Power of Data in Quantum Machine Learning

Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, Jarrod McClean

Raphael Blankson

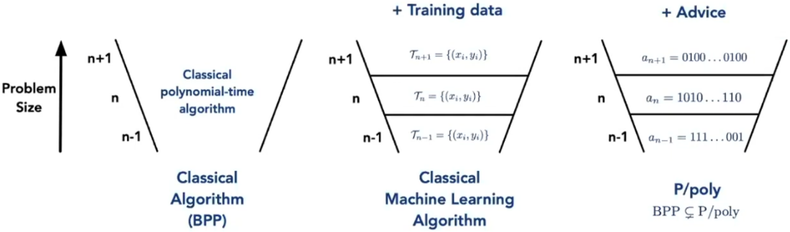
April 27, 2021

Novosibirsk State University

Intro

Intro

- Machine learning is just classical algorithm + training data.
- Training data are a set of information of the form (x, y), where x is the input and y is the output.
- Training data can be seen as a restricted form of advice.



Computational Power of Data

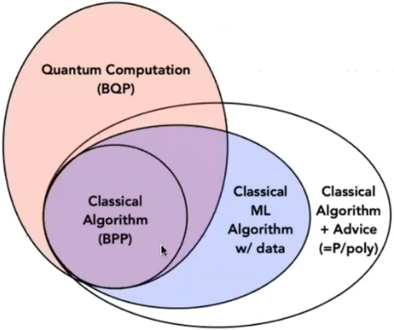
- We define the complexity class of classical algorithms with data as follows. Given a language L in the complexity class.

- There exists a probabilistic Turing machine M that takes input x of size n along with a training data of size $\text{poly}(n)$:
$$\mathcal{T} = \{(x_i, y_i)\},$$
where x_i is sampled from some distribution $\mathcal{D}_n$ over all inputs of size n and $y_i = 1$ if $x_i \in L$ else $y_i = 0$.

- M runs for polynomial time on all input x.
For all $x \in L$, M outputs 1 with probability $\geq 2/3$.
For all $x \notin L$, M outputs 1 with probability $< 1/3$.

Computational power of Data

- Classical algorithms that could obtain and learn from data can be computationally more powerful.



Power of Data: Implications

1. Classical ML algorithm can learn to solve some quantum many-body problems (by learning from data obtained in nature).

2. Classical ML algorithm can rival existing quantum ML even for learning quantum models (that are hard to simulate classically).

We will focus on the second implication for this talk.

Characterizing quantum advantage in learning problems

- People often expect deep quantum neural network or classically-hard quantum kernel function to yield quantum advantage in machine learning.
- But this may not be true due to the availability of data.
- When would existing quantum models be void of quantum advantage?

Quantum Model

- The quantum model we consider:

  1. x: a classical vector
  2. $\rho(x) = U_{\text{enc}}(x)|0^n\rangle\langle 0^n|U_{\text{enc}}(x)^\dagger$
  3. U: a unitary evolution applied on $\rho(x)$
  4. O: an observable measured on $U\rho(x)U^\dagger$

- This also corresponds to computation that can be performed by quantum computers.

Example Quantum Model

- Quantum neural networks consist of
  > embedding input x_i into quantum Hilbert space ρ_i
  > unitary evolution U_{QNN}
  > obtain expectation value of observable O

- Quantum kernel methods consist of
  > embedding input x_i into quantum Hilbert space ρ_i
  > training a kernel method with kernel function

$$k(x_i, x_j) = \mathrm{Tr}(\rho_i \rho_j)$$

A useful fact: Quantum kernel method is equivalent to training an arbitrarily deep quantum neural network that measures any observable at the end.

Machine Learning Model

- Consider a training data $\{x^i, \mathrm{Tr}(OU\rho(x^i)U^\dagger)\}_{i=1}^N$, and we train a classical neural network $f(x)$ with large hidden layer to minimize

$$\min_f \lambda R(f) + \sum_{i=1}^{N} \|f(x^i) - \mathrm{Tr}(OU\rho(x^i)U^\dagger)\|^2.$$

- The trained neural network [1, 2] is equivalent to

$$f(x) = \mathrm{Tr}\left(U^\dagger OU \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} k(x, x^i)(K^T K + \lambda I)_{ij}^{-1} k(x^j, x^k)\rho(x^k) \right),$$

for $k(x, x')$: a kernel function, $K_{ij} = k(x^i, x^j)$, and $\lambda \geq 0$.

- Also holds for more traditional ML models (e.g., kernel SVM) as well as quantum kernel methods.

Is this Quantum Model Learnable?

