Innovating machine learning with near-term quantum computing

Maria Schuld
University of KwaZulu-Natal and Xanadu

IPAM Workshop @ UCLA, November 2019
Agenda

- Motivation
- Mathematics of QML
- Variational quantum models
- Links to neural nets
- Links to kernel methods
- Open problems
MOTIVATION
Machine learning is important for the “quantum revolution”. 

1990: Hidden subgroup
- Shor
- QFT
- Adiabatic optimization
- Quantum Walks
- HHL/ Matrix inversion
- Grover
- Group isomorphism
- Quantum Simulation
- Deutsch-Josza

2019: QUANTUM SOFTWARE

QUANTUM HARDWARE
Machine learning is important for the “quantum revolution”.
Should AI researchers care about quantum computing?
How can quantum computers improve machine learning?

**MOTIVATION**

- speed
- robustness
- #data
- output
The “first wave of QML” wanted to speed up ML.

<table>
<thead>
<tr>
<th>Problems</th>
<th>QIP tools</th>
<th>Applied to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulating linear algebra calculus with qubits</td>
<td>matrix inversion, inner products, eigenvalue decomposition, singular value decomposition</td>
<td>quantum phase estimation, postselective amplitude update, Hamiltonian simulation, density matrix exponentiation</td>
</tr>
<tr>
<td>Optimisation with Grover search</td>
<td>finding closest neighbours, Markov chains</td>
<td>amplitude amplification, quantum walks</td>
</tr>
<tr>
<td>Sampling from quantum states</td>
<td>sampling from model distribution</td>
<td>quantum annealing, quantum rejection sampling</td>
</tr>
<tr>
<td>Optimisation with ground states of Hamiltonians</td>
<td>combinatorial optimisation</td>
<td>adiabatic quantum computing, quantum annealing, quantum simulation</td>
</tr>
</tbody>
</table>
The “second wave of qml” trains quantum computations.
THE MATHEMATICS OF QML
Quantum theory is a mathematical framework invented in the 1930s. The key developments include:

- **1900**: Max Planck discovers quantisation of black-body radiation.
- **1905**: Albert Einstein discovers quantisation of light.
- **1912**: Max Born applies quantisation to atomic spectrum.
- **1923**: Erwin Schrödinger formulates wave mechanics.
- **1925**: Heisenberg, Jordan, and Born formulate matrix mechanics.
- **1926**: de Broglie postulates the duality of waves and particles.
- **1929**: Paul Dirac and John von Neumann unite approaches & formulate modern quantum theory.

Quantum theory branches out into physical subdisciplines.
Quantum theory calculates the expectations of measurements.

- A quantum state $|\psi\rangle$ lives in a **Hilbert space** $\mathcal{H}$ with scalar product $\langle \psi | \psi \rangle$.
- An **observable** is represented by a Hermitian operator $O$ on $\mathcal{H}$. The eigenvectors of $O$ form an orthonormal basis of $\mathcal{H}$ with real eigenvalues. Every $|\psi\rangle \in \mathbb{C}^N$ can hence be expressed in $O$’s eigenbasis $\{|\psi_i\rangle\}_{i=1}^N$, $|\psi\rangle = \sum_{i=1}^N a_i |\psi_i\rangle$, where the $a_i \in \mathbb{C}$ are the **amplitudes**.
- The effect of applying $O$ to an element $|\psi\rangle \in \mathbb{C}^N$ is fully defined by the eigenvalue equations $O|\psi_i\rangle = \lambda_i |\psi_i\rangle$ with eigenvalues $\lambda_i$. **Expectation values** of the observable property are calculated by $\mathbb{E}(O) = \langle \psi | O |\psi\rangle$.
- The dynamic evolution of a quantum state is represented by a **unitary operator** $U = U(t_2, t_1)$ mapping $|\psi(t_1)\rangle$ to $U(t_2, t_1)|\psi(t_1)\rangle = |\psi(t_2)\rangle$ with $U^\dagger U = 1$. $U$ is the solution of the corresponding **Schrödinger equation** $i\hbar \partial_t |\psi\rangle = H|\psi\rangle$ with **Hamiltonian** $H$. 

**THE MATHEMATICS OF QML**
From probabilities to amplitudes.

Consider a set of $N$ measurement outcomes $X = \{x_1, ..., x_N\}$ occurring with probability $p_1, ..., p_N$. The expectation of the measurement is given by:

$$\langle X \rangle = \sum_{i=1}^{N} p_i x_i = \bar{p}^T \bar{x}$$
From probabilities to amplitudes.

