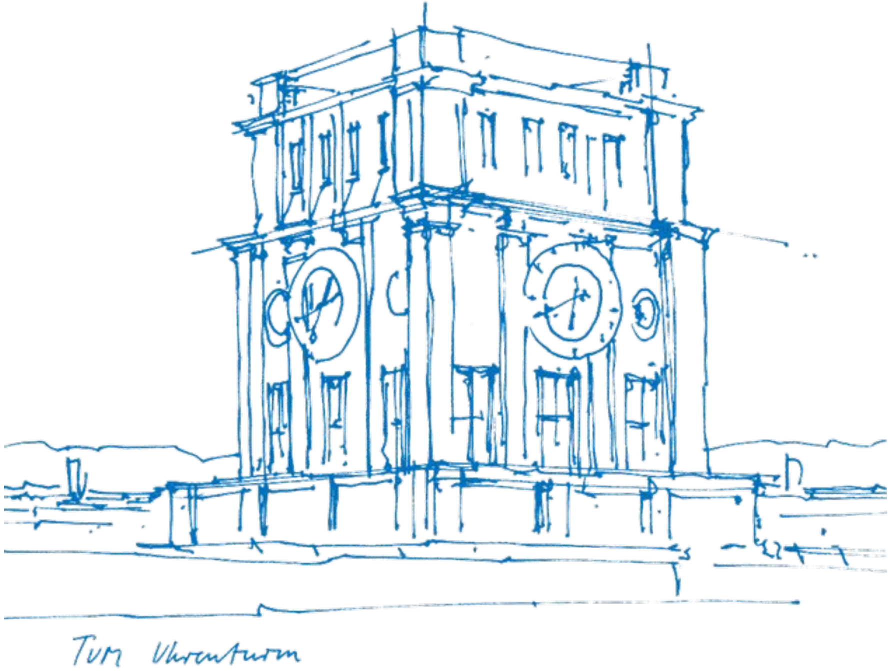
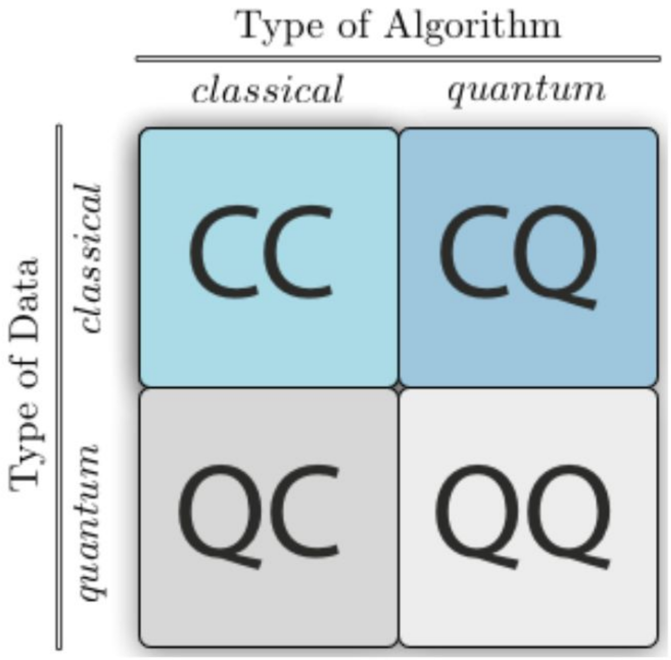
Quantum Autoencoders