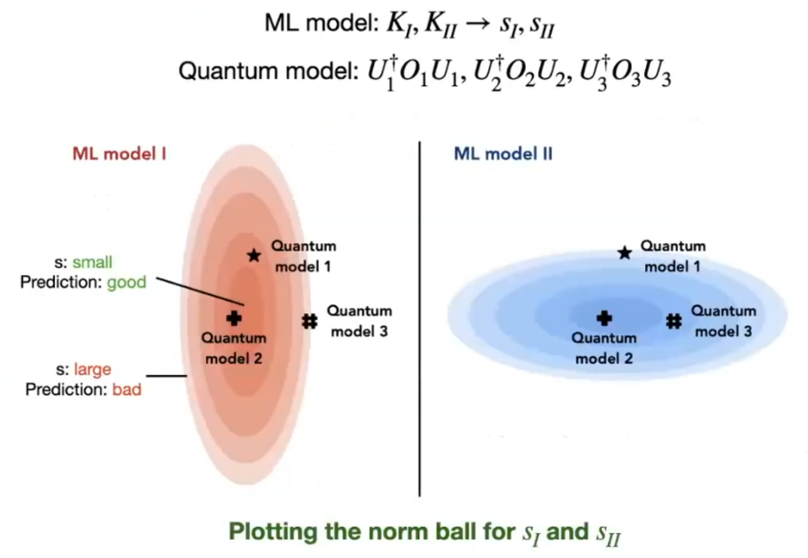
- Quantum model: $x \to \rho(x) \to$ measure $U^\dagger O U$

- Machine learning model: defines geometry on input x
$$K_{ij} = k(x^i, x^j) = \phi(x^i)^T \phi(x^j),$$
where i, j are taken over all training data.

- Define $s_K(U^\dagger O U) = \mathrm{Tr}(A(U^\dagger O U) \otimes (U^\dagger O U)) \geq 0$,
where $A = \sum_{ij} (K^{-1})_{ij} \rho(x^i) \otimes \rho(x^j)$.

- If $s_K(U^\dagger O U)$ is small, then the ML model can learn to predict the quantum model well.

Visualize $S_k(O^U)$

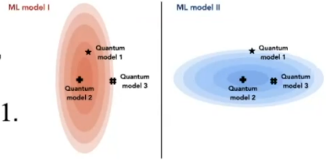


Geometric difference

- Consider two ML models K_1, K_2. The corresponding $s_{K_1}(O^U)$ and $s_{K_2}(O^U)$ are two different squared norms defined on the space of observables.
- $s_{K_1}(O^U)$ and $s_{K_2}(O^U)$ can be related through

$$s_{K_1} \leq g(K_1 || K_2)^2 s_{K_2},$$

where $g(K_1 || K_2) = \sqrt{\| \sqrt{K_2} K_1^{-1} \sqrt{K_2} \|_\infty} \geq 1.$

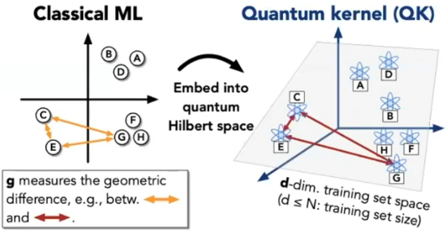


- There exists data set that satisfies $s_{K_1} = g(K_1 || K_2)^2 s_{K_2}$.
- If $g(K_1 || K_2)$ is small, then K_1 will always have a similar or smaller model complexity compared with K_2. (Hence K_1 would predict similar or better)

Geometric difference for accessing quantum advantage

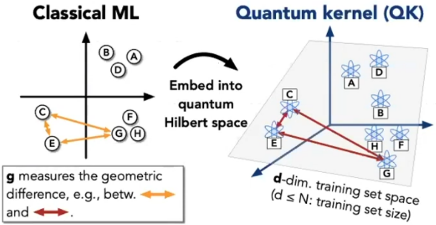
- Consider $g_{\mathrm{CQ}} = g(K^{\mathrm{C}}||K^{\mathrm{Q}})$: geometric difference between classical ML model and quantum ML model, e.g., quantum kernel SVM.

- If g_{CQ} is small, then classical ML will predict similar or better than quantum ML for learning any quantum model $U^{\dagger}OU$.

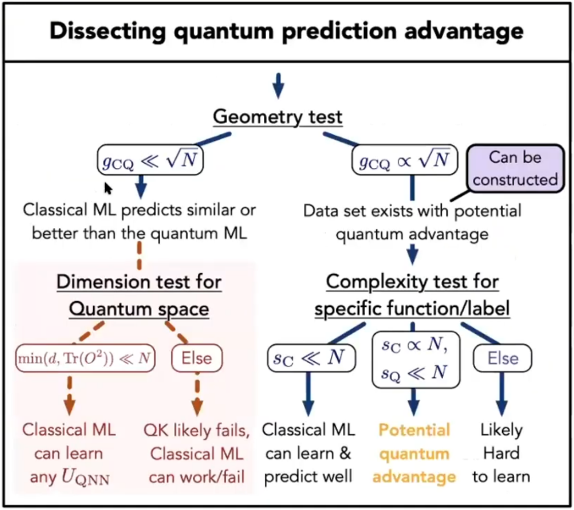


Geometric difference for accessing quantum advantage

- Consider $g_{CQ} = g(K^C || K^Q)$: geometric difference between classical ML model and quantum ML model, e.g., quantum kernel SVM.
- If g_{CQ} is large, because there exists data set that satisfies $s_C = g_{CQ}^2 s_Q$, we have $s_C \gg s_Q$ implying a prediction advantage using the quantum ML.