Consider

\[ \vec{q} = \begin{pmatrix} \sqrt{p_1} \\ \vdots \\ \sqrt{p_N} \end{pmatrix} = \sqrt{p_1} \begin{pmatrix} 1 \\ \vdots \\ 0 \end{pmatrix} + \cdots + \sqrt{p_N} \begin{pmatrix} 0 \\ \vdots \\ 1 \end{pmatrix}, \quad X = \begin{pmatrix} x_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & x_N \end{pmatrix}. \]

The expectation value can now be written as

\[ \langle X \rangle = \vec{q}^T X \vec{q} = \sum_{i=1}^{N} p_i x_i. \]
From probabilities to amplitudes.

Replace $q$ with a complex amplitude vector $\psi = (\alpha_1, ..., \alpha_N)^T \in \mathbb{C}^N$.

Replace $X$ by a complex, self-adjoint matrix $O \in \mathbb{C}^{N \times N}$.

The eigenvalues $\sigma_i$ of $O$ correspond to the outcomes of measurements.
Unitary transformations describe evolutions.

Time evolutions of a quantum system are described by unitary transformations of the amplitude vector,

\[
\begin{pmatrix}
  u_{11} & \cdots & u_{1N} \\
  \vdots & \ddots & \vdots \\
  u_{N1} & \cdots & u_{NN}
\end{pmatrix}
\begin{pmatrix}
  \alpha_1 \\
  \vdots \\
  \alpha_N
\end{pmatrix}
= \begin{pmatrix}
  \alpha'_1 \\
  \vdots \\
  \alpha'_N
\end{pmatrix}, \quad \sum_{i=1}^{N} |\alpha_i|^2 = \sum_{i=1}^{N} |\alpha'_i|^2 = 1.
\]
Quantum computing is a special case.

Quantum system $\rightarrow$ $n$ qubits

Measurement outcomes $\rightarrow$ $X = \{00...0, ..., 11...1\}$.

States $\rightarrow$ $\psi = (\alpha_1, ..., \alpha_2^n)^T$

Evolution $\rightarrow$ $U = G_L, ..., G_1$

Expectation of measuring first qubit $\rightarrow$ $\psi^\dagger (\sigma_z \otimes \mathbb{1}^{(n-1)}) \psi$
VAR. QUANTUM MODELS
Variational models are trainable circuits.
Variational models are trainable circuits.

$$\text{Quantum device} \xrightarrow{U_\theta} \text{Classical device}$$

$$\text{Cost}(\theta)$$

update

$$\theta^{(t)} \rightarrow \theta^{(t+1)}$$
Variational models are trainable circuits.
Variational models consist of three elements.

Farhi & Neven 1802.06002, Schuld et al. 1804.00633, Benedetti et al. 1906.07682
Variational models consist of three elements.

**CIRCUIT**

- **State prep.**
  - $|0\rangle$
  - $\ldots$
  - $|0\rangle$

- **Model circuit**
  - $S_x$
  - $U_\theta$

- **Measurement**
  - $\langle \sigma_z \rangle$

- **Post-process.**
  - $y$

**FORMAL**

- $x \rightarrow \psi(x)$
  - nonlin. map
- $\rightarrow \psi' = U_\theta \psi(x)$
  - unitary transform.
- $\rightarrow (\sigma_k^q |\psi'_k|^2)$
  - expectation
- $\rightarrow y$

Farhi & Neven 1802.06002, Schuld et al. 1804.00633, Benedetti et al. 1906.07682

VARIATIONAL QML MODELS
LINKS TO NEURAL NETS
Variational models are linear, symmetric neural networks.
Variational models are linear, symmetric neural networks.

Fig. 8.1 The single qubit gate $G_{q_2}$ (left) and the controlled single qubit gate $c_{q_3}G_{q_2}$ (right) from the examples applied to a system of 3 qubits drawn in the graphical representation of neural networks. The gates take a quantum state with amplitude...
Variational models are linear, symmetric neural networks.
We can do gradient descent on variational models.

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

LINKS TO NEURAL NETS
We can do gradient descent on variational models.

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

LINKS TO NEURAL NETS
We can do gradient descent on variational models.
We can do gradient descent on variational models.

Stochastic gradient descent.

Sweke et al 1910.01155

LINKS TO NEURAL NETS
We are investigating the expressivity of variational models.
We are investigating the expressivity of variational models.
We are investigating the expressivity of variational models.