Seminar: Advanced Topics in Quantum Computing

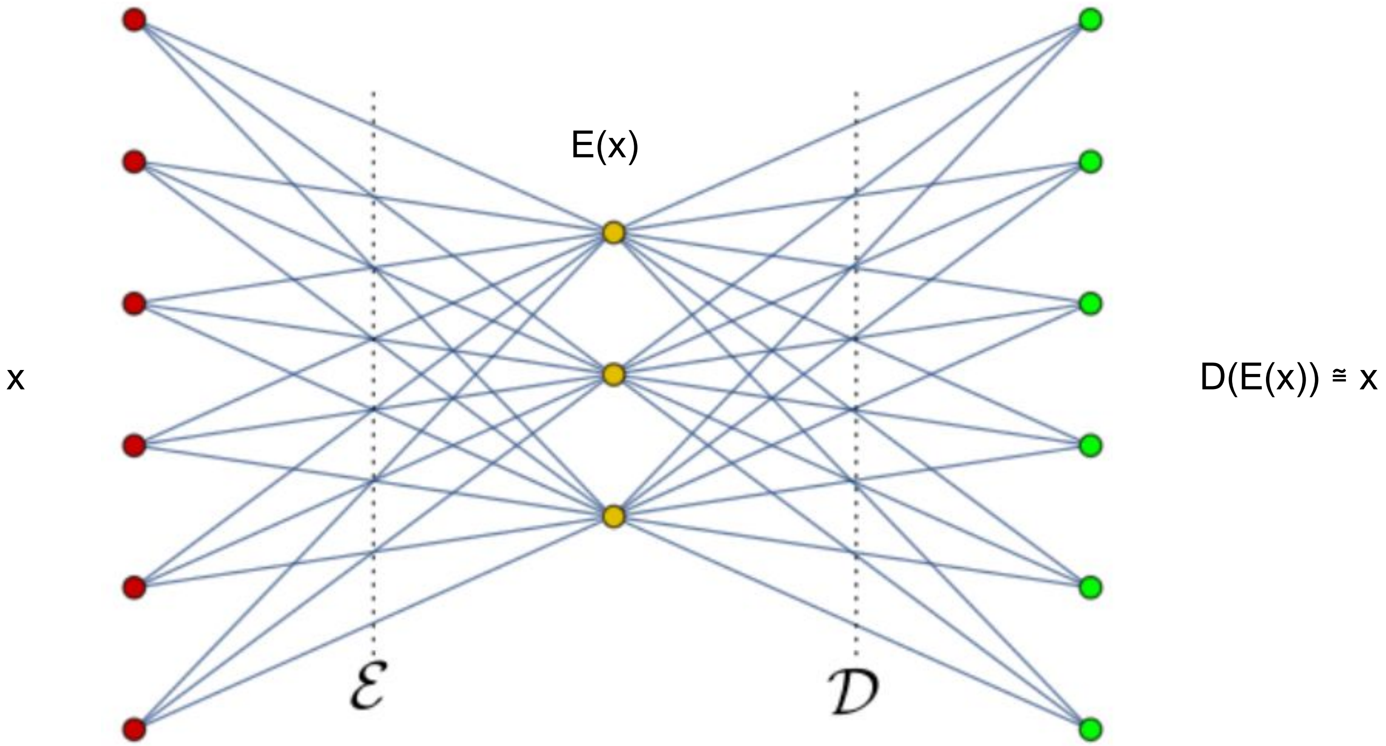


Theodora-Augustina Drăgan
M. Sc. Informatics
10th June 2021

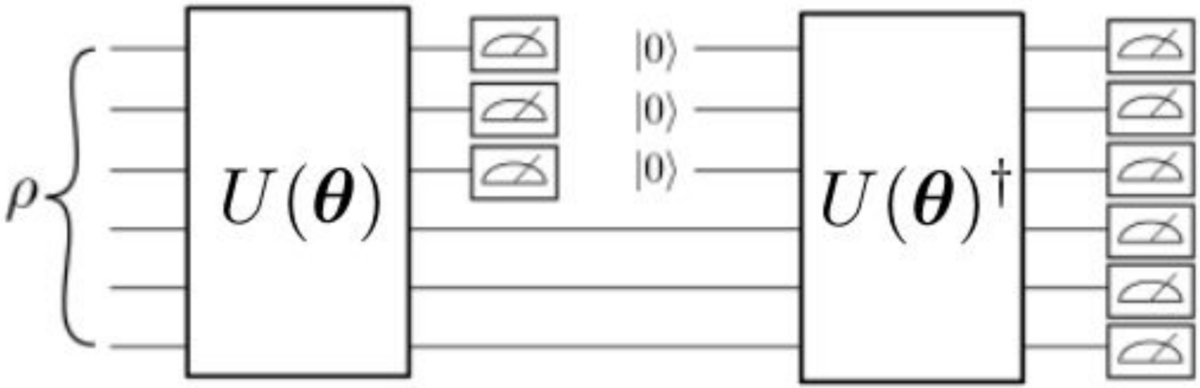
Quantum Machine Learning - Types[1]



