Applied Quantum Algorithms - Lecture 8 - Quantum Neural Networks

Giorgio Tosti Balducci

Delft University of Technology, The Netherlands

Apr 26, 2023

Contents

Supervised learning

Aim: Given some data and their **labels**, predict the function[1] that assigns **unseen** data to the correct label.

[1] A different formulation may be to predict the distributions $p(x, y)$ or $p\left(y|x\right)$.

Supervised learning

Aim: Given some data and their **labels**, predict the function[1] that assigns **unseen** data to the correct label.

$$
\begin{array}{ll}
\text{Given} & D = \{(x_i, y_i) \mid x_i \in S;\ y_i = f(x_i)\}, \\
\text{where} & S \subseteq \mathbb{R}^n, \\
\text{predict} & f : S \to \mathit{labels}
\end{array}
$$

[1] A different formulation may be to predict the distributions $p(x, y)$ or $p(y|x)$.

Supervised learning

Aim: Given some data and their **labels**, predict the function[1] that assigns **unseen** data to the correct label.

$$
\begin{array}{ll}
\text{Given} & D = \{(x_i, y_i) \mid x_i \in S;\ y_i = f(x_i)\}, \\
\text{where} & S \subseteq \mathbb{R}^n, \\
\text{predict} & f : S \to labels
\end{array}
$$

- *Classification*: discrete labels
- *Regression*: continuous labels

[1] A different formulation may be to predict the distributions $p(x, y)$ or $p(y|x)$.

Supervised learning

function family and training

Instead of searching among all possible $f\colon S \to \mathit{labels}$, we choose a *function family* or *ansatz*, characterized by *free parameters*.

$$\{ f_\theta | \, f_\theta : S, \Theta \to \mathit{labels} \}$$
$$\Theta \subseteq \mathbb{R}^p$$

Supervised learning

function family and training

Instead of searching among all possible $f\colon S \to \mathit{labels}$, we choose a *function family* or *ansatz*, characterized by *free parameters*.

$$\{f_\theta \mid f_\theta : S, \Theta \to \mathit{labels}\}$$
$$\Theta \subseteq \mathbb{R}^p$$

Training a supervised learning model - practical definition

Find θ s.t. f_θ best approximizes f w.r.t. a metric of error or *loss*.

Supervised learning

function family and training

Instead of searching among all possible $f: S \rightarrow labels$, we choose a *function family* or *ansatz*, characterized by *free parameters*.

$$\{f_\theta | \, f_\theta : S, \Theta \rightarrow labels\}$$
$$\Theta \subseteq \mathbb{R}^p$$

Training a supervised learning model - practical definition

Find θ s.t. f_θ best approximizes f w.r.t. a metric of error or *loss*.

Mean squared error	$\sum_{i=1}^{N} \left(y_i - f_\theta\left(x_i\right)\right)^2$	Regression
Hinge loss	$\sum_{i=1}^{N} \max\left(0, 1 - y_i f(x_i)\right)$	Classification

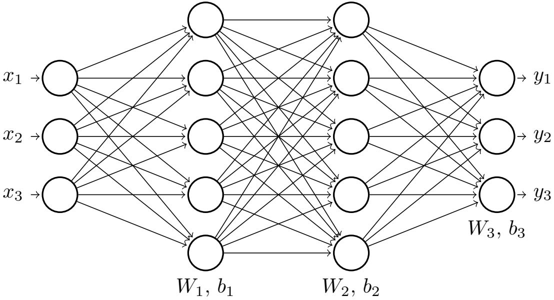
Supervised learning algorithms

- Neural networks (this lecture)
- Support vector machines (next lecture)

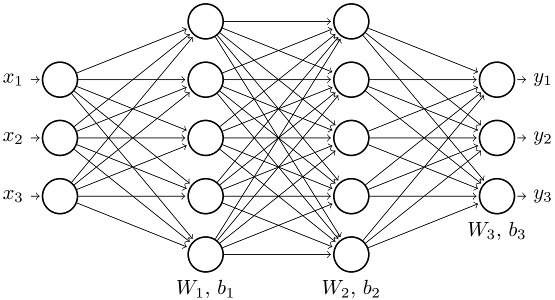
Supervised learning algorithms

- Neural networks (this lecture)
- Support vector machines (next lecture)
- Naive Bayes
- Linear regression
- Logistic regression
- k-nearest neighbours
- Random forest
- ...

Classical neural networks



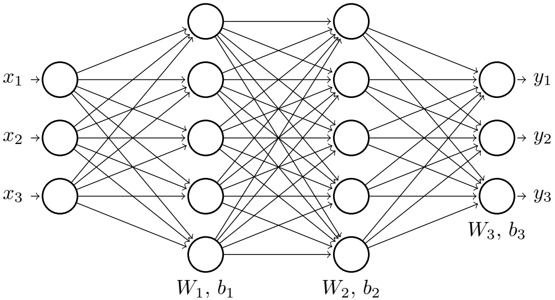
Classical neural networks



- $f_\theta = f\Big(\mathbf{x}, W^{(1)}, \mathbf{b}^{(1)} W^{(2)}, \mathbf{b}^{(2)}, \ldots, W^{(l)}, \mathbf{b}^{(l)}\Big)$
- Layered structure: $\mathbf{x}^{(l+1)} = \sigma\left(W^{(l)}\mathbf{x}^{(l)} + \mathbf{b}^{(l)}\right)$

[2] M. A. Nielsen (2018). *Neural Networks and Deep Learning*. misc

Classical neural networks



- $f_\theta = f\left(\mathbf{x}, W^{(1)}, \mathbf{b}^{(1)} W^{(2)}, \mathbf{b}^{(2)}, \ldots, W^{(l)}, \mathbf{b}^{(l)}\right)$
- Layered structure: $\mathbf{x}^{(l+1)} = \sigma\left(W^{(l)}\mathbf{x}^{(l)} + \mathbf{b}^{(l)}\right)$

Furthermore, NNs are universal function approximators[2].