Sim, Johnson, Aspuru-Guzik 1905.10876

LINKS TO NEURAL NETS
We are investigating the expressivity of variational models.

**Theorem 3.** The expressive power of MPQCs and TPQCs with $O(\text{poly}(N))$ single qubits gates and CNOT gates, and classical neural networks with $O(\text{poly}(N))$ trainable parameters, where $N$ refers to the number of qubits or the visible units, can be ordered as: MPQCs $>$ DBM $>$ long range RBM $>$ TPQCs $>$ short range RBM.
LINKS TO KERNELS
QC and kernel methods share the same basic idea.

Hilbert space

Feature space

\[ |q_1\rangle \ldots |q_n\rangle \]

\[ x^1, \ldots, x^M \]
QC and kernel methods share the same basic idea.
Kernel methods use inner products in Hilbert space.

\[ \kappa(x, x') = \langle \phi(x), \phi(x') \rangle \]
Kernel methods use inner products in Hilbert space.

\[ \kappa(x, x') = |\langle \phi(x) | \phi(x') \rangle|^2 \]
Variational QML models are similar to kernel methods.
Variational QML models are similar to kernel methods.
Variational QML models are similar to kernel methods.
Variational QML models are similar to kernel methods.

\[ \kappa(x, x') = |\langle \phi(x) | \phi(x') \rangle|^2 \]
Variational QML models are similar to kernel methods.
EXAMPLE OF A QUANTUM EMBEDDING
A Gaussian Boson Sampler can help to compute kernels.

Schuld, Bradler, Israel, Su, Gupt 1905.12646

LINKS TO KERNEL METHODS
The covariance matrix encodes the graph.

A Gaussian state of $M$ optical modes is fully described by a covariance matrix $\sigma \in \mathbb{R}^{2M \times 2M}$ as well as a displacement vector $d \in \mathbb{R}^{2M}$.

We can associate such a state with an adjacency matrix $A$ via

$$
\sigma = (1 - X\tilde{A})^{-1} - \frac{1}{2}, \quad \text{with} \quad X = \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix}, \quad \tilde{A} = c \begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix}.
$$
The features are probabilities of detecting certain photon events.

\[
\begin{align*}
  p(O_{[000000]}) \\
p(O_{[100000]}) \\
p(O_{[200000]}) \\
p(O_{[110000]}) \\
p(O_{[300000]}) \\
p(O_{[210000]}) \\
p(O_{[111000]}) \\
p(O_{[400000]}) \\
p(O_{[220000]}) \\
p(O_{[310000]}) \\
p(O_{[211000]}) \\
p(O_{[111100]}) \\
\ldots
\end{align*}
\]

An orbit is a set of photon click events which are permutations of each other.

\[i.e., O_{[111100]} = \{[111100], [111010], [010111], [100111], [001111], \ldots\}\]
Orbits count the number of $r$-matchings in subgraphs.

\[ p(O_{[1,1,1,0]}) = p([1, 1, 1, 0]) + p([1, 1, 0, 1]) + p([1, 0, 1, 1]) + p([0, 1, 1, 1]) \]
Orbits count the number of $r$-matchings in subgraphs.

\[ p(O_{[2,1,1,0]}) = p([2,1,1,0]) + p([1,2,0,1]) + p([1,0,1,2]) + \ldots \]
We can test the kernel on standard graph data sets.