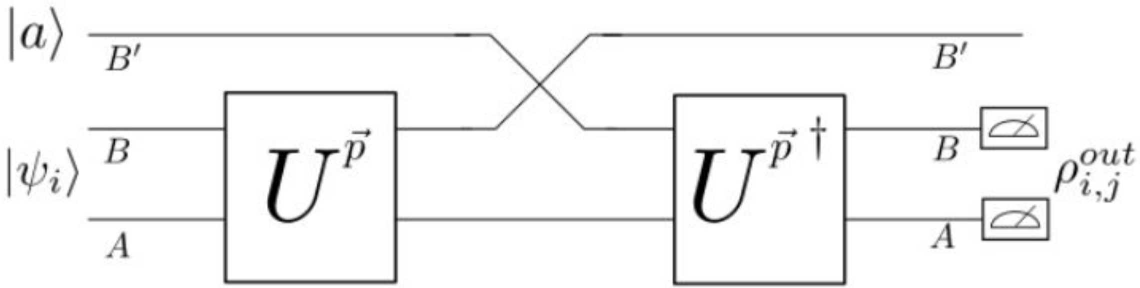
Autoencoders[2]



Quantum Autoencoders

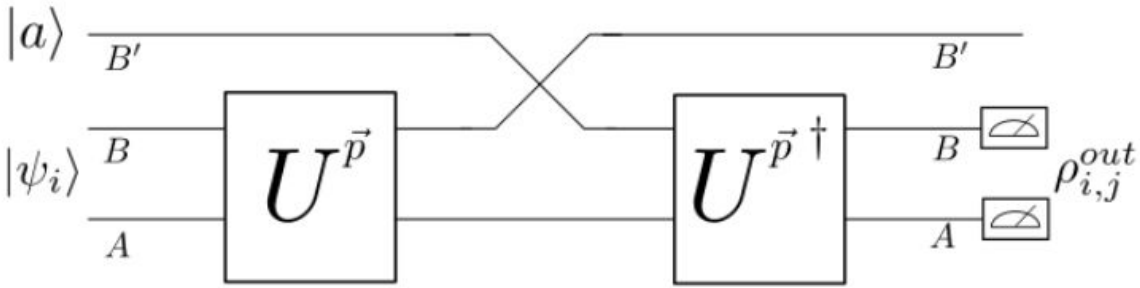


Quantum Autoencoder Circuit



- input data: $\{|\psi_i\rangle_{AB}\}$ is an ensemble of states on n+k qubits, where subsystems A and B are comprised of n and k qubits
- evaluating the performance: measuring the deviation from the initial input state $|\psi_i\rangle$ to the output ρ_{out}^{i}, for which we compute the expected fidelity $F(|\psi_i\rangle,\rho_{out}^{i}) = \langle\psi_i|\rho_{out}^{i}|\psi_i\rangle$
- for a successful autoencoding, $F(|\psi_i\rangle,\rho_{out}^{i}) \approx 1$
- $\{U^p\}$ is a family of unitary operators acting on n+k qubits, where $p=\{p_1,p_2,\ldots\}$ is some set of parameters defining a unitary quantum circuit