[2] M. A. Nielsen (2018). *Neural Networks and Deep Learning*. misc

Classical neural networks

Training - backpropagation

Training neural networks uses gradient-based algorithms

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_{\theta^{(t)}} L$$

Classical neural networks

Training - backpropagation

Training neural networks uses gradient-based algorithms

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_{\theta^{(t)}} L$$

Once we define

1. free parameters
2. model
3. loss function

we can build the so-called *computation graph*.

Classical neural networks

Training - backpropagation

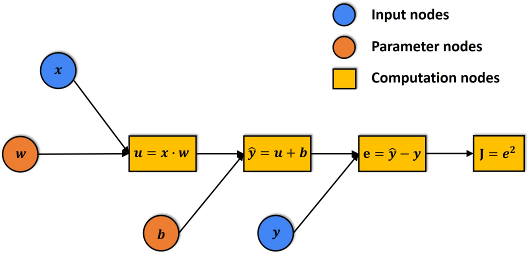
Training neural networks uses gradient-based algorithms

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_{\theta^{(t)}} L$$

Once we define

1. free parameters
2. model
3. loss function

we can build the so-called *computation graph*.



Classical neural networks

Training - backpropagation

To build the gradient at time (epoch) t, we need 2 steps

1. *forward* direction - get the loss value and evaluate local gradients
2. *backward* direction: multiplications (chain rule)

Classical neural networks

Training - backpropagation

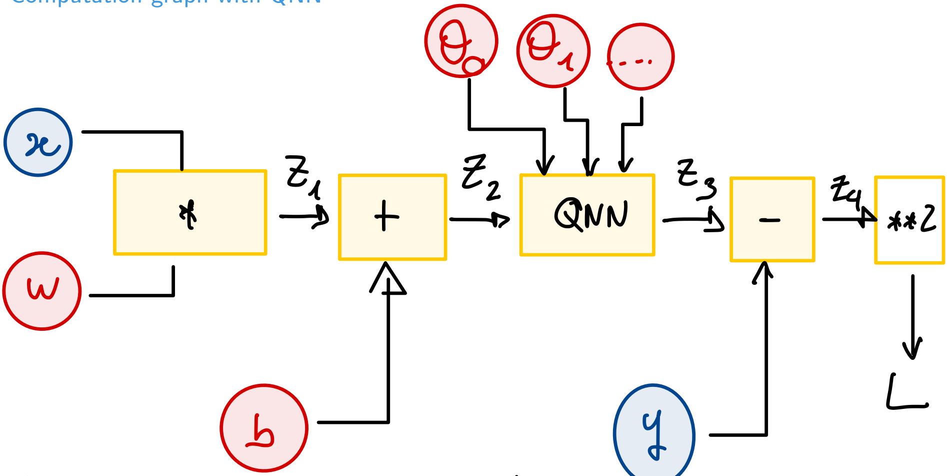
To build the gradient at time (epoch) t, we need 2 steps

1. *forward* direction - get the loss value and evaluate local gradients
2. *backward* direction: multiplications (chain rule)

It works differently in QNNs, since *we cannot store intermediate gradients*, due to the no-cloning theorem. More later...

Intermezzo

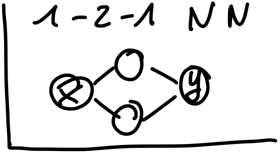
Computation graph with QNN

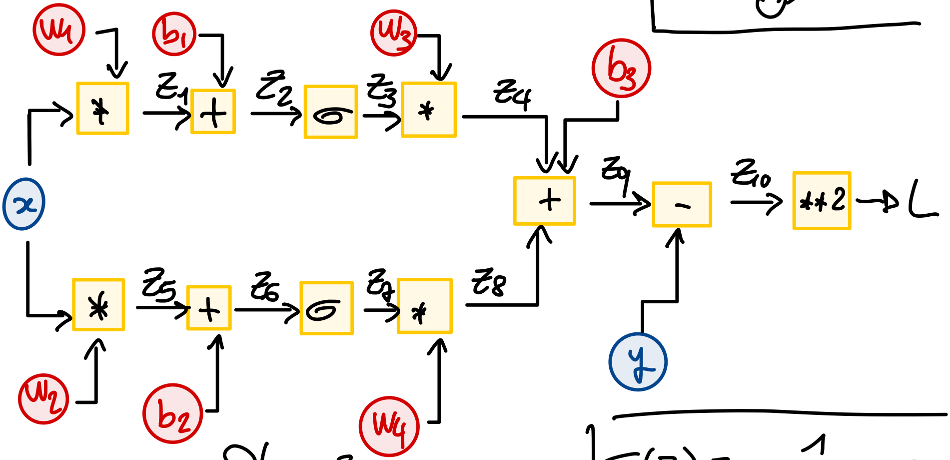


- Backprop @ graph level
- No backprop for $\frac{\partial QNN}{\partial \theta_i}$

Classical neural networks

Backpropagation example





$$\frac{\partial L}{\partial w_1} = ?$$

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Quantum machine learning

Some informal definitions

Quantum-enhanced machine learning

Quantum computation to speed-up classical machine learning operations.

Example: SVM with quantum linear systems solver algorithm

Quantum machine learning

Some informal definitions

Quantum-enhanced machine learning

Quantum computation to speed-up classical machine learning operations.

Example: SVM with quantum linear systems solver algorithm

Limitations

- Input/output problem (QRAM?)
- De-quantization argument: for a fair comparison, *both classical and quantum* algorithm must have access to a QRAM → classical algorithm also runs in $O(\log(N))$

Quantum machine learning

Some informal definitions

Machine learning in quantum feature spaces

Information is *encoded* into quantum states and classical algorithms find the optimal model.

[3] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd (Sept. 2017). "Quantum machine learning". In: *Nature* 549.7671, pp. 195–202. DOI: 10.1038/nature23474

Quantum machine learning

Some informal definitions

Machine learning in quantum feature spaces

Information is *encoded* into quantum states and classical algorithms find the optimal model.

Note: From here on, we will be focusing on ML in quantum feature spaces. For a review of quantum-enhanced ML, see [3]

[3] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd (Sept. 2017). "Quantum machine learning". In: *Nature* 549.7671, pp. 195–202. DOI: 10.1038/nature23474

Why QML?