<table>
<thead>
<tr>
<th>Graph</th>
<th>Number of Graph Nodes</th>
<th>Number of Graph Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIDS</td>
<td>1728 [86%]</td>
<td>257 [84%]</td>
</tr>
<tr>
<td>IMDB-BINARY</td>
<td>806 [81%]</td>
<td>179 [96%]</td>
</tr>
<tr>
<td>MUTAG</td>
<td>118 [39%]</td>
<td>204 [34%]</td>
</tr>
<tr>
<td>NCI1</td>
<td>1836 [45%]</td>
<td>515 [46%]</td>
</tr>
<tr>
<td>PROTEINS</td>
<td>284 [81%]</td>
<td>1080 [38%*]</td>
</tr>
</tbody>
</table>
The results are promising.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>GBS ($d = 0.0$)</th>
<th>GBS ($d = 0.25$)</th>
<th>GS</th>
<th>RW</th>
<th>SM</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIDS</td>
<td>99.60 ± 0.05</td>
<td><strong>99.62 ± 0.03</strong></td>
<td>98.44 ± 0.09</td>
<td>56.95 ± 7.99</td>
<td>79.20 ± 0.68</td>
</tr>
<tr>
<td>BZR_MD</td>
<td><strong>62.73 ± 0.71</strong></td>
<td>62.13 ± 1.44</td>
<td>60.60 ± 1.77</td>
<td>49.88 ± 3.74</td>
<td>61.90 ± 1.21</td>
</tr>
<tr>
<td>COX2_MD</td>
<td>44.98 ± 1.80</td>
<td>50.11 ± 0.97</td>
<td>55.04 ± 3.33</td>
<td>57.72 ± 3.26</td>
<td><strong>66.94 ± 1.22</strong></td>
</tr>
<tr>
<td>ENZYMES</td>
<td>22.29 ± 1.60</td>
<td>28.01 ± 1.83</td>
<td>35.87 ± 2.19</td>
<td>21.13 ± 1.91</td>
<td><strong>36.70 ± 2.83</strong></td>
</tr>
<tr>
<td>ER_MD</td>
<td>70.36 ± 0.78</td>
<td><strong>70.41 ± 0.47</strong></td>
<td>65.65 ± 1.06</td>
<td>68.75 ± 0.53</td>
<td>68.21 ± 0.99</td>
</tr>
<tr>
<td>FINGERPRINT</td>
<td>65.42 ± 0.49</td>
<td><strong>65.85 ± 0.36</strong></td>
<td>64.10 ± 1.52</td>
<td>47.69 ± 0.21</td>
<td>47.14 ± 0.62</td>
</tr>
<tr>
<td>IMDB-BIN</td>
<td>64.09 ± 0.34</td>
<td>68.71 ± 0.59</td>
<td>68.37 ± 0.62</td>
<td>66.38 ± 0.21</td>
<td>out of time*</td>
</tr>
<tr>
<td>MUTAG</td>
<td><strong>86.41 ± 0.33</strong></td>
<td>85.58 ± 0.59</td>
<td>81.08 ± 0.93</td>
<td>83.02 ± 1.08</td>
<td>83.14 ± 0.24</td>
</tr>
<tr>
<td>NCI1</td>
<td><strong>63.61 ± 0.00</strong></td>
<td>62.79 ± 0.00</td>
<td>49.96 ± 3.27</td>
<td>52.36 ± 2.63</td>
<td>51.36 ± 1.88</td>
</tr>
<tr>
<td>PROTEINS</td>
<td><strong>66.88 ± 0.22</strong></td>
<td>66.14 ± 0.48</td>
<td>65.91 ± 1.29</td>
<td>56.27 ± 1.23</td>
<td>63.03 ± 0.84</td>
</tr>
<tr>
<td>PTC_FM</td>
<td>53.84 ± 0.96</td>
<td>52.45 ± 1.78</td>
<td><strong>59.48 ± 1.95</strong></td>
<td>51.97 ± 2.68</td>
<td>54.92 ± 2.94</td>
</tr>
</tbody>
</table>
OPEN PROBLEMS
This is only the beginning of the journey..

- How can we benchmark quantum models?
This is only the beginning of the journey..

- How can we benchmark quantum models?
- Is there a connection between quantum theory and deep learning?
This is only the beginning of the journey..

- How can we benchmark quantum models?
- Is there a connection between quantum theory and deep learning?
- Do quantum models inherently regularise?
This is only the beginning of the journey..

- How can we benchmark quantum models?
- Is there a connection between quantum theory and deep learning?
- Do quantum models inherently regularise?
- How does noise impact applications?
This is only the beginning of the journey..

- How can we benchmark quantum models?
- Is there a connection between quantum theory and deep learning?
- Do quantum models inherently regularise?
- How does noise impact applications?
- What domains is QML good for?
This is only the beginning of the journey..

- How can we benchmark quantum models?
- Is there a connection between quantum theory and deep learning?
- Do quantum models inherently regularise?
- How does noise impact applications?
- What domains is QML good for?
- Do QML ideas scale?
This is only the beginning of the journey..

- How can we benchmark quantum models?
- Is there a connection between quantum theory and deep learning?
- Do quantum models inherently regularise?
- How does noise impact applications?
- What domains is QML good for?
- Do QML ideas scale?
- What optimization strategies work for variational circuits?
This is only the beginning of the journey.

- How can we benchmark quantum models?
- Is there a connection between quantum theory and deep learning?
- Do quantum models inherently regularise?
- How does noise impact applications?
- What domains is QML good for?
- Do QML ideas scale?
- What optimization strategies work for variational circuits?
- Which circuit architectures are good for ML?
Thank you!

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