Quantum Autoencoder Circuit : Efficient testing



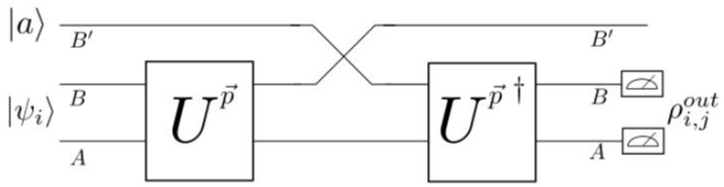
- We introduce $|a\rangle_{B'}$ is a fixed pure reference state of k qubits

- We wish to find the unitary U^p which maximizes the average fidelity: $C_1(\vec{p}) = \sum_i p_i \cdot F(|\psi_i\rangle, \rho_{i,\vec{p}}^{out})$ also named the cost function,

  where $\rho_{i,\vec{p}}^{out} = (U^{\vec{p}})^\dagger_{AB'} \operatorname{Tr}_B \left[U^{\vec{p}}_{AB} \left[\psi_{i\,AB} \otimes a_{B'} \right] (U^{\vec{p}}_{AB})^\dagger \right] (U^{\vec{p}})_{AB'}$

- Last step, we introduce the SWAP operation

Quantum Autoencoder Circuit



Proof why we can do the SWAP Test:

$$
\begin{aligned}
&F(|\psi_i\rangle_{AB} \otimes |a\rangle_{B'}, U_{AB}^{\dagger} V_{BB'} U_{AB} |\psi_i\rangle_{AB} \otimes |a\rangle_{B'}) = \\
&F(U_{AB} |\psi_i\rangle_{AB} \otimes |a\rangle_{B'}, V_{BB'} U_{AB} |\psi_i\rangle_{AB} \otimes |a\rangle_{B'}) = \\
&F(|\psi_i'\rangle_{AB} \otimes |a\rangle_{B'}, V_{BB'} |\psi_i'\rangle_{AB} \otimes |a\rangle_{B'}) = \\
&F(|\psi_i'\rangle_{AB} \otimes |a\rangle_{B'}, |\psi_i'\rangle_{AB'} \otimes |a\rangle_{B}),
\end{aligned}
$$

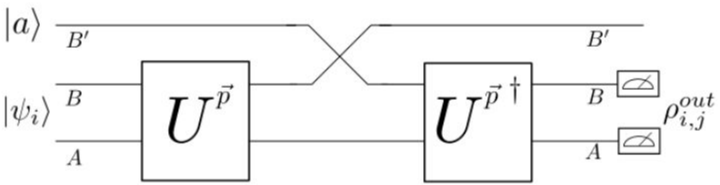
where $U|\psi_i\rangle = |\psi_i'\rangle$

The terms in the cost function can be found by tracing over the B' system:

$$
\begin{aligned}
&F(\mathrm{Tr}_{B'}[\psi'_{i\,AB} \otimes a'_{B}], \mathrm{Tr}_{B'}[\psi'_{i\,AB'} \otimes a_{B}]) \\
&F(\psi'_{i\,AB}, \rho'_{A} \otimes a_{B}),
\end{aligned}
$$

where $\rho_A' = \mathrm{Tr}_{B'} |\psi_i'\rangle\langle\psi_i'|_{AB'}]$

Quantum Autoencoder Circuit



Finally, let us consider tracing instead over the AB system and looking at the “trash” system B’:

$$F(\mathrm{Tr}_{AB}\left[\psi'_{i\,AB} \otimes a'_{B}\right], \mathrm{Tr}_{AB}\left[\psi'_{i\,AB'} \otimes a_{B}\right]) = F(|a\rangle_{B'}, \rho'_{B'}),$$

$$\text{where } \rho'_{B'} = \mathrm{Tr}_A[|\psi'_i\rangle \langle\psi'_i|_{AB'}]$$

Perfect fidelity is obtained if for any input state:

$$U|\psi_i\rangle_{AB} = |\psi_i^c\rangle_A \otimes |a\rangle_B \;,$$ where $|\psi_i^c\rangle_A$ is a compression of the input state

This is because if the B and B’ system are identical when the swap occurs, the entire circuit will be reduced to the identity map. However, this happens exactly when the trash state, the dismissed qubits, is equal to the reference state.