Quantum prediction advantage



Limitations of Quantum Kernel Methods

- When the quantum states $\rho(x^i)$ for the training set span a large dimension quantum Hilbert space, all inputs are too far apart, so

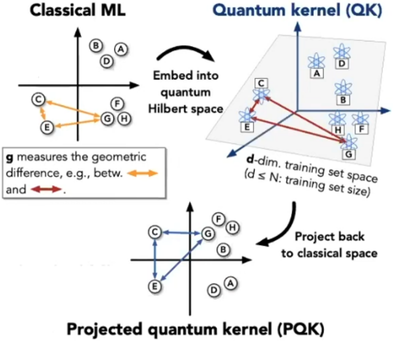
$$K^{Q} \approx I \quad \text{and} \quad g_{CQ} = \sqrt{\|\sqrt{K_Q} K_C^{-1} \sqrt{K_Q}\|_\infty} \approx 1.$$

- This means classical ML can often compete or outperform quantum kernel methods in learning any quantum models.
- One could rigorously show that for simple quantum models, quantum kernel need exponential number of data, while classical ML only need linear.
- We see classical ML outperforming quantum kernel throughout numerics.

Prediction error bound for QK: $\mathbb{E}_x |g(x) - \mathrm{Tr}(O^U \rho(x))| \leq \mathcal{O}\left(\sqrt{\frac{\min(d, \mathrm{Tr}(O^2))}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right)$

Proposed Solution

- Large quantum Hilbert space dimension makes quantum ML suffers more than classical ML.
- Projects quantum states back to classical space, e.g. using reduced observable or classical shadow [1].
- Define kernel in the classical space.
- We call this the projected quantum kernel (PQK).

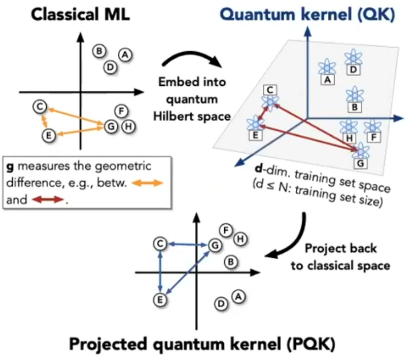


Projected quantum Kernel - PQK

- PQK requires quantum computer to compute (by going through QK).
- PQK results in much higher geometric difference. (because QK has $g \approx 1$)
- Simple-to-prove rigorous advantage in a learning problem based on discrete logarithm [1].

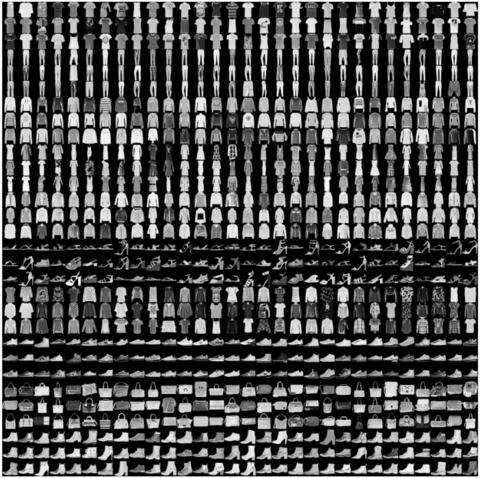
$$y(x) = \begin{cases} +1, & \log_g(x) \in [s, s + \frac{p-3}{2}], \\ -1, & \log_g(x) \notin [s, s + \frac{p-3}{2}], \end{cases}$$

- The proof that QK can learn the above problem is much more complicated [1].



