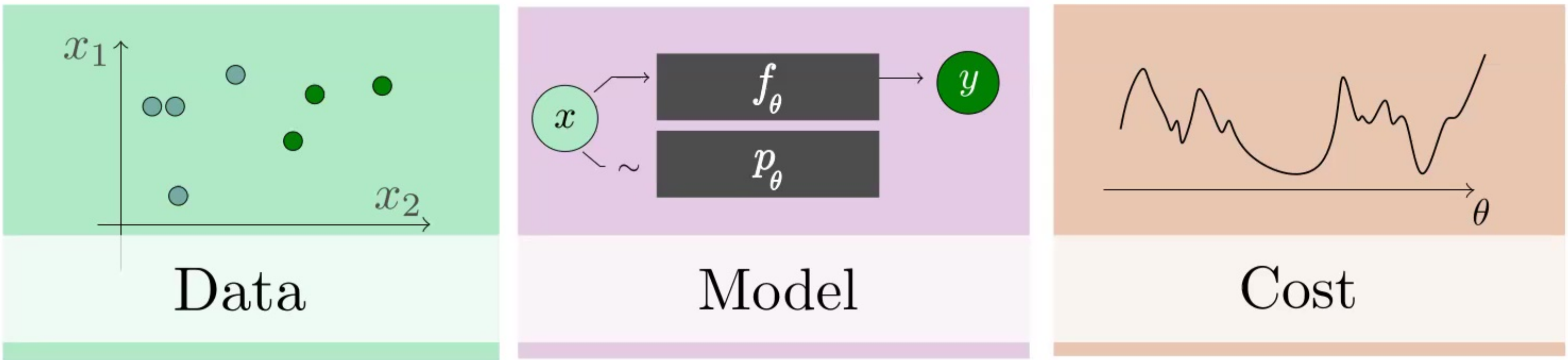
Summary of the talk

- In supervised machine learning, we train models $f(x)$ with data so that the model captures the relation between inputs x and target outputs y.
- A popular approach in quantum machine learning is to use variational circuits (that somehow encode inputs x) as models $f(x)$.
- These "circuit learners" are linear models $f(x) = \langle w, \phi(x) \rangle_{\mathcal{F}}$ in some high-dimensional feature space $\mathcal{F}$.
- Machine learning knows a lot about such models, which can help us immensely to analyse circuit learners.
- For example, it tells us that variational training is usually a bad idea, and that the power of quantum computing has to lie in the data encoding strategy.

Further reading.

- Rebentrost, Mohseni and Lloyd (2013) Quantum support vector machine for big data classification, arXiv:1307.0471.
- Chatterjee, Yu (2013) Generalized Coherent States, Reproducing Kernels, and Quantum Support Vector Machines, arXiv:1612.03713.
- Schuld, Killoran (2018) Quantum machine learning in feature Hilbert spaces, arXiv:1803.07128.
- Havlicek et al. (2018) Supervised learning with quantum enhanced feature spaces, arXiv:1804.11326.
- Liu, Arunachalam, Temme (2020) A rigorous and robust quantum speed-up in supervised machine learning, arXiv:2010.02174.
- Huang et al. (2020) Power of data in quantum machine learning, arXiv:2011.01938.
- Schuld (2021) Quantum machine learning models are kernel methods, arXiv:2101.11020.

The problem of supervised learning.



Find the model
that minimises the cost
on all data.

→

Find the model
that minimises the cost
on a training set.

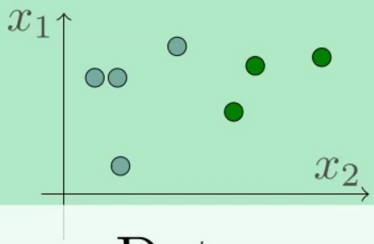
The problem of supervised learning.

Given some data samples $(x^1, y^1)...(x^M, y^M)$ of inputs $x \in \mathcal{X}$ and their labels $y \in \mathcal{Y}$, a model family $f_\theta(x)$ and a loss $L(f(x), y)$, find the parameters θ that solve

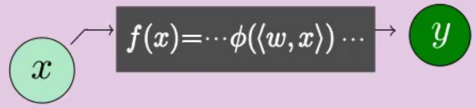
$$\min_\theta \frac{1}{M} \sum_{m=1}^{M} L(f_\theta(x), y) + \lambda g(f_\theta).$$

This is called *(regularised) empirical risk minimisation.*

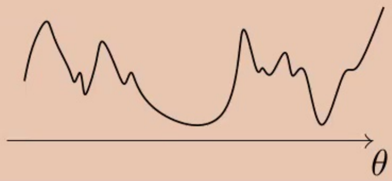
The problem of supervised learning.