The advantage POV

As with all quantum algorithms, we ask ourselves: *how is QML better than ML?*

[4] Y. Liu, S. Arunachalam, and K. Temme (July 2021). "A rigorous and robust quantum speed-up in supervised machine learning". In: *Nature Physics* 17.9, pp. 1013–1017. DOI: 10.1038/s41567-021-01287-z

[5] the key idea of kernel theory, more in the next lecture

Why QML?

The advantage POV

As with all quantum algorithms, we ask ourselves: *how is QML better than ML?*

Quantum advantage can be proved by

- Mapping data to certain states
- These states are hard to produce classically, but easy to produce (BQP) via a quantum algorithm
  - e.g. by solving the discrete log problem [4]
- We can compute inner products efficiently[5]

[4] Y. Liu, S. Arunachalam, and K. Temme (July 2021). "A rigorous and robust quantum speed-up in supervised machine learning". In: *Nature Physics* 17.9, pp. 1013–1017. DOI: 10.1038/s41567-021-01287-z

[5] the key idea of kernel theory, more in the next lecture

Why QML?

The advantage POV

However...

- Are there any *practically interesting* dataset in which QML can achieve advantage?

Why QML?

The advantage POV

However...

- Are there any *practically interesting* dataset in which QML can achieve advantage?
- Is advantage even the right thing to look for?

Why QML?

Is quantum advantage the right goal for quantum machine learning? [6]

Advantage is just one of the possible points of view and maybe not the most useful one.

[6] M. Schuld and N. Killoran (July 2022). "Is Quantum Advantage the Right Goal for Quantum Machine Learning?" In: *PRX Quantum* 3.3. DOI: 10.1103/prxquantum.3.030101

Why QML?

Is quantum advantage the right goal for quantum machine learning? [6]

Advantage is just one of the possible points of view and maybe not the most useful one.

1. What is(are) the fundamental building block(s) of QML models?
   - a.k.a. what is the *quantum perceptron*?

[6] M. Schuld and N. Killoran (July 2022). "Is Quantum Advantage the Right Goal for Quantum Machine Learning?" In: *PRX Quantum* 3.3. DOI: 10.1103/prxquantum.3.030101

Why QML?

Is quantum advantage the right goal for quantum machine learning? [6]

Advantage is just one of the possible points of view and maybe not the most useful one.

1. What is(are) the fundamental building block(s) of QML models?
   - a.k.a. what is the *quantum perceptron*?
2. Can classical learning theory give us guarantees on the learning performance of QML models?
   - Yes, and the link is kernel theory.

[6] M. Schuld and N. Killoran (July 2022). "Is Quantum Advantage the Right Goal for Quantum Machine Learning?" In: *PRX Quantum* 3.3. DOI: 10.1103/prxquantum.3.030101

Why QML?

Is quantum advantage the right goal for quantum machine learning? [6]

Advantage is just one of the possible points of view and maybe not the most useful one.

1. What is(are) the fundamental building block(s) of QML models?
   - a.k.a. what is the *quantum perceptron*?
2. Can classical learning theory give us guarantees on the learning performance of QML models?
   - Yes, and the link is kernel theory.
3. Can we automate the learning process of (certain) QML models?
   - In particular, automatic differentiation.

[6] M. Schuld and N. Killoran (July 2022). "Is Quantum Advantage the Right Goal for Quantum Machine Learning?" In: *PRX Quantum* 3.3. DOI: 10.1103/prxquantum.3.030101

The quantum neural network model

3 ingredients

- A data encoding unitary: $U(\mathbf{x})$
- A variational unitary (anzatz): $W(\theta)$
- An observable (measurement operator): $\mathcal{M}$

The quantum neural network model

3 ingredients

- A data encoding unitary: $U(\mathbf{x})$
- A variational unitary (anzatz): $W(\theta)$
- An observable (measurement operator): $\mathcal{M}$

$$f(\mathbf{x}, \theta) = \langle \psi(\mathbf{x}, \theta) | \mathcal{M} | \psi(\mathbf{x}, \theta) \rangle ,$$

The quantum neural network model

3 ingredients

- A data encoding unitary: $U(\mathbf{x})$
- A variational unitary (anzatz): $W(\theta)$
- An observable (measurement operator): $\mathcal{M}$

$$f(\mathbf{x}, \theta) = \langle \psi(\mathbf{x}, \theta) | \mathcal{M} | \psi(\mathbf{x}, \theta) \rangle,$$