Experiments

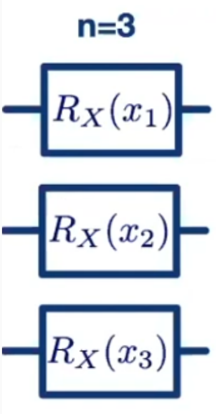
Experiments - Fashion-MNIST



- MNIST is too easy (can predict well with one pixel) and overused.
- Fashion-MNIST is a harder alternative with the same format.
- Focus only on binary classification (dresses vs shirt)

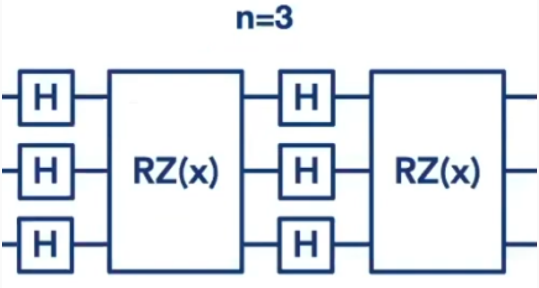
Experiments - Embedding Strategy

- First map each image to n-dimensional vector by PCA.
- Three (3) different embedding strategies used:
  - E1: Separable Rotation circuit



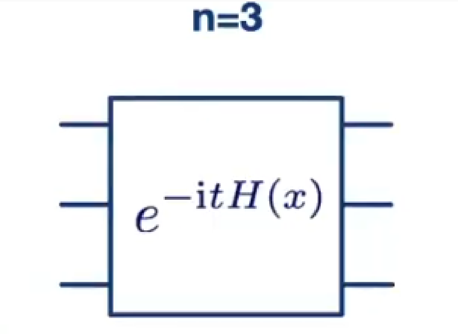
Experiments - Embedding Strategy

- E2: IQP circuit (by IBM)



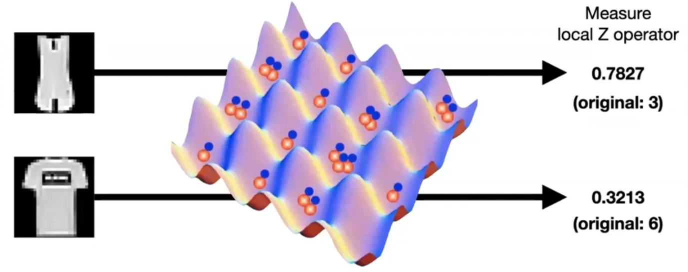
Experiments - Embedding Strategy

- E3: Hamiltonian circuit



$$H(x) = \sum_i x_i \langle \sigma_i, \sigma_{i+1} \rangle$$

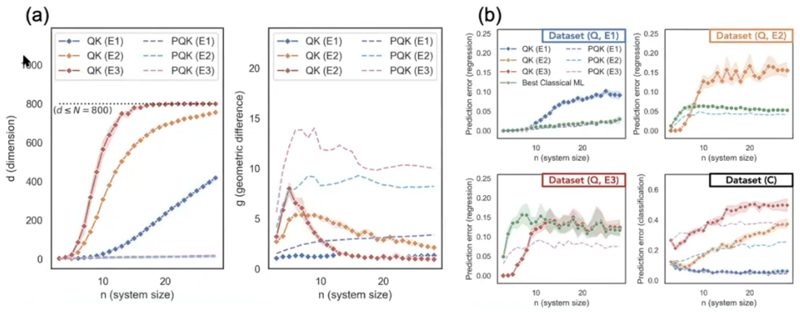
- Dataset generated by quantum process: replacing the original labels with output from hard-to-simulate Hamiltonian evolution.



Classical ML methods:

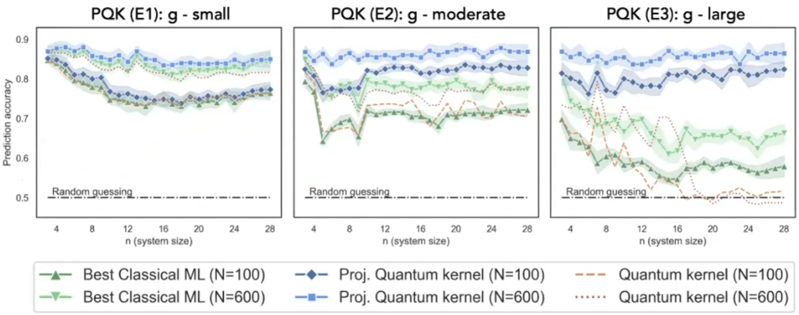
- Random Forest
- Gradient Boosting
- Adaboost
- Gaussian kernel
- Linear model
- Convolutional neural network
- Feedforward neural network - All hyper-parameters were properly tuned and report the best model.

Results



1. As dimension increases, geometric difference for QK decreases.
2. Small geometric difference results in similar to or underperform classical ML.
3. PQK projects to small dimensional space, but have large geometric difference.

Results



1. When geometric difference is large, data sets exists with large prediction advantage.
2. One can see significant advantage using quantum ML for these data sets.
3. BQP should still be larger than P/poly (hence any classical ML with data).

Conclusion

Conclusion

- Data provide computational power that enables classical ML algorithms to become stronger than one expects.
- Classical ML can rival quantum ML and could outperform existing quantum ML on quantum tasks.
- However, quantum ML should still be stronger than classical ML (existing QML are not great).
- Quantum advantage in prediction accuracy is still possible - more investigations are needed to justify this claim.

Thanks

Questions?