Data



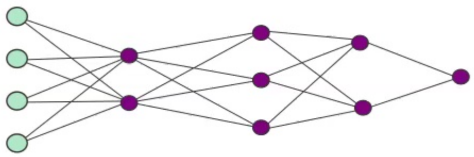
Model



Cost

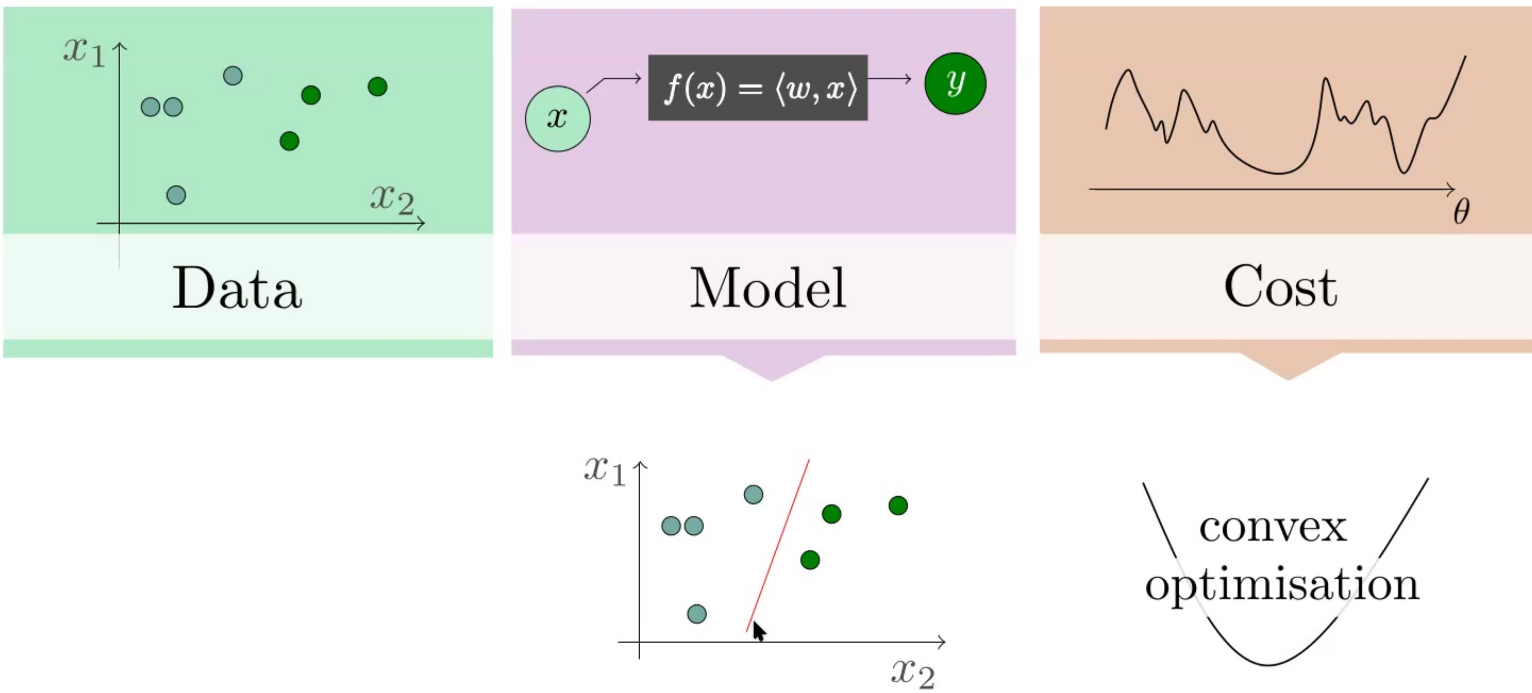
Big

composable
& differentiable

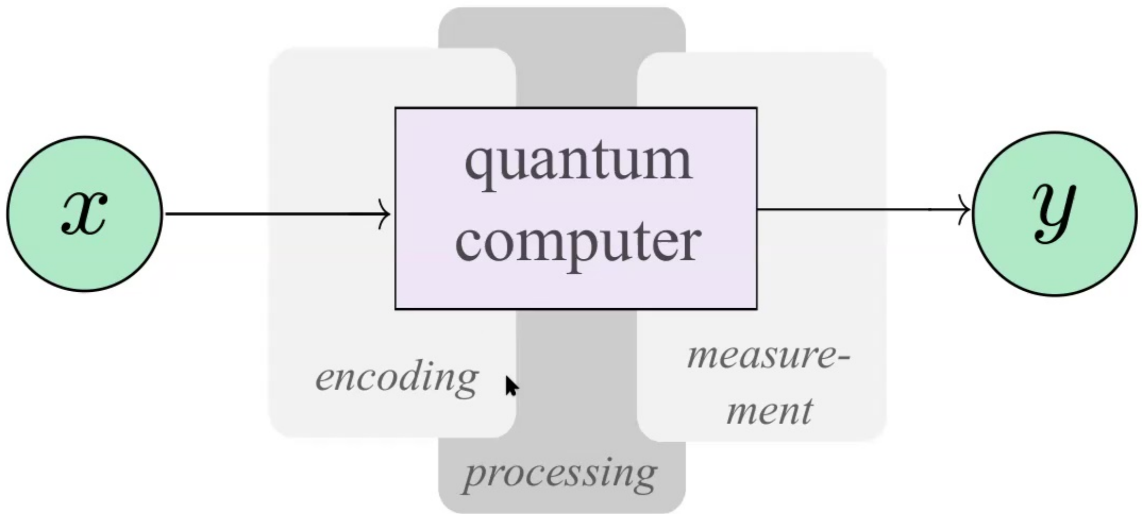


- gradient descent
- high performance hardware
- special purpose software

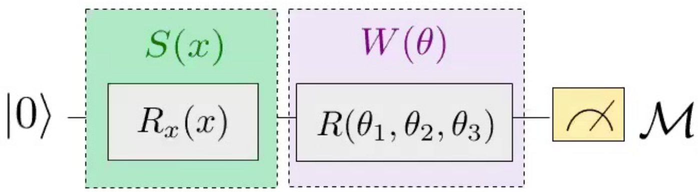