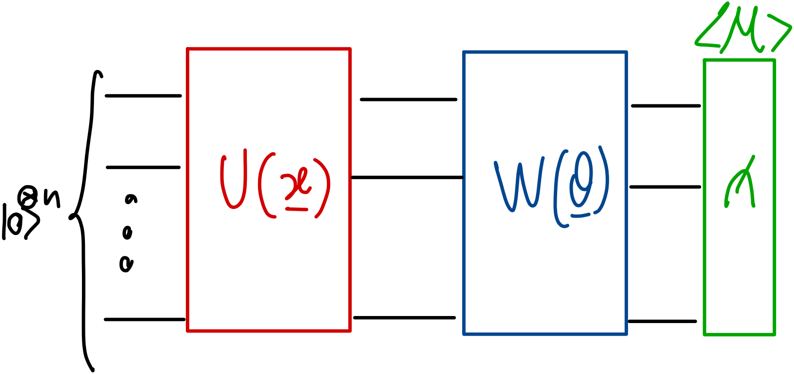
where

$$|\psi(\mathbf{x}, \theta)\rangle = W(\theta)\, U(\mathbf{x})\, |0\rangle$$

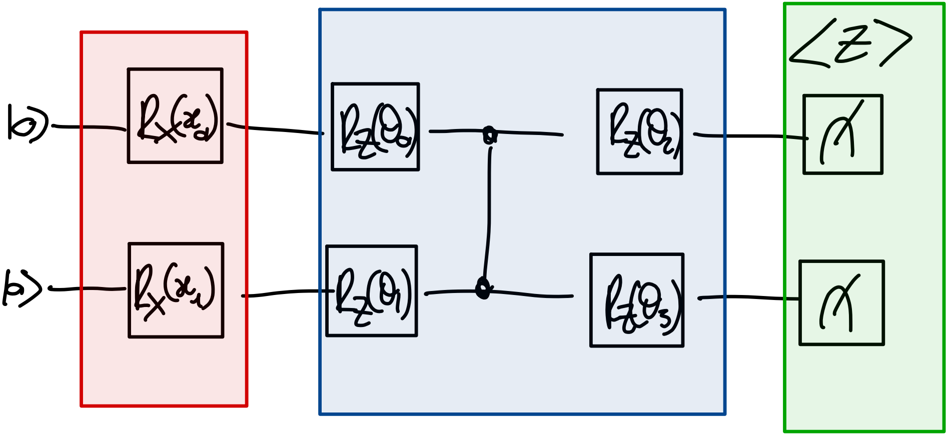
and

$$\mathbf{x} \in \mathbb{R}^m,\ \theta \in \mathbb{R}^p$$

The quantum neural network model



A simple QNN example



QNN model evaluation

The model $f = \langle \mathcal{M} \rangle_{\mathbf{x},\theta}$ is the expectation value of the observable $\mathcal{M}$ of the state $\psi(\mathbf{x}, \theta)$

[7] https://en.wikipedia.org/wiki/Hadamard_test_(quantum_computation)

[8] https://en.wikipedia.org/wiki/Swap_test

QNN model evaluation

The model $f = \langle \mathcal{M} \rangle_{\mathbf{x},\theta}$ is the expectation value of the observable $\mathcal{M}$ of the state $\psi(\mathbf{x}, \theta)$

- In practice, we can compute $\left\langle \mathcal{M}_{\mathbf{x},\theta} \right\rangle$ if we know how to measure in $\mathcal{M}$ basis
- See previous slide where we measured in the X basis.

[7] https://en.wikipedia.org/wiki/Hadamard_test_(quantum_computation)

[8] https://en.wikipedia.org/wiki/Swap_test

QNN model evaluation

The model $f = \langle \mathcal{M} \rangle_{\mathbf{x},\theta}$ is the expectation value of the observable $\mathcal{M}$ of the state $\psi(\mathbf{x}, \theta)$

- In practice, we can compute $\left\langle \mathcal{M}_{\mathbf{x},\theta} \right\rangle$ if we know how to measure in $\mathcal{M}$ basis
- See previous slide where we measured in the X basis.
- Alternatively, we use Hadamard[7] or SWAP[8] tests

[7] https://en.wikipedia.org/wiki/Hadamard_test_(quantum_computation)

[8] https://en.wikipedia.org/wiki/Swap_test

QNN model evaluation

The model $f = \langle \mathcal{M} \rangle_{\mathbf{x},\theta}$ is the expectation value of the observable $\mathcal{M}$ of the state $\psi(\mathbf{x}, \theta)$

- In practice, we can compute $\left\langle \mathcal{M}_{\mathbf{x},\theta} \right\rangle$ if we know how to measure in $\mathcal{M}$ basis
- See previous slide where we measured in the X basis.
- Alternatively, we use Hadamard[7] or SWAP[8] tests
- Finite sampling: $O(1/\varepsilon^2)$ measurement to estimate $\langle M \rangle$ up to precision ε
    - Due to Chebyshev's inequality
    - In practice, a convergence study is necessary

[7] https://en.wikipedia.org/wiki/Hadamard_test_(quantum_computation)

[8] https://en.wikipedia.org/wiki/Swap_test

QNN model training

Training is hybrid, as for all variational algorithms. At each epoch

QNN model training

Training is hybrid, as for all variational algorithms. At each epoch

1. (Quantum step) *Evaluate* the model $f = \langle \mathcal{M} \rangle_{\mathbf{x},\theta^{(t)}}$ and maybe $\nabla_\theta f$.
   - More on gradients later on...
2. (Classical step 1) *Forward step* in the computation graph.
   - i.e. we evaluate a loss function $L\left(\theta^{(t)}\right)$ and the local gradients.
3. (Classical step 2) *Backward step* in the computational graph.
4. (Classical step 3) *Parameter update*.

$$\mathrm{GD}: \qquad \theta^{(t+1)} = \theta^{(t)} - \eta \nabla_{\theta^{(t)}} L$$

QNN model training

Do you think it is a good idea to evaluate $\nabla_\theta f$ with finite differences?
Why? Why not?

Encoding classical data in quantum states

[9] even better, with high *margin*

Encoding classical data in quantum states

A good choice of data embedding can make the difference in terms of

[9] even better, with high *margin*

Encoding classical data in quantum states

- Runtime of the QML routine
  - how does the number of operations scale in the number of features N and dataset dimension M

[9] even better, with high *margin*

Encoding classical data in quantum states

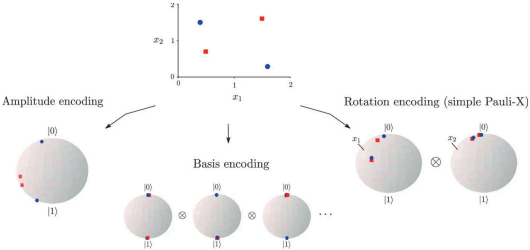
- Runtime of the QML routine
  - how does the number of operations scale in the number of features N and dataset dimension M
- Learning performance
  - Encoding data means to *map them to a high-dimensional Hilbert space*
  - E.g., in classification, we can linearly separate the data of different classes[9], but also completely shuffle them!

