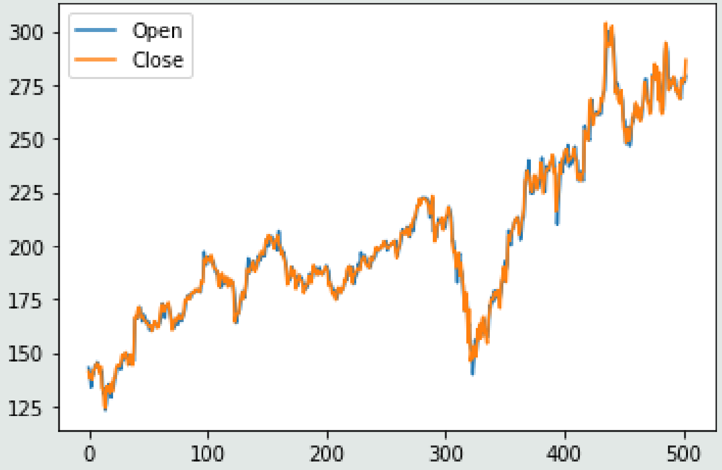
Training Deep Quantum Neural Networks

Alexandra-Maria **DOBRESCU**

MSc AI 1st Year UPB

doi.org/10.1038/s41467-020-14454-2

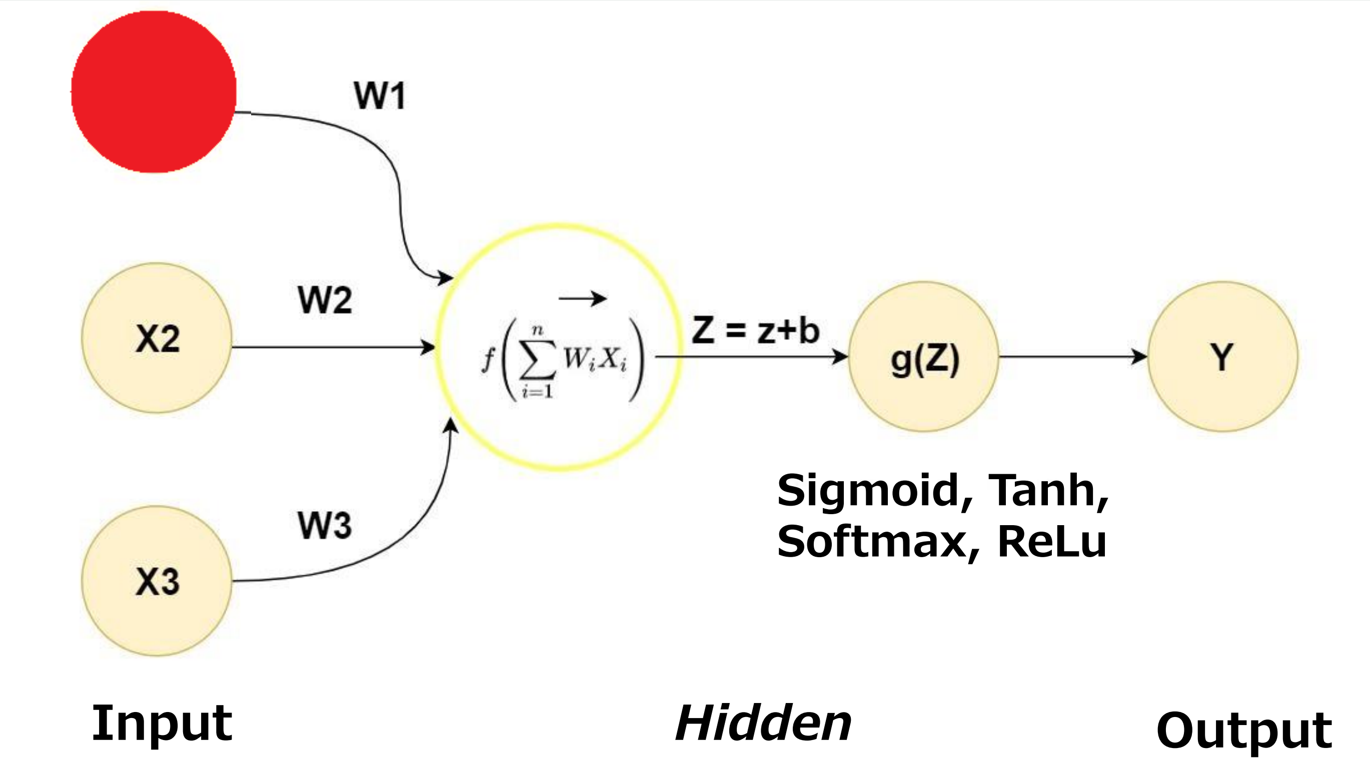


Overview
Why are Neural Networks important?

Used both in research and industry to solve complex problems to improve the decision process.