The problem of supervised learning.

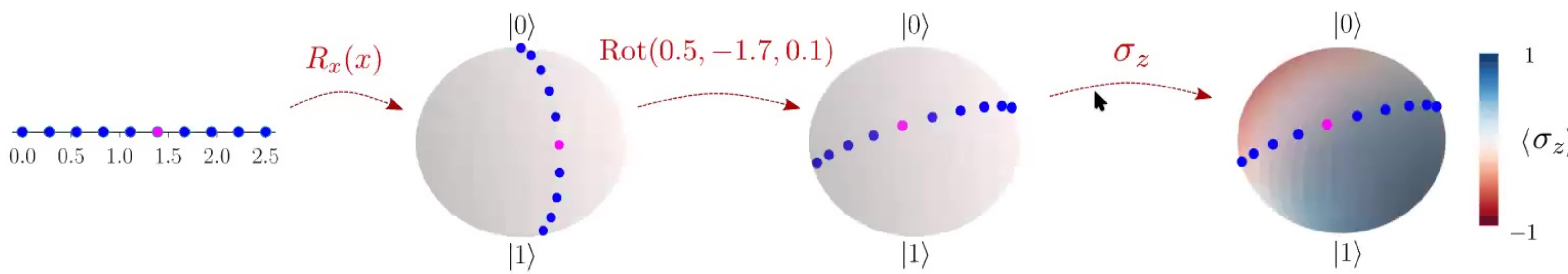


We can use quantum circuits as models.



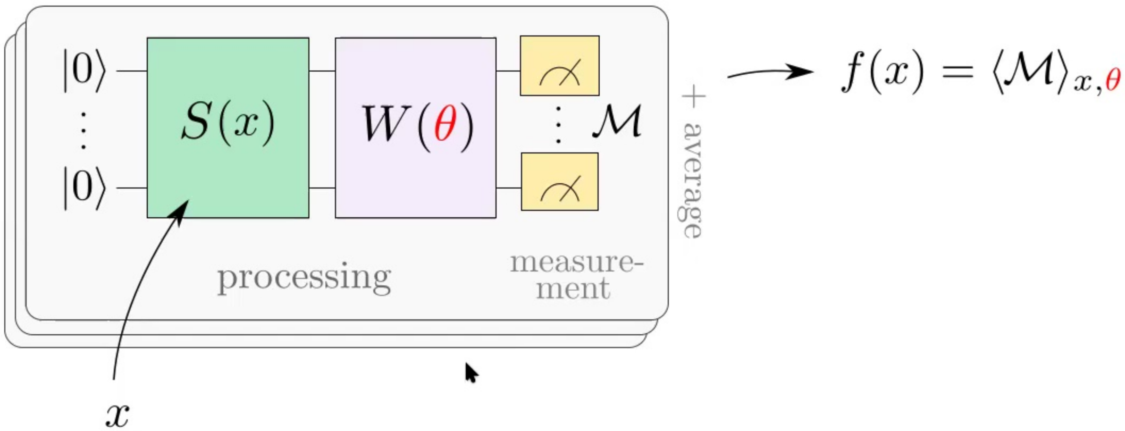
We can use quantum circuits as models.





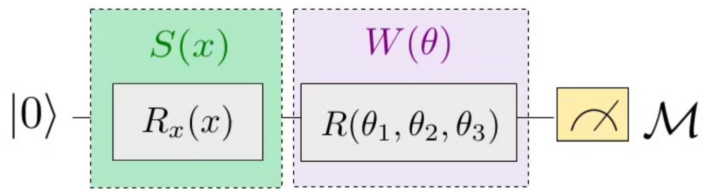
$$f_\theta(x) = \cos(\theta_2)\cos(x) - \sin(\theta_1)\sin(\theta_2)\sin(x)$$