[9] even better, with high *margin*

Encoding classical data in quantum states

Data encoding as a feature map

$$\phi(\mathbf{x}) : \mathcal{X} \rightarrow \mathcal{F} \subseteq \mathcal{H}^{2^n}$$



Encoding classical data in quantum states

Amplitude encoding

Encoding classical data in quantum states

Amplitude encoding

We have $\mathbf{x} \in \mathbb{R}^N$. To make it a valid quantum state:

1. Normalize: $\frac{\mathbf{x}}{\|x\|_2}$
2. Pad until the closest power of 2: $x_N, \ldots, x_{2^n-1} = 0$

Encoding classical data in quantum states

Amplitude encoding

We have $\mathbf{x} \in \mathbb{R}^N$. To make it a valid quantum state:

1. Normalize: $\frac{\mathbf{x}}{\|x\|_2}$
2. Pad until the closest power of 2: $x_N, \ldots, x_{2^n-1} = 0$

Amplitude encoding (AE) is the identity transformation of this preprocessed state

$$\phi(\mathbf{x}) = |x\rangle = \sum_{i=0}^{N-1} x_i |i\rangle$$

[10] M. Mottonen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa (2004). *Transformation of quantum states using uniformly controlled rotations*. arXiv: quant-ph/0407010 [quant-ph]

Encoding classical data in quantum states

Amplitude encoding

We have $\mathbf{x} \in \mathbb{R}^N$. To make it a valid quantum state:

1. Normalize: $\frac{\mathbf{x}}{\|x\|_2}$
2. Pad until the closest power of 2: $x_N, \dots, x_{2^n-1} = 0$

Amplitude encoding (AE) is the identity transformation of this preprocessed state

$$\phi(\mathbf{x}) = |x\rangle = \sum_{i=0}^{N-1} x_i |i\rangle$$

- AE is a linear transformation: if data is not linearly separable in the (normalized and padded) original space, then it won't be in the mapped space.
- num. qubits: $\log(N)$
- runtime: $O(N)$[10] *in general*

[10] M. Mottonen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa (2004). *Transformation of quantum states using uniformly controlled rotations*. arXiv: quant-ph/0407010 [quant-ph]

Encoding classical data in quantum states

Amplitude encoding

By adding an additional $\log(M)$ qubits, we can also encode data in superposition

$$|\psi_{\mathbf{x}}\rangle = \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} \sum_{i=0}^{N-1} x_i |i\rangle |m\rangle = \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} |\psi_{\mathbf{x}_m}\rangle |m\rangle$$

- num. qubits: $\log(NM)$
- runtime: $O(NM)$

Encoding classical data in quantum states

Basis encoding

Encoding classical data in quantum states

Basis encoding

For basis encoding, we first need to choose a binary representation of the features x_i. E.g., in floating point

$$b_i = b_{i,s} b_{i,n_e-1} \dots b_{i,0} b_{i,n_m-1} \dots b_{i,0}$$

Encoding classical data in quantum states

Basis encoding

For basis encoding, we first need to choose a binary representation of the features x_i. E.g., in floating point

$$b_i = b_{i,s} b_{i,n_e-1} \ldots b_{i,0} b_{i,n_m-1} \ldots b_{i,0}$$

With $\tau = 1 + n_e + n_m$ we can encode *each feature of each datapoint* into a different basis state. If we concatenate the features:

$$\phi(\mathbf{x}) = |b_0 \ldots b_{N-1}\rangle$$

Encoding classical data in quantum states

Basis encoding

For basis encoding, we first need to choose a binary representation of the features x_i. E.g., in floating point

$$b_i = b_{i,s} b_{i,n_e-1} \dots b_{i,0} b_{i,n_m-1} \dots b_{i,0}$$

With $\tau = 1 + n_e + n_m$ we can encode *each feature of each datapoint* into a different basis state. If we concatenate the features:

$$\phi(\mathbf{x}) = |b_0 \dots b_{N-1}\rangle$$

Of course, we can also create superpositions

$$|\psi_{\mathbf{x}}\rangle = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} \left|\mathbf{x}^{(i)}\right\rangle$$

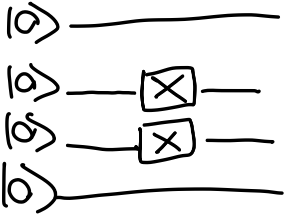
Encoding classical data in quantum states

Basis encoding

	Single sample	Dataset
num. qubits	$N\tau$	$N\tau$
runtime	$O(N\tau)$	$O(Mn\tau)$[11]

$$x = [01, 10]_b \xrightarrow{\Phi} |0110\rangle$$

$N = 2$,
$\tau = 2$



[11] D. Ventura and T. Martinez (1998). *Quantum Associative Memory*. arXiv: quant-ph/9807053 [quant-ph]

Encoding classical data in quantum states

Basis encoding

If basis encoding achieves perfect orthogonality between datapoints, why shouldn't we always use it for classification?

Encoding classical data in quantum states

Time-evolution encoding

Encoding classical data in quantum states

Time-evolution encoding

We're not constrained to *exactly* representing features. For instance, each feature can be the *evolution time of a certain Hamiltonian*

$$\phi(\mathbf{x}) = e^{-iH\mathbf{x}} |0\rangle$$

Encoding classical data in quantum states

Time-evolution encoding

We're not constrained to *exactly* representing features. For instance, each feature can be the *evolution time of a certain Hamiltonian*

$$\phi(\mathbf{x}) = e^{-iH\mathbf{x}} |0\rangle$$

Typically, we do *angle encoding*, i.e. each feature is encoded as a Pauli rotation.

Encoding classical data in quantum states

Time-evolution encoding

We're not constrained to *exactly* representing features. For instance, each feature can be the *evolution time of a certain Hamiltonian*

$$\phi(\mathbf{x}) = e^{-iH\mathbf{x}} |0\rangle$$

Typically, we do *angle encoding*, i.e. each feature is encoded as a Pauli rotation.

Example: one feature, per rotation, per qubit, using Pauli X

$$\phi(\mathbf{x}) = e^{-iXx_{N-1}} \otimes e^{-iXx_{N-2}} \otimes \cdots \otimes e^{-iXx_0} |0\rangle$$

Encoding classical data in quantum states

Time-evolution encoding

For angle encoding:

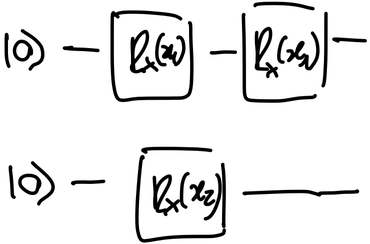
	Single sample	Dataset
num. qubits	N	$N\lceil\log(M)\rceil$
runtime	$O(N)$	$O(MN)$

Encoding classical data in quantum states

Time-evolution encoding

For angle encoding:

	Single sample	Dataset
num. qubits	N	$N\lceil \log(M) \rceil$
runtime	$O(N)$	$O(MN)$