Classical Neural Networks

The perceptron



Feed Forward Neural Network

Quantum Machine Learning

QC

Classical ML → Improve Quantum Tasks:

- Simulation of many body systems [2]
- Adaptive Quantum Computation [3]
- Quantum metrology [4]

Quantum algorithms
& Classical Data:

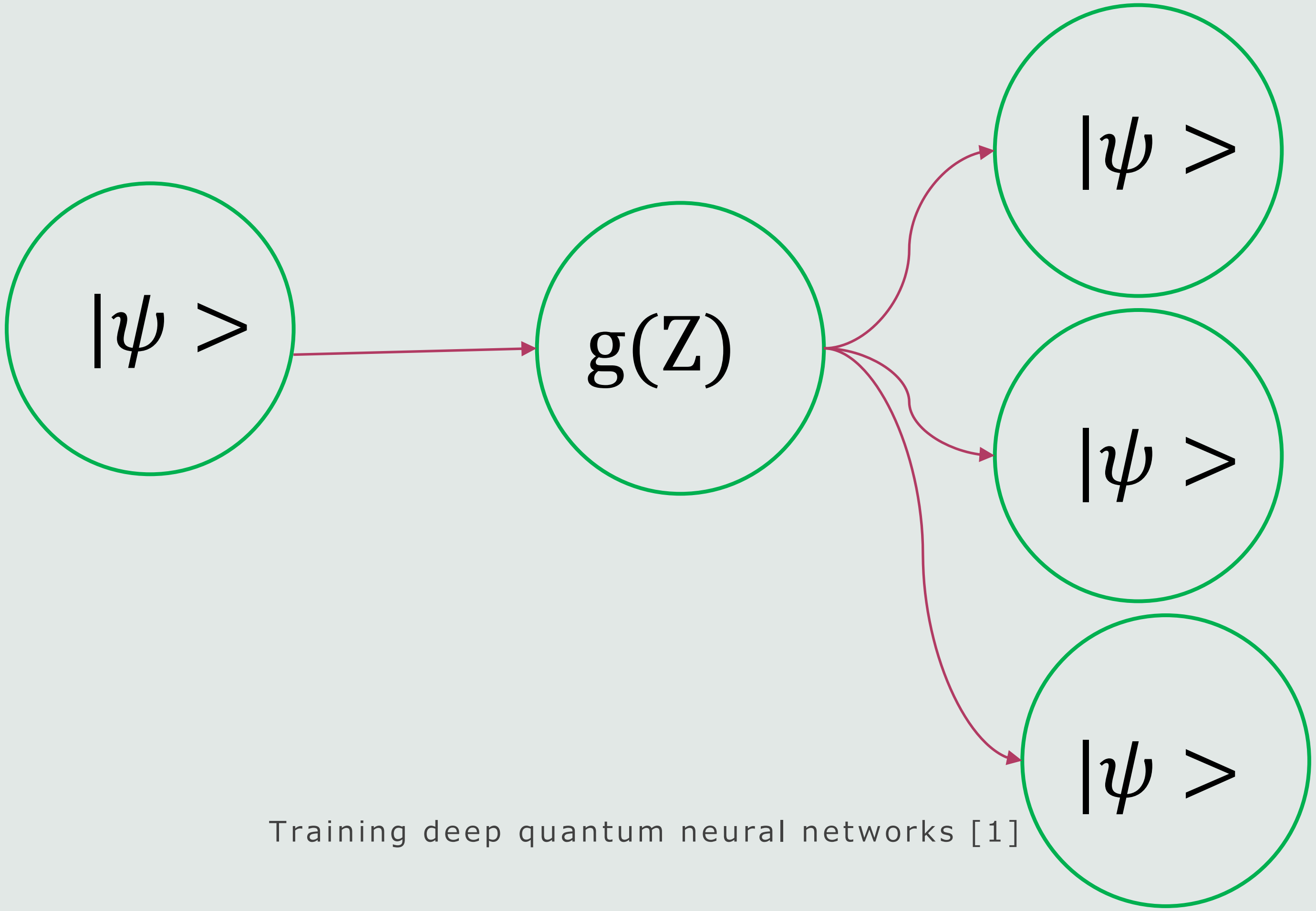
- Speed Up classical ML
- Link to quantum states?