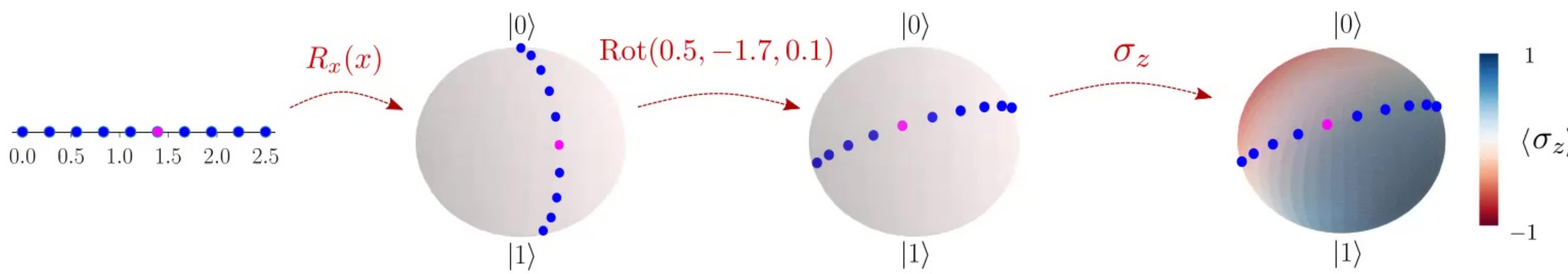
We can use quantum circuits as models.



5 / 12

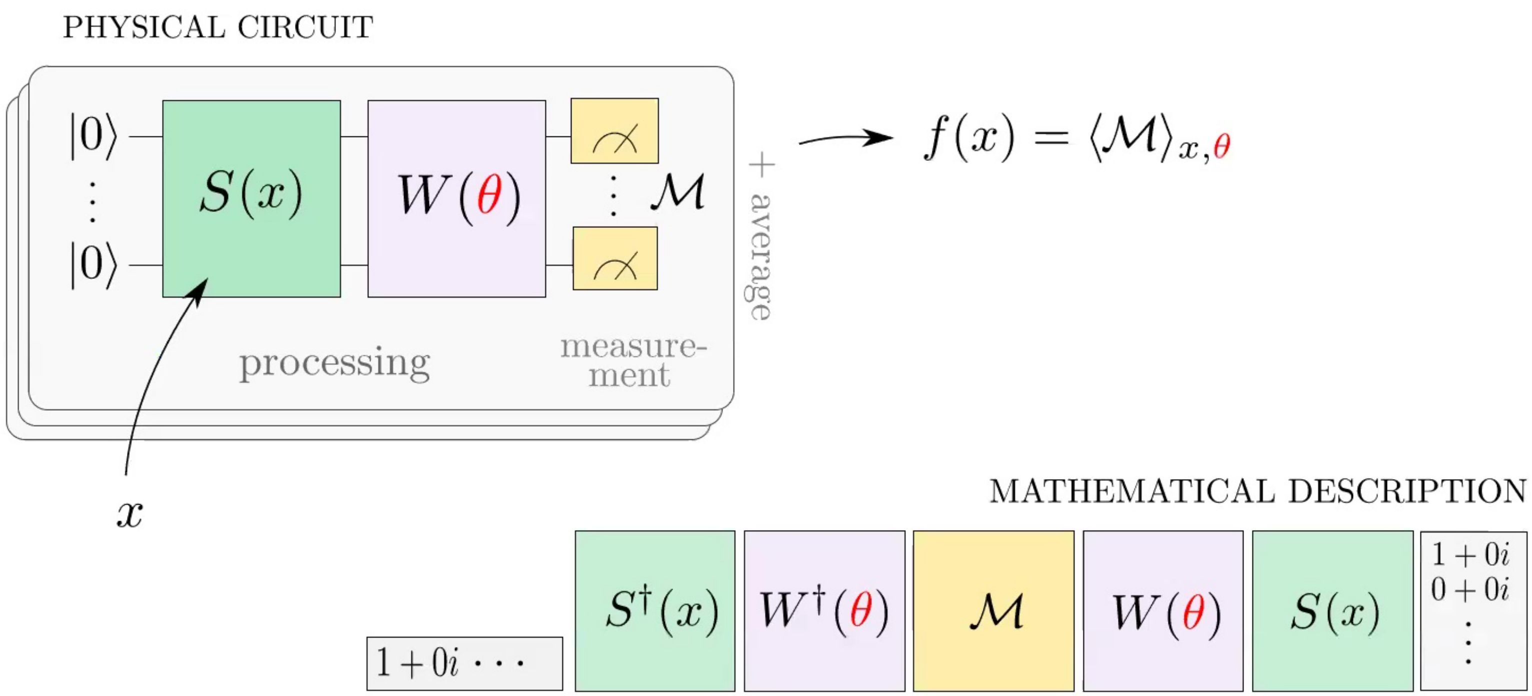
We can use quantum circuits as models.