Encoding classical data in quantum states

Hamiltonian encoding of a dataset

[12] S. Lloyd (1996). "Universal Quantum Simulators". In: *Science* 273.5278, pp. 1073–1078. DOI: 10.1126/science.273.5278.1073. eprint: https://www.science.org/doi/pdf/10.1126/science.273.5278.1073

Encoding classical data in quantum states

Hamiltonian encoding of a dataset

If we keep the original data matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$ and we can rewrite it as an Hermitian matrix $H_{\mathbf{X}}$, then we can encode the entire dataset as a Hamiltonian simulation

$$\phi : |0\rangle \to e^{-iH_{\mathbf{X}} t}$$

[12] S. Lloyd (1996). "Universal Quantum Simulators". In: *Science* 273.5278, pp. 1073–1078. DOI: 10.1126/science.273.5278.1073. eprint: https://www.science.org/doi/pdf/10.1126/science.273.5278.1073

Encoding classical data in quantum states

Hamiltonian encoding of a dataset

If we keep the original data matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$ and we can rewrite it as an Hermitian matrix $H_{\mathbf{X}}$, then we can encode the entire dataset as a Hamiltonian simulation

$$\phi : |0\rangle \rightarrow e^{-iH_{\mathbf{X}} t}$$

- The general way to simulate $H_{\mathbf{X}}$ is to use *Trotterization*[12]

[12] S. Lloyd (1996). "Universal Quantum Simulators". In: *Science* 273.5278, pp. 1073–1078. DOI: 10.1126/science.273.5278.1073. eprint: https://www.science.org/doi/pdf/10.1126/science.273.5278.1073

Encoding classical data in quantum states

Hamiltonian encoding of a dataset

If we keep the original data matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$ and we can rewrite it as an Hermitian matrix $H_{\mathbf{X}}$, then we can encode the entire dataset as a Hamiltonian simulation

$$\phi : |0\rangle \to e^{-iH_{\mathbf{X}} t}$$

- The general way to simulate $H_{\mathbf{X}}$ is to use *Trotterization*[12]
  - *Requirement*: we need to know how to decompose $H_{\mathbf{X}}$ in terms that are *easy to simulate*.
  - We can always decompose an Hermitian matrix in $O(MN)$ Pauli strings.

[12] S. Lloyd (1996). "Universal Quantum Simulators". In: *Science* 273.5278, pp. 1073–1078. DOI: 10.1126/science.273.5278.1073. eprint: https://www.science.org/doi/pdf/10.1126/science.273.5278.1073

Encoding classical data in quantum states

Hamiltonian encoding of a dataset

If we keep the original data matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$ and we can rewrite it as an Hermitian matrix $H_{\mathbf{X}}$, then we can encode the entire dataset as a Hamiltonian simulation

$$\phi : |0\rangle \to e^{-iH_{\mathbf{X}}t}$$

- The general way to simulate $H_{\mathbf{X}}$ is to use *Trotterization*[12]
    - *Requirement*: we need to know how to decompose $H_{\mathbf{X}}$ in terms that are *easy to simulate*.
    - We can always decompose an Hermitian matrix in $O(MN)$ Pauli strings.
- More efficient algorithms exist for *sparse* Hamiltonians, but they require an *oracle*

[12] S. Lloyd (1996). "Universal Quantum Simulators". In: *Science* 273.5278, pp. 1073–1078. DOI: 10.1126/science.273.5278.1073. eprint: https://www.science.org/doi/pdf/10.1126/science.273.5278.1073

Encoding classical data in quantum states

Hamiltonian encoding of a dataset

If we keep the original data matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$ and we can rewrite it as an Hermitian matrix $H_{\mathbf{X}}$, then we can encode the entire dataset as a Hamiltonian simulation

$$\phi : |0\rangle \rightarrow e^{-iH_{\mathbf{X}}t}$$

- The general way to simulate $H_{\mathbf{X}}$ is to use *Trotterization*[12]
    - *Requirement*: we need to know how to decompose $H_{\mathbf{X}}$ in terms that are *easy to simulate*.
    - We can always decompose an Hermitian matrix in $O(MN)$ Pauli strings.
- More efficient algorithms exist for *sparse* Hamiltonians, but they require an *oracle*
    - Row i, jth non-zero element

$$|i, j, 0\rangle \rightarrow |i, j, H_{ij}\rangle$$

    - Likely need to hard-code it and full of multi-controlled gates

[12] S. Lloyd (1996). "Universal Quantum Simulators". In: *Science* 273.5278, pp. 1073–1078. DOI: 10.1126/science.273.5278.1073. eprint: https://www.science.org/doi/pdf/10.1126/science.273.5278.1073

Encoding classical data in quantum states

Hamiltonian encoding of a dataset

- Num. qubits: $\log(N)$
- Runtime: $O(MN)$ or $O(\log(MN))$
  - $O(\log(MN))$ only for sparse $\mathbf{X}$ and a few other cases.

Encoding classical data in quantum states

Quantum models as sums of periodic functions

Encoding classical data in quantum states

Quantum models as sums of periodic functions

1. time-evolution embedding: $S(x_i) = e^{iHx_i}$ for feature i.
2. *layered* model circuit

$$U(\mathbf{x}, \theta) = W_{N-1}(\theta_{N-1}) \prod_{i=N-2}^{0} S_i(x_i) W_i(\theta_i)$$

3. $H = \mathrm{diag}\left(\lambda_0^{\mathrm{i}}, \ldots \lambda_{\mathrm{d}-1}^{\mathrm{i}}\right)$

[13] M. Schuld, R. Sweke, and J. J. Meyer (Mar. 2021). "Effect of data encoding on the expressive power of variational quantum-machine-learning models". In: *Physical Review A* 103.3, p. 032430. DOI: 10.1103/physreva.103.032430

Encoding classical data in quantum states

Quantum models as sums of periodic functions

1. time-evolution embedding: $S(x_i) = e^{iHx_i}$ for feature i.
2. *layered* model circuit

$$U(\mathbf{x}, \theta) = W_{N-1}\left(\theta_{N-1}\right) \prod_{i=N-2}^{0} S_i(x_i) W_i\left(\theta_i\right)$$