CQ

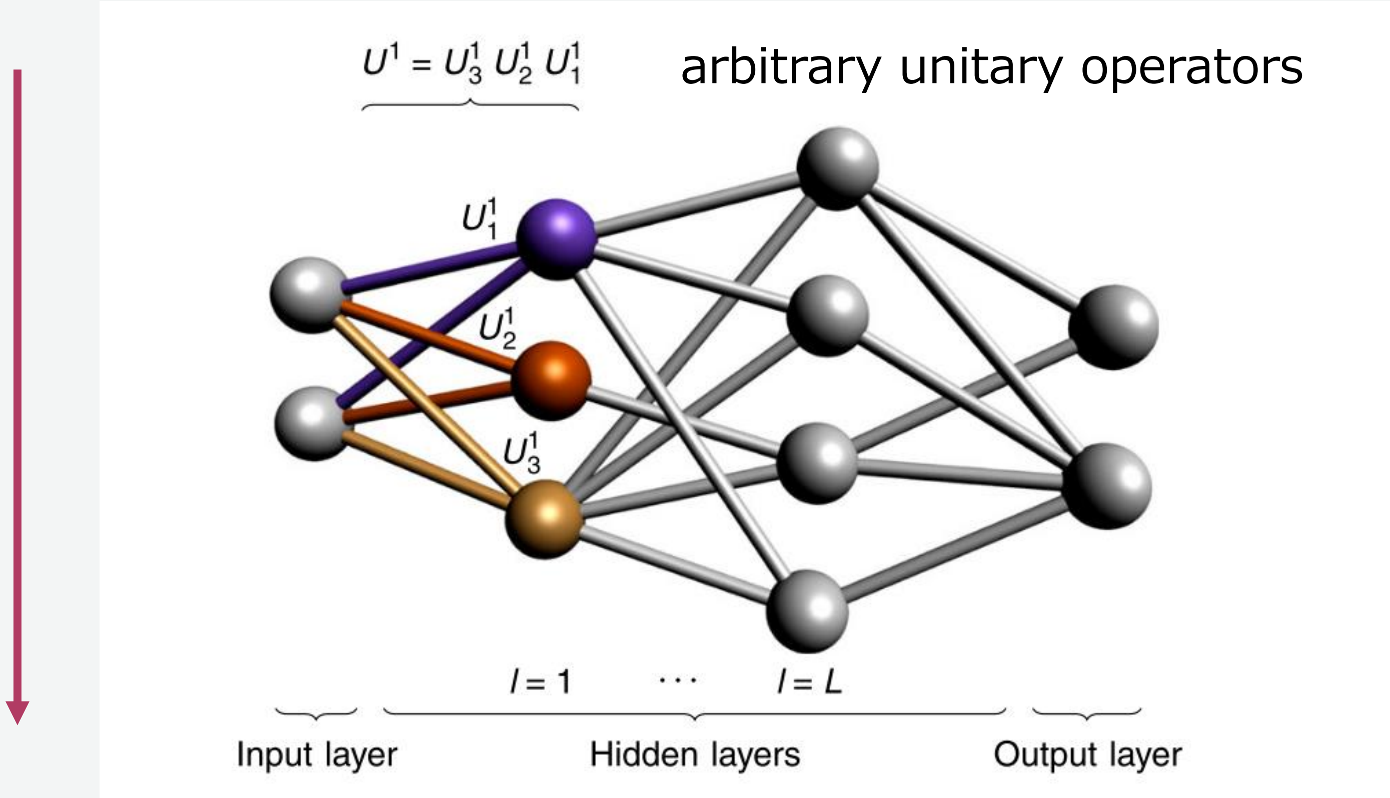
Quantum Computing Devices → **QQ** → Learning Tasks with Quantum Data