Quantum Autoencoder Circuit

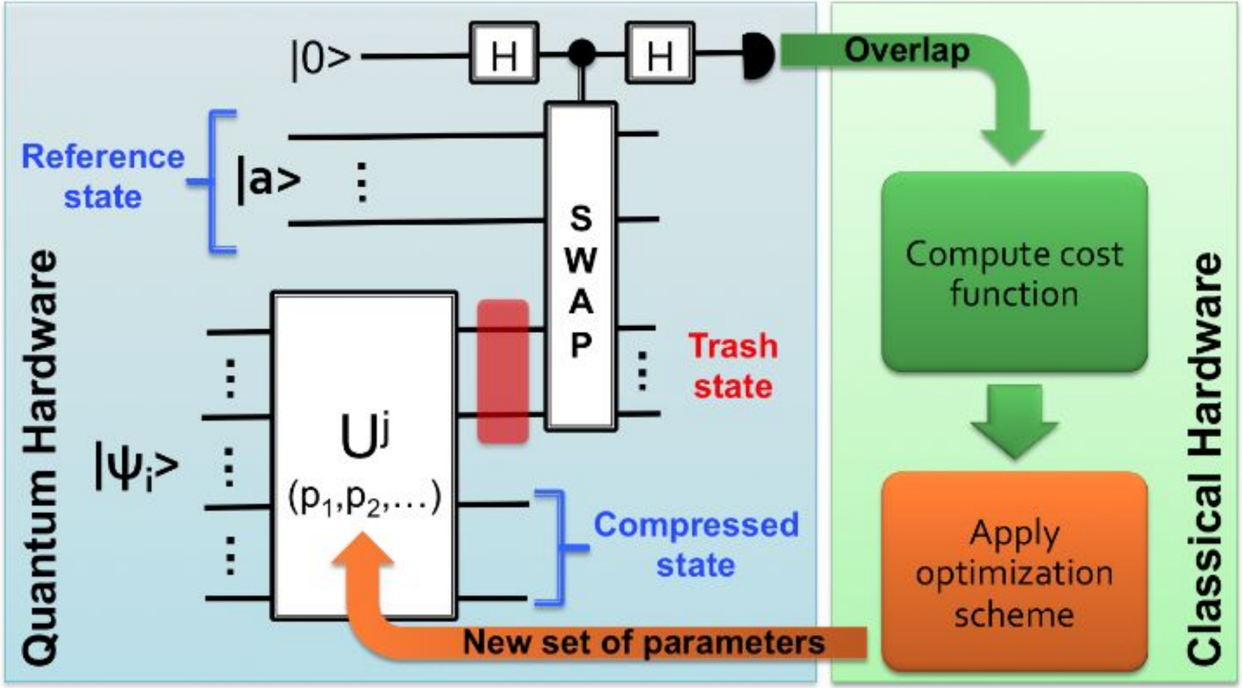
Thus it becomes possible to learn the optimal parameters p for the U^p encoder by training only on the trash state. The previous equation:

$$U|\psi_i\rangle_{AB} = |\psi_i^c\rangle_A \otimes |a\rangle_B$$

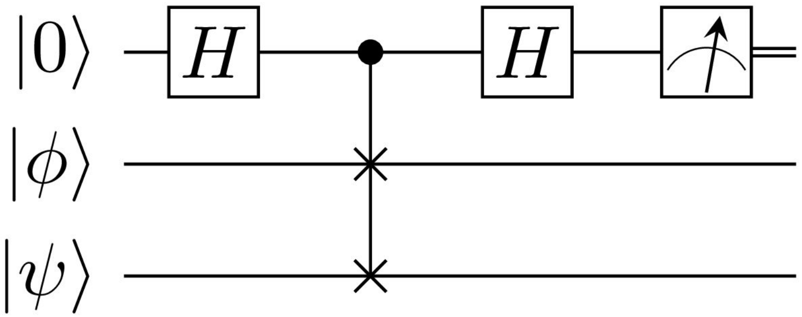
is completely independent of $U^\dagger$, and thus we can describe a second cost function, that will actually be used during training, defined in terms of trash state fidelity:

$$C_2(\vec{p}) = \sum_i p_i \cdot F(\mathrm{Tr}_A \left[U^{\vec{p}} |\psi_i\rangle \langle\psi_i|_{AB} (U^{\vec{p}})^\dagger\right], |a\rangle_B)$$

Testing the outcome of the QAE: The SWAP Test



The SWAP Test



Consider two states: $|\phi\rangle$ and $|\psi\rangle$.

The state of the system at the beginning of the circuit is $|0, \phi, \psi\rangle$.

After the Hadamard gate, the state of the system is $\frac{1}{\sqrt{2}}(|0, \phi, \psi\rangle + |1, \phi, \psi\rangle)$

The controlled SWAP gate transforms the state into $\frac{1}{\sqrt{2}}(|0, \phi, \psi\rangle + |1, \psi, \phi\rangle)$

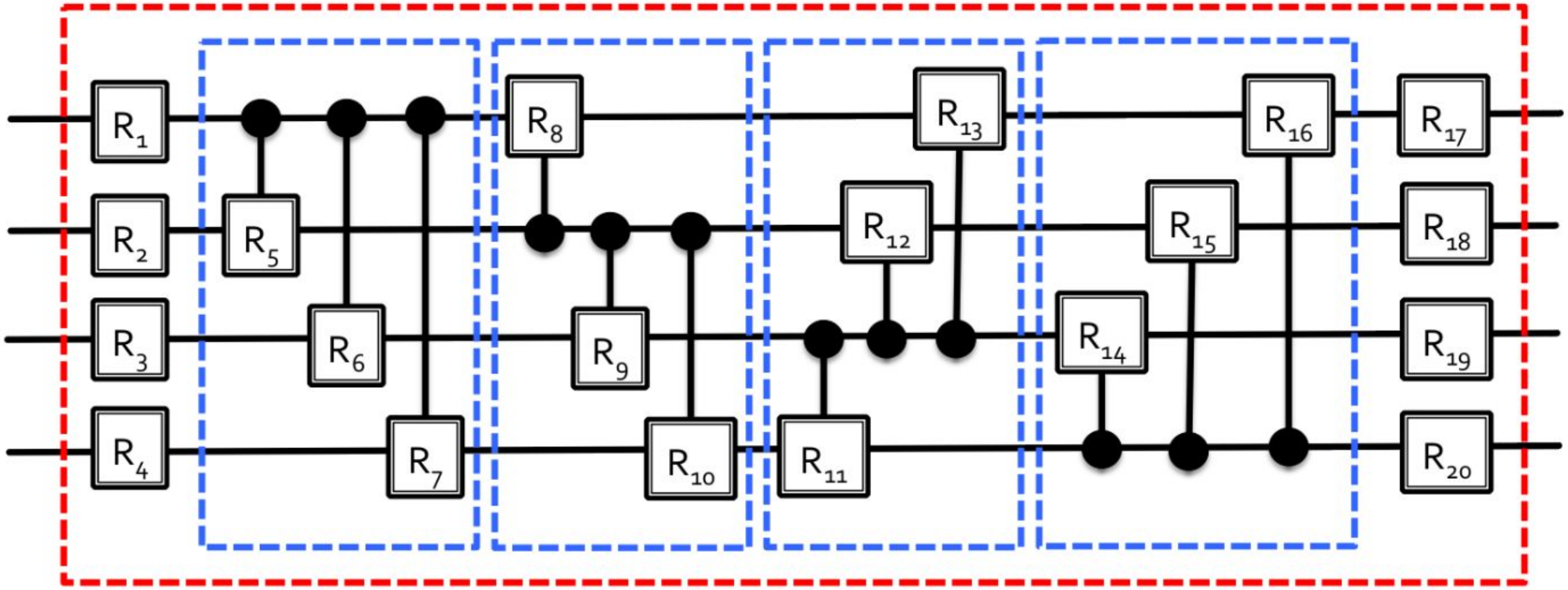
The second Hadamard gate results in

$$\frac{1}{2}(|0, \phi, \psi\rangle + |1, \phi, \psi\rangle + |0, \psi, \phi\rangle - |1, \psi, \phi\rangle) = \frac{1}{2}|0\rangle(|\phi, \psi\rangle + |\psi, \phi\rangle) + \frac{1}{2}|1\rangle(|\phi, \psi\rangle - |\psi, \phi\rangle)$$

The Measurement gate on the first qubit ensures that it's 0 with a probability of:

$$P(\text{First qubit} = 0) = \frac{1}{2}\Big(\langle\phi|\langle\psi| + \langle\psi|\langle\phi|\Big)\frac{1}{2}\Big(|\phi\rangle|\psi\rangle + |\psi\rangle|\phi\rangle\Big) = \frac{1}{2} + \frac{1}{2}|\langle\psi|\phi\rangle|^2$$