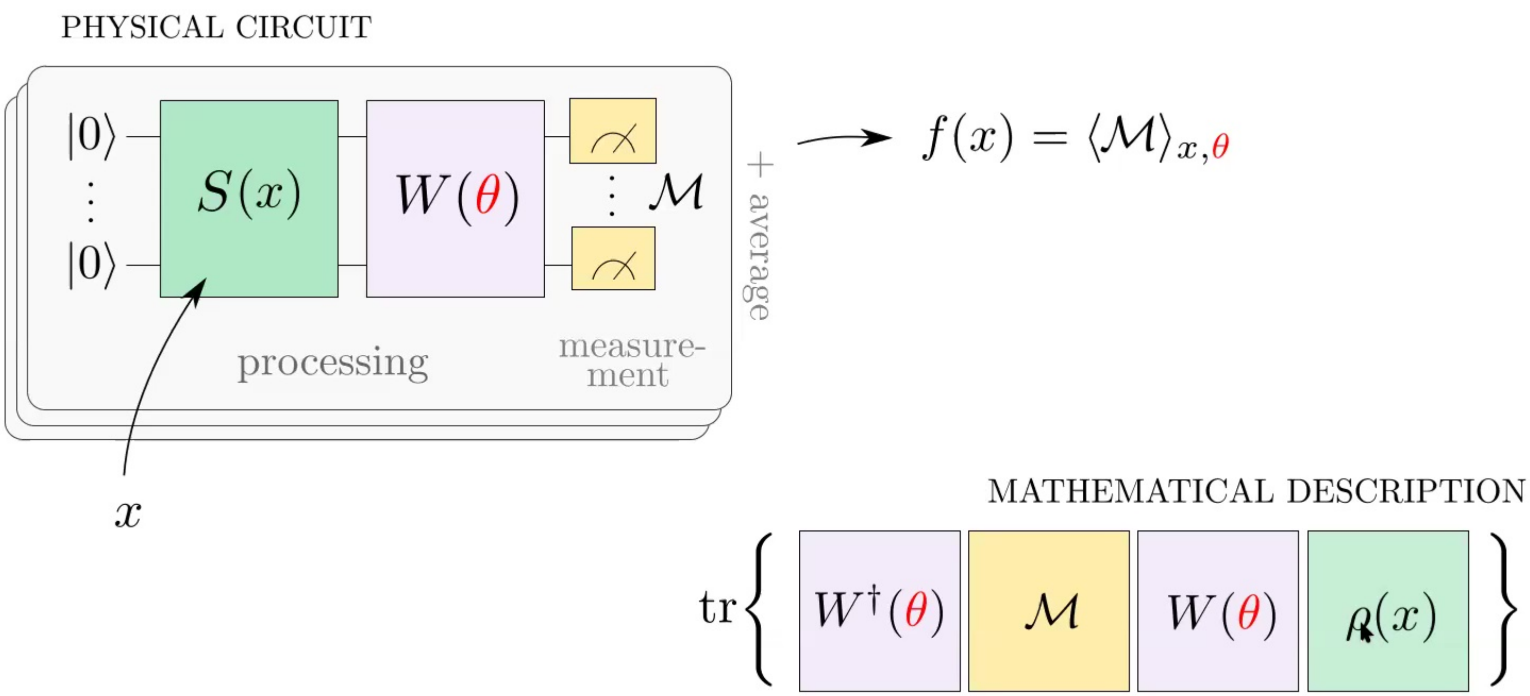


$$f_\theta(x) = \cos(\theta_2)\cos(x) - \sin(\theta_1)\sin(\theta_2)\sin(x)$$

These circuit learners are linear models.

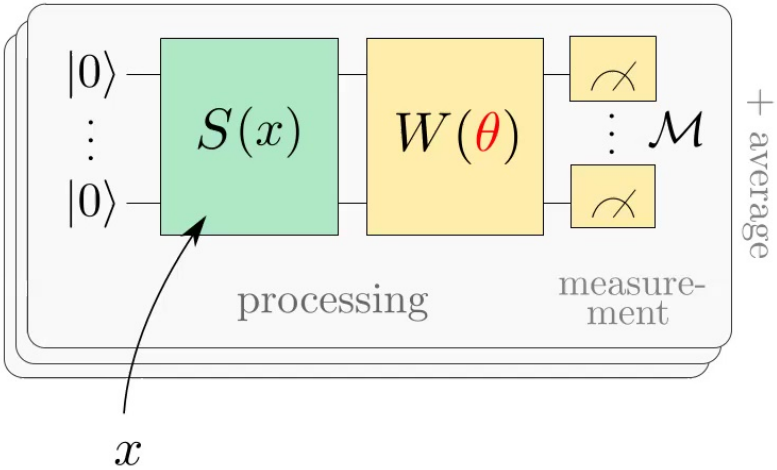


These circuit learners are linear models.



These circuit learners are linear models.

PHYSICAL CIRCUIT



$$f(x) = \langle \mathcal{M}(\theta) \rangle_x$$

MATHEMATICAL DESCRIPTION

$$\mathrm{tr}\left\{ \mathcal{M}(\theta)\, \rho(x) \right\}$$

These circuit learners are linear models.

$$f(x) = \mathrm{tr}\{\mathcal{M}(\theta)\ \rho(\mathrm{x})\} = \langle \mathrm{w}, \phi(\mathrm{x}) \rangle_{\mathcal{F}}$$

Note: circuit learners are linear in $\mathcal{M}$, *not* in θ.

Entering the world of kernel theory.