"quantum learning of parametrised unitary operations" [5]

Quantum generalisation of the perceptron



X No-cloning theorem!



The quantum architecture

quantum circuit of quantum perceptrons

How is the information processed?

The training algorithm [6] – part 1

Step 1: ***Initialisation procedure***

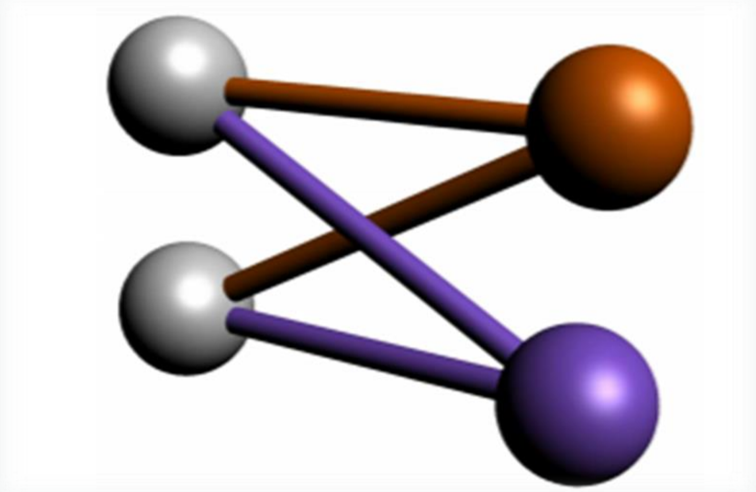
- State $s = 0$
- Choose $U_1^{\text{out}}(0)$ and $U_2^{\text{out}}(0)$ randomly

Step 2: ***Feedforward***

- Input state: $|\Phi_x^{in}> = |\phi_x^{in}> \otimes |00>_{out}$
- Unitaries to the input state:

$$|\psi_x>= U_2^{\text{out}}(s)U_1^{\text{out}}(0)|\Phi_x>$$

- Trace out: $\rho_x^{\theta}(s) = tr_{\text{in}}(|\psi_x> < \psi_x|)$



Task: Learn an unknown unitary V
Set of training data:
N pairs $(|\phi_x^{in}>, V|\phi_x^{out}>)$

How is the information processed?