The Programmable Encoder Circuit



Dataset Preprocessing

- Used the MNIST dataset (pictures are of handwritten digits), which contains 60.000 pictures of 28 x 28 pixels
- Resized the selected pictures to a smaller dimension (2 x {2, 3, 4} pixels) and then flatten the values to obtain the values $[b_1, .., b_n]$, where n is 4, 6 or 8
- In order to transform them to quantum bits, they are encoded using rotational gates (here chosen the RX gate)
- The final qubit register is :

  $|0\ 0..0\ q_1 ... q_n\rangle$

- We have one leading $|0\rangle$, which is the ancilla qubit for the SWAP test, afterwards k times $|0\rangle$as the reference state, k being equal to the number of discarded qubits after the compression, and q_i = $RX(b_i)$, the quantum angle embedding of the input state

Technological Stack

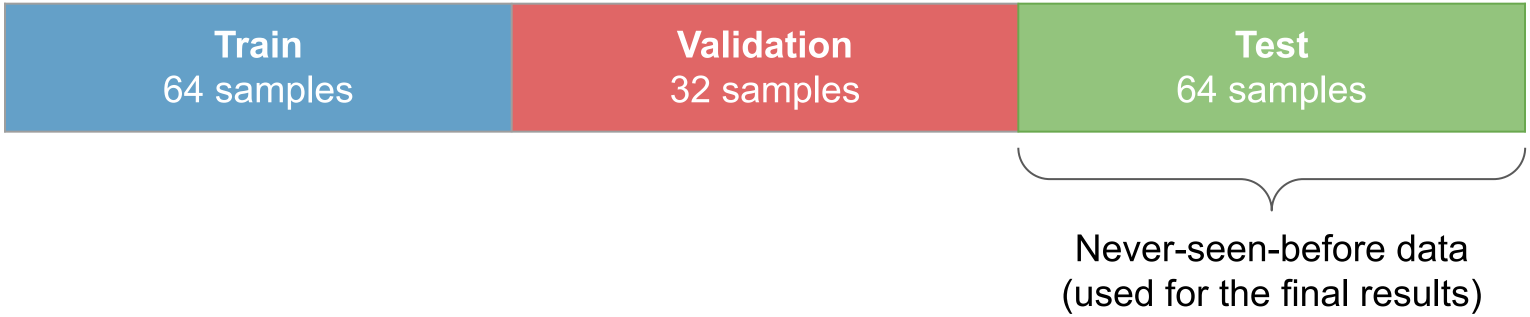
PENNYLANE

a cross-platform Python library used for the backend of the quantum circuit simulation, which enables computations such as using gradient descent [3]