- SVM (general definition, i.e., Christmann & Steinwart): $f(x) = \langle w, \phi(x) \rangle_{\mathcal{F}}$.
- Computations in feature space through kernel $\kappa(x, x') = \langle \phi(x'), \phi(x) \rangle_{\mathcal{F}}$.
- *Representer theorem:* Model that minimises regularised empirical risk can be written as

$$f_\alpha(x) = \sum_{m=1}^{M} \alpha_m \kappa(x, x^m).$$

- We can find this model by solving the problem

$$\min_\alpha \frac{1}{M} \sum_{m=1}^{M} L(y^m, f_\alpha(x^m)) + \sum_{i,j=1}^{M} \alpha_i \alpha_j \kappa(x^i, x^j).$$

Supervised circuit learners are support vector machines.

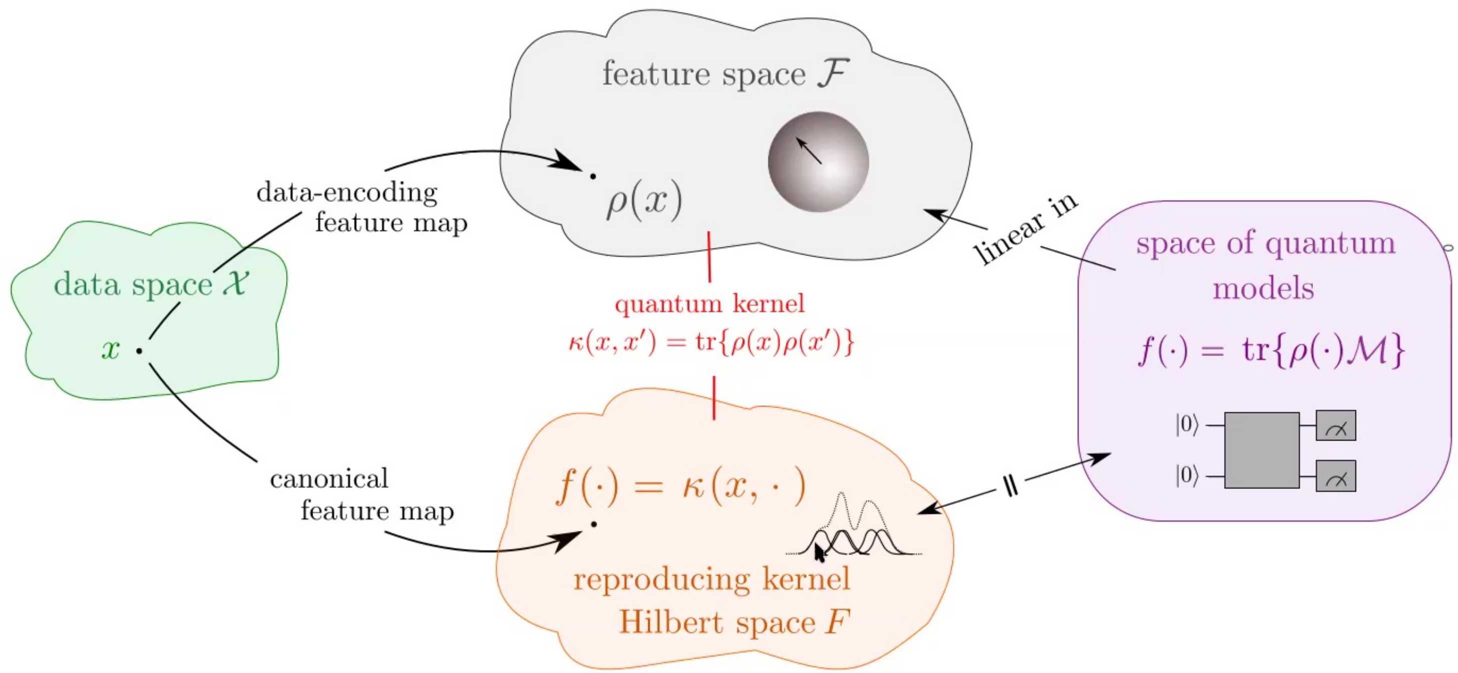
- Quantum model: $f(x) = \langle M_\theta \rangle_x = \mathrm{tr}\{\rho(x)\mathcal{M}(\theta)\}$.
- Computations in feature space through kernel $\kappa(x, x') = \mathrm{tr}\{\rho(x)\rho(x')\}$.
- *Representer theorem:* Model that minimises regularised empirical risk can be written as