3. $H = \mathrm{diag}\left(\lambda_0^{\mathrm{i}}, \ldots \lambda_{\mathrm{d}-1}^{\mathrm{i}}\right)$

Then, the quantum model $f(\mathbf{x}, \theta) = \langle \psi\left(\mathbf{x}, \theta\right) | \mathcal{M} | \psi\left(\mathbf{x}, \theta\right) \rangle$ can be expressed as a multidimensional series of periodic functions[13]

$$f(\mathbf{x}, \theta) = \sum_{\omega_0 \in \Omega_0} \cdots \sum_{\omega_{N-1} \in \Omega_{N-1}} c_{\omega_0 \ldots \omega_{N-1}}\left(\theta\right) e^{i\omega_0 x_0} \ldots e^{i\omega_{N-1} x_{N-1}},$$

[13] M. Schuld, R. Sweke, and J. J. Meyer (Mar. 2021). "Effect of data encoding on the expressive power of variational quantum-machine-learning models". In: *Physical Review A* 103.3, p. 032430. DOI: 10.1103/physreva.103.032430

Encoding classical data in quantum states

Quantum models as sums of periodic functions

1. time-evolution embedding: $S(x_i) = e^{iHx_i}$ for feature i.
2. *layered* model circuit

$$U(\mathbf{x}, \theta) = W_{N-1}(\theta_{N-1}) \prod_{i=N-2}^{0} S_i(x_i) W_i(\theta_i)$$

3. $H = \mathrm{diag}\left(\lambda_0^{\mathrm{i}}, \ldots \lambda_{\mathrm{d}-1}^{\mathrm{i}}\right)$

Then, the quantum model $f(\mathbf{x}, \theta) = \langle \psi(\mathbf{x}, \theta) | \mathcal{M} | \psi(\mathbf{x}, \theta) \rangle$ can be expressed as a multidimensional series of periodic functions[13]

$$f(\mathbf{x}, \theta) = \sum_{\omega_0 \in \Omega_0} \cdots \sum_{\omega_{N-1} \in \Omega_{N-1}} c_{\omega_0 \ldots \omega_{N-1}}(\theta) \, e^{i\omega_0 x_0} \ldots e^{i\omega_{N-1} x_{N-1}},$$

where **the frequencies are completely determined by the eigenvalues of the data-encoding Hamiltonian**

$$\Omega_i = \left\{ \lambda_s^i - \lambda_t^i | s, t \in \{1, \ldots d\} \right\}$$

[13] M. Schuld, R. Sweke, and J. J. Meyer (Mar. 2021). "Effect of data encoding on the expressive power of variational quantum-machine-learning models". In: *Physical Review A* 103.3, p. 032430. DOI: 10.1103/physreva.103.032430

Encoding classical data in quantum states

Quantum models as sums of periodic functions

Furthermore, if λ_s^i, λ_t^i are integer (s.a. for Pauli gates), $f(\mathbf{x}, \theta)$ becomes a *Fourier series*

$$f(\mathbf{x}, \theta) = \sum_{n_0 \in \Omega_0} \cdots \sum_{n_{N-1} \in \Omega_{N-1}} c_{n_0, \ldots n_{N-1}}(\theta)\, e^{i n_0 x_0} \ldots e^{i n_{N-1} x_{N-1}}$$

Encoding classical data in quantum states

Quantum models as sums of periodic functions

Furthermore, if λ_s^i, λ_t^i are integer (s.a. for Pauli gates), $f(\mathbf{x}, \theta)$ becomes a *Fourier series*

$$f(\mathbf{x}, \theta) = \sum_{n_0 \in \Omega_0} \cdots \sum_{n_{N-1} \in \Omega_{N-1}} c_{n_0, \ldots n_{N-1}}(\theta)\, e^{i n_0 x_0} \ldots e^{i n_{N-1} x_{N-1}}$$

In particular, by repeating the encoding of a feature, we can obtain multiple of the period.

For instance, if two of the features are the same, then $x_i = x_j = x$

$$e^{i n_i x_i} e^{i n_j x_j} = e^{i\left(n_i + n_j\right) x}$$

Encoding classical data in quantum states

Quantum models as sums of periodic functions

Furthermore, if λ_s^i, λ_t^i are integer (s.a. for Pauli gates), $f(\mathbf{x}, \theta)$ becomes a *Fourier series*

$$f(\mathbf{x}, \theta) = \sum_{n_0 \in \Omega_0} \cdots \sum_{n_{N-1} \in \Omega_{N-1}} c_{n_0, \dots n_{N-1}}(\theta)\, e^{i n_0 x_0} \dots e^{i n_{N-1} x_{N-1}}$$

In particular, by repeating the encoding of a feature, we can obtain multiple of the period.
For instance, if two of the features are the same, then $x_i = x_j = x$

$$e^{i n_i x_i} e^{i n_j x_j} = e^{i\left(n_i + n_j\right) x}$$

From this observation and the universality of Fourier series follows the *universality of quantum models with time evolution encoding*.

Practical matters

Access to Fourier coefficients in time evolution encoding

What about the coefficients $c_{n_0,\dots n_{N-1}}(\theta)$?

Practical matters

Access to Fourier coefficients in time evolution encoding

What about the coefficients $c_{n_0,\dots n_{N-1}}(\theta)$?

- All the elements of the circuit model determine them (data encoding, ansatz, measurement).

Practical matters

Access to Fourier coefficients in time evolution encoding

What about the coefficients $c_{n_0,\ldots n_{N-1}}(\theta)$?

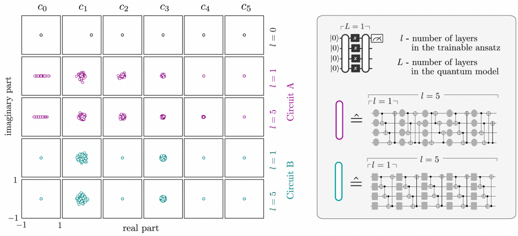
- All the elements of the circuit model determine them (data encoding, ansatz, measurement).
- No matter the frequency that we encode for x_i, if some $c_{n_i} = 0$, then we cannot *access* those frequencies.

Practical matters

Access to Fourier coefficients in time evolution encoding

What about the coefficients $c_{n_0,\ldots n_{N-1}}(\theta)$?