PyTorch

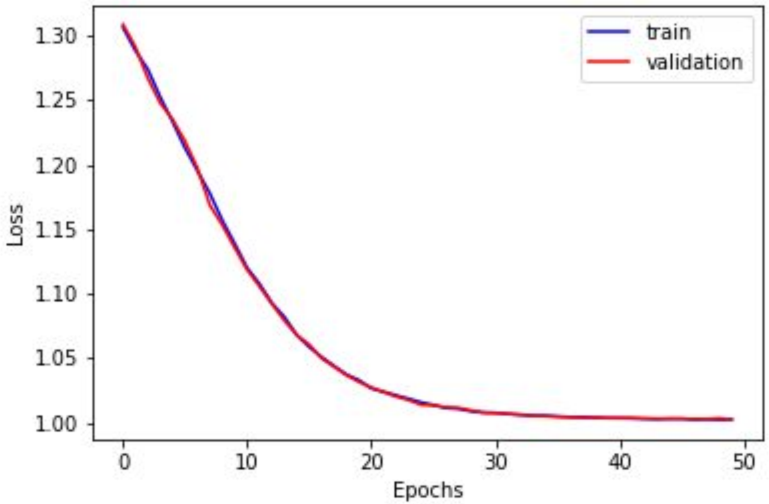
an open source machine learning library based on the Torch library, used here for the classical optimization of the parameters of the quantum gates

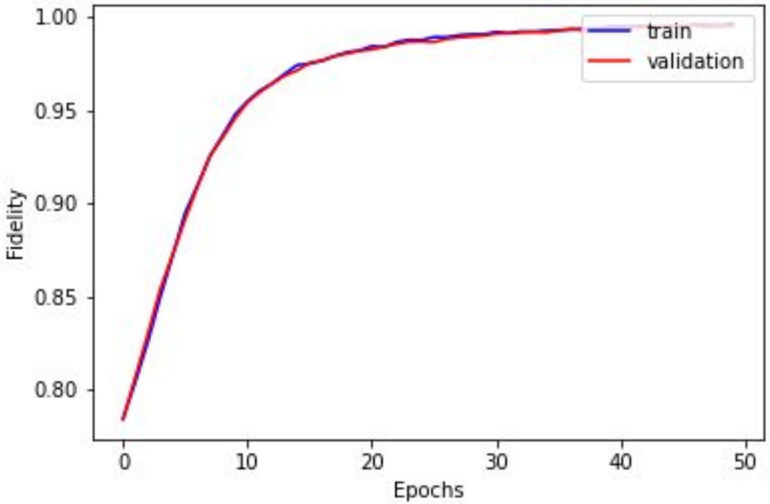
Dataset Splitting and Hyperparameters



- the loss presented in the paper: loss = $-\log_{10}(1\text{-fidelity})$
- chosen loss to minimise: loss = 1 / fidelity
- number of samples in a batch: 8
- optimizer: Adam
- learning rate: 0.003

Results (4 → 2 qubits compression)





Results

Number of input qubits	Number of qubits after compression	Fidelity	Time to train and test
4	3	0.997	15 mins
4	2	0.989	
6	5	0.996	50 mins
6	4	0.985	
6	3	0.902	
8	6	0.939	6 hours

Thank you for your attention!

Questions?

References

[1] Aïmeur, E., Brassard, G., & Gambs, S. (2006, June). Machine learning in a quantum world. In *Conference of the Canadian Society for Computational Studies of Intelligence* (pp. 431-442). Springer, Berlin, Heidelberg.

[2] Romero, J., Olson, J. P., & Aspuru-Guzik, A. (2017). Quantum autoencoders for efficient compression of quantum data. *Quantum Science and Technology*, *2*(4), 045001. (Paper the implementation is based on)

[3] Bergholm, V., Izaac, J., Schuld, M., Gogolin, C., Alam, M. S., Ahmed, S., ... & Killoran, N. (2018). Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968*.