$$f_\alpha(x) = \sum_{m=1}^{M} \alpha_m \mathrm{tr}\{\rho(x')\rho(x)\}.$$

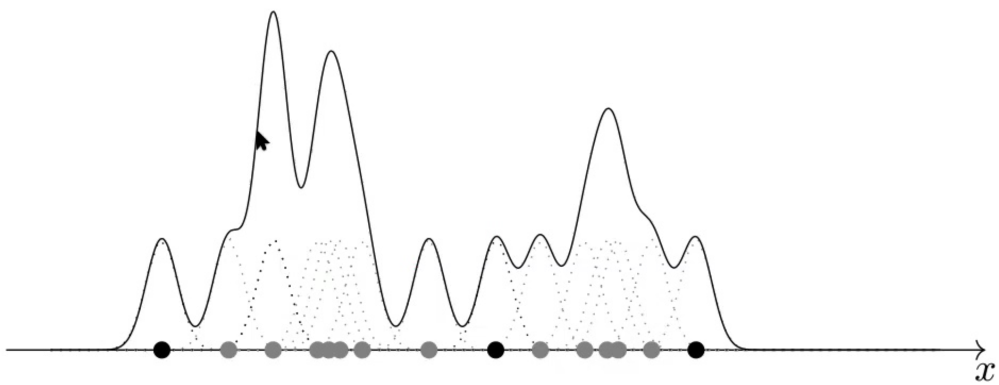
- We can find this model by solving the problem

$$\min_\alpha \frac{1}{M} \sum_{m=1}^{M} L(y^m, f_\alpha(x^m)) + \sum_{i,j=1}^{M} \alpha_i \alpha_j \mathrm{tr}\{\rho(x^i)\rho(x^j)\}.$$

The proofs use Reproducing Kernel Hilbert Spaces.

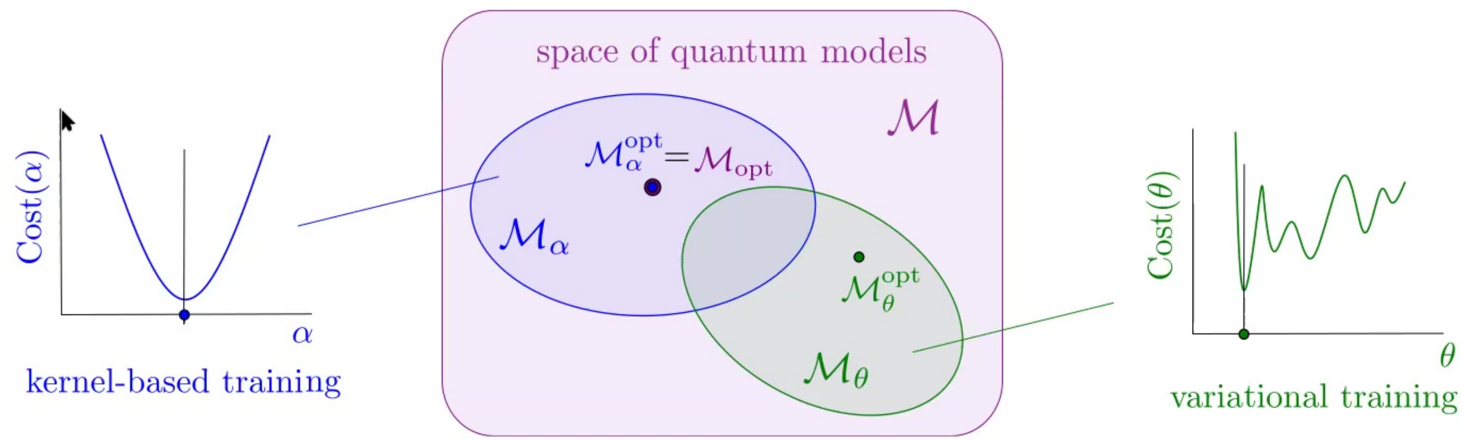


The proofs use Reproducing Kernel Hilbert Spaces.

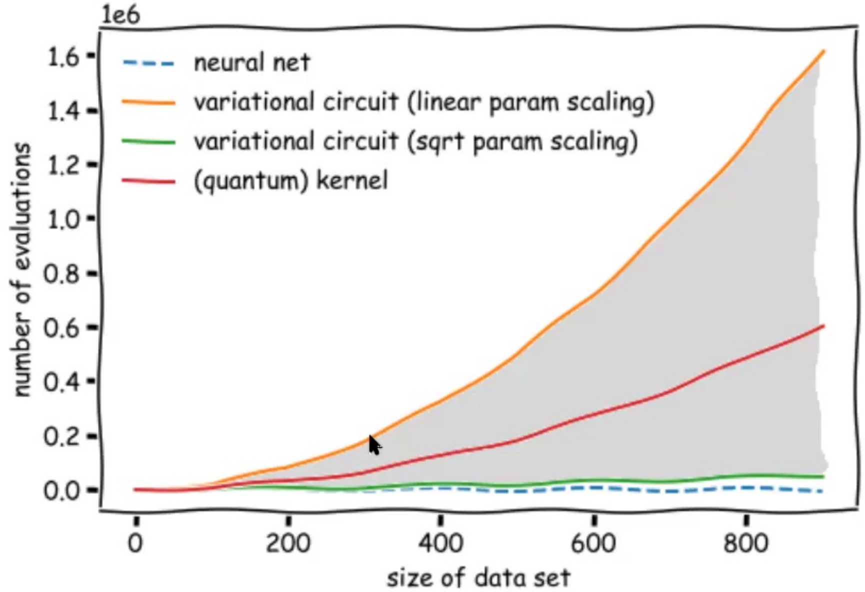


What does this mean for variational training?

```
svm = SVC(kernel=kernel_matrix).fit(X_train, y_train)
svm.predict(X_test)
```