- All the elements of the circuit model determine them (data encoding, ansatz, measurement).
- No matter the frequency that we encode for x_i, if some $c_{n_i} = 0$, then we cannot *access* those frequencies.
- How each element affects each coefficient is a matter of active research.



Practical matters

Gradients

Computing gradients is done with the *parameter shift rule* (PSR). E.g. for Pauli gates:

$$\frac{\partial f}{\partial \theta_i} = \frac{f(\theta_i + s) - f(\theta_i - s)}{2}$$

Practical matters

Gradients

Computing gradients is done with the *parameter shift rule* (PSR). E.g. for Pauli gates:

$$\frac{\partial f}{\partial \theta_i} = \frac{f(\theta_i + s) - f(\theta_i - s)}{2}$$

How many *model evaluations* we need to compute the *full gradient*?

Practical matters

Gradients

Computing gradients is done with the *parameter shift rule* (PSR). E.g. for Pauli gates:

$$\frac{\partial f}{\partial \theta_i} = \frac{f(\theta_i + s) - f(\theta_i - s)}{2}$$

How many *model evaluations* we need to compute the *full gradient*?

backpropagation	2
PSR	$2 n_{\text{params}} n_{\text{shots}}$

Practical matters

Gradients

Is this a fair comparison?

Practical matters

Gradients

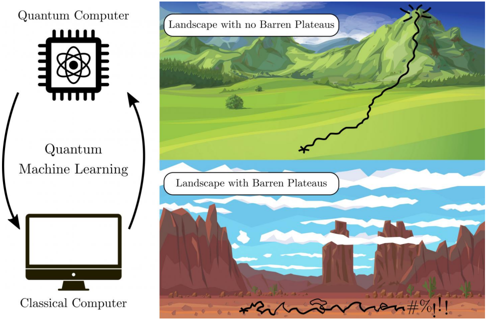
Is this a fair comparison?

Not really...

	n. steps	$O(N)$ linear algebra	parallelization	optimized libraries
Classical	$+$	-	$++$	$++$
Quantum	$-$	$+$	$+$	$+$

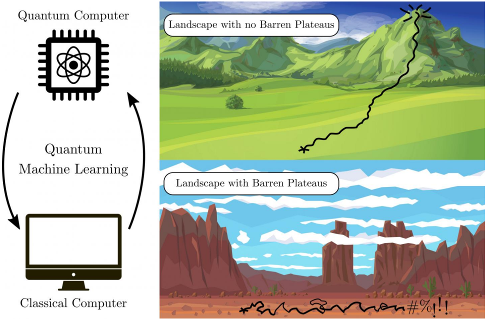
Practical matters

Barren plateaus



Practical matters

Barren plateaus



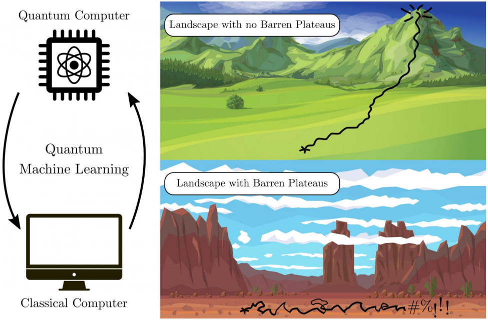
$$\mathrm{Var}\left[\partial_{\theta_i} C(\theta)\right] \leq F(n),$$

$$F(n) \in O\left(\frac{1}{b^n}\right), \; b > 1$$

[14] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven (Nov. 2018). "Barren plateaus in quantum neural network training landscapes". In: *Nature Communications* 9.1. DOI: 10.1038/s41467-018-07090-4

Practical matters

Barren plateaus



$$\mathrm{Var}\left[\partial_{\theta_i} C(\theta)\right] \leq F(n),$$

$$F(n) \in O\left(\frac{1}{b^n}\right), \; b > 1$$

- Different reasons, the main one is the ansatz $W(\theta)$
- *Problem-agnostic* ansatze suffer from BPs with high n and high depths[14]

[14] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven (Nov. 2018). "Barren plateaus in quantum neural network training landscapes". In: *Nature Communications* 9.1. DOI: 10.1038/s41467-018-07090-4

Links for demos

https://colab.research.google.com/github/debrevitatevitae/LabSessionQML/blob/main/variational-classifier-parity.ipynb

Bibliography I

Biamonte, J., P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd (Sept. 2017). "Quantum machine learning". In: *Nature* 549.7671, pp. 195–202. DOI: 10.1038/nature23474.

Liu, Y., S. Arunachalam, and K. Temme (July 2021). "A rigorous and robust quantum speed-up in supervised machine learning". In: *Nature Physics* 17.9, pp. 1013–1017. DOI: 10.1038/s41567-021-01287-z.

Lloyd, S. (1996). "Universal Quantum Simulators". In: *Science* 273.5278, pp. 1073–1078. DOI: 10.1126/science.273.5278.1073. eprint: https://www.science.org/doi/pdf/10.1126/science.273.5278.1073.

McClean, J. R., S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven (Nov. 2018). "Barren plateaus in quantum neural network training landscapes". In: *Nature Communications* 9.1. DOI: 10.1038/s41467-018-07090-4.

Mottonen, M., J. J. Vartiainen, V. Bergholm, and M. M. Salomaa (2004). *Transformation of quantum states using uniformly controlled rotations*. arXiv: quant-ph/0407010 [quant-ph].

Bibliography II

Nielsen, M. A. (2018). *Neural Networks and Deep Learning*. misc.

Schuld, M. and N. Killoran (July 2022). "Is Quantum Advantage the Right Goal for Quantum Machine Learning?" In: *PRX Quantum* 3.3. DOI: 10.1103/prxquantum.3.030101.

Schuld, M., R. Sweke, and J. J. Meyer (Mar. 2021). "Effect of data encoding on the expressive power of variational quantum-machine-learning models". In: *Physical Review A* 103.3, p. 032430. DOI: 10.1103/physreva.103.032430.

Ventura, D. and T. Martinez (1998). *Quantum Associative Memory*. arXiv: quant-ph/9807053 [quant-ph].