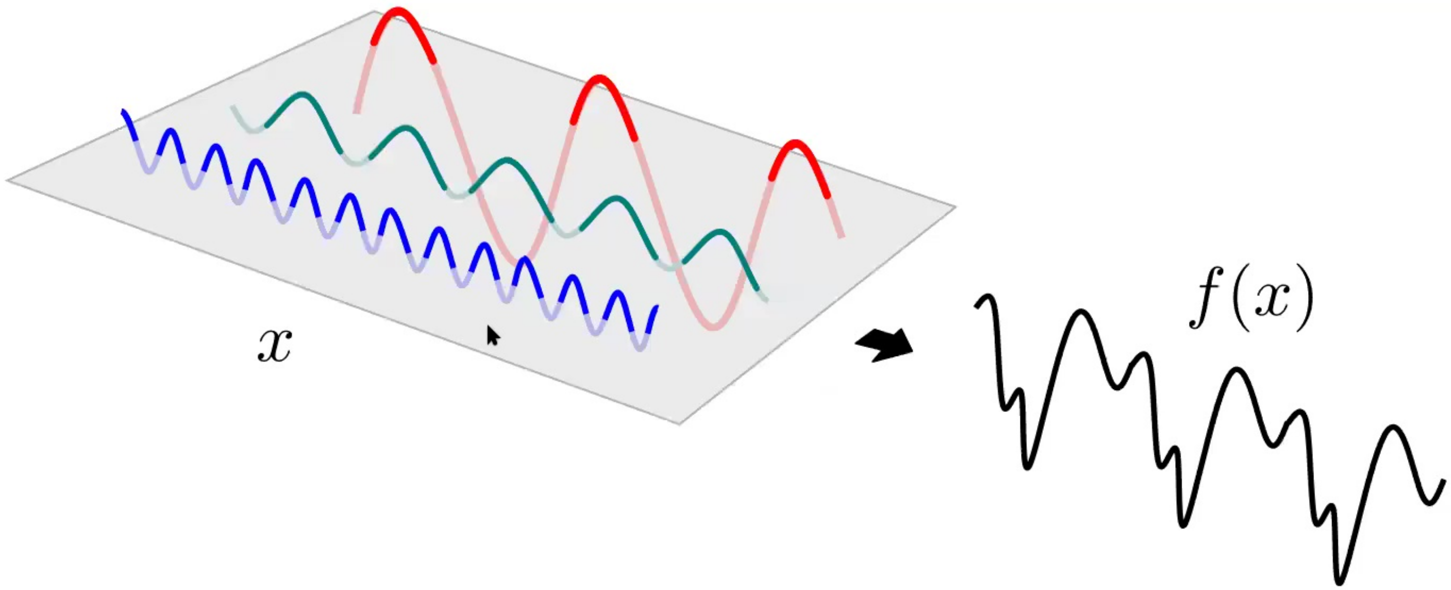
What does this mean for variational training?



Kernel methods: $\mathcal{O}(n_{\text{data}}^2)$, Var circuits: $\mathcal{O}(n_{\text{data}} n_{\text{params}})$, NNs: $\mathcal{O}(n_{\text{data}})$

https://pennylane.ai/qml/demos/tutorial_kernel_based_training.html

How do quantum kernels look like?



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

How do quantum kernels look like?

Let $\mathcal{X} = \mathbb{R}^N$ and $S(\mathbf{x})$ be a quantum circuit that encodes the data inputs $\mathbf{x} = (x_1, \ldots, x_N) \in \mathcal{X}$ into a n-qubit quantum state $S(\mathbf{x})|0\rangle = |\phi(\mathbf{x})\rangle$ via gates of the form e^{-ix_iG} for $i = 1, \ldots, N$, where wlog G is a $d \leq 2^n$-dimensional diagonal operator with spectrum $\lambda_1, \ldots, \lambda_d$. Between such data-encoding gates, and before and after the entire encoding circuit, arbitrary unitary evolutions $W^{(1)}, \ldots, W^{(N+1)}$ can be applied. The quantum kernel $\kappa(\mathbf{x}, \mathbf{x}')$ can be written as

$$\kappa(\mathbf{x}, \mathbf{x}') = \sum_{\mathbf{s},\mathbf{t} \in \Omega} e^{i\mathbf{s}\mathbf{x}} e^{i\mathbf{t}\mathbf{x}'} c_{\mathbf{st}},$$

where $\Omega \subseteq \mathbb{R}^N$, and $c_{\mathbf{st}} \in \mathbb{C}$. For every $\mathbf{s}, \mathbf{t} \in \Omega$ we have $-\mathbf{s}, -\mathbf{t} \in \Omega$ and $c_{\mathbf{st}} = c^*_{-\mathbf{s}-\mathbf{t}}$, which guarantees that the quantum kernel is real-valued.

Thank you!

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