The training algorithm [6] – part 2

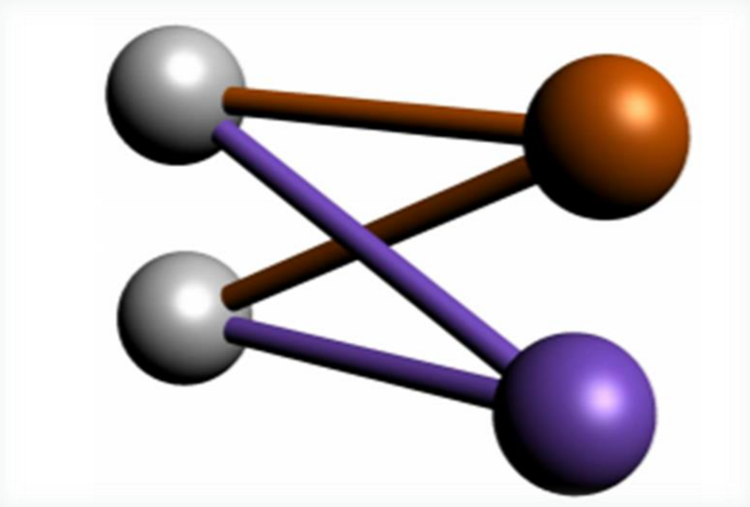
Step 3: *Update parameters*

- Update each perceptron unitary:

$$U_j^l(s+\varepsilon) = \mathrm{e}^{\mathrm{i}\varepsilon k_j^l(s)} U_j^l(s)$$

- The cost function: *Fidelity* - an essentially unique measure of closeness for pure quantum states.

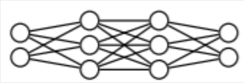
$$C(s) = \frac{1}{N} \sum_{x=1}^{N} < \psi_x | \rho_{x(s)}^{out} | \psi_x >$$



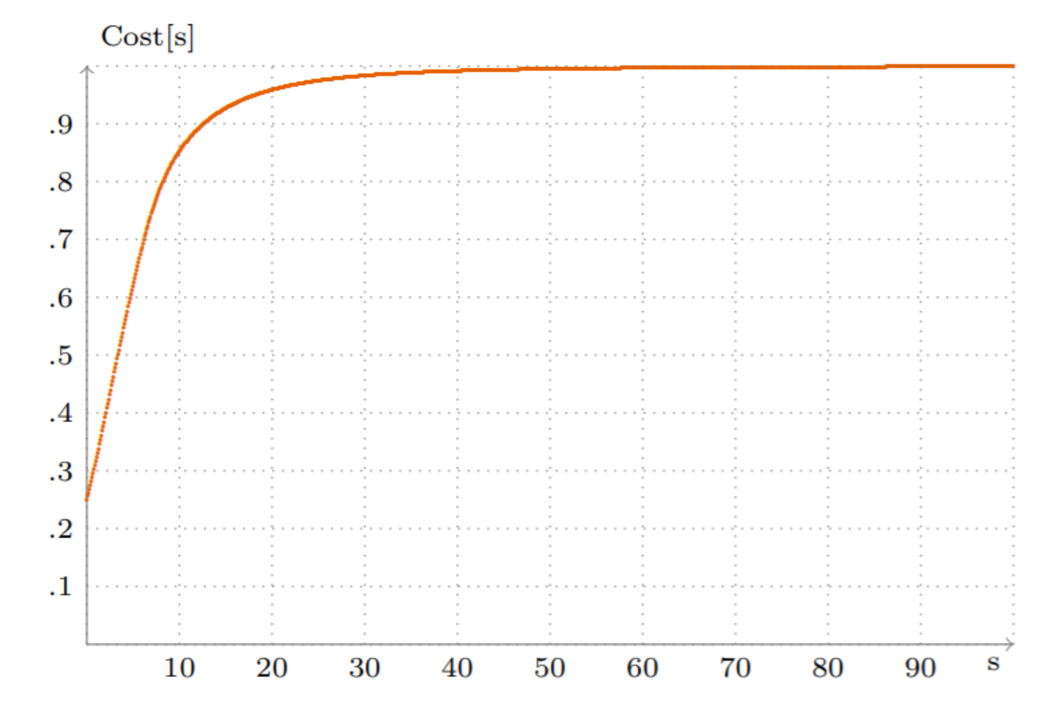
Classical case: *minimise* the cost function

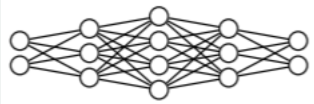
Quantum case: *maximise* the fidelity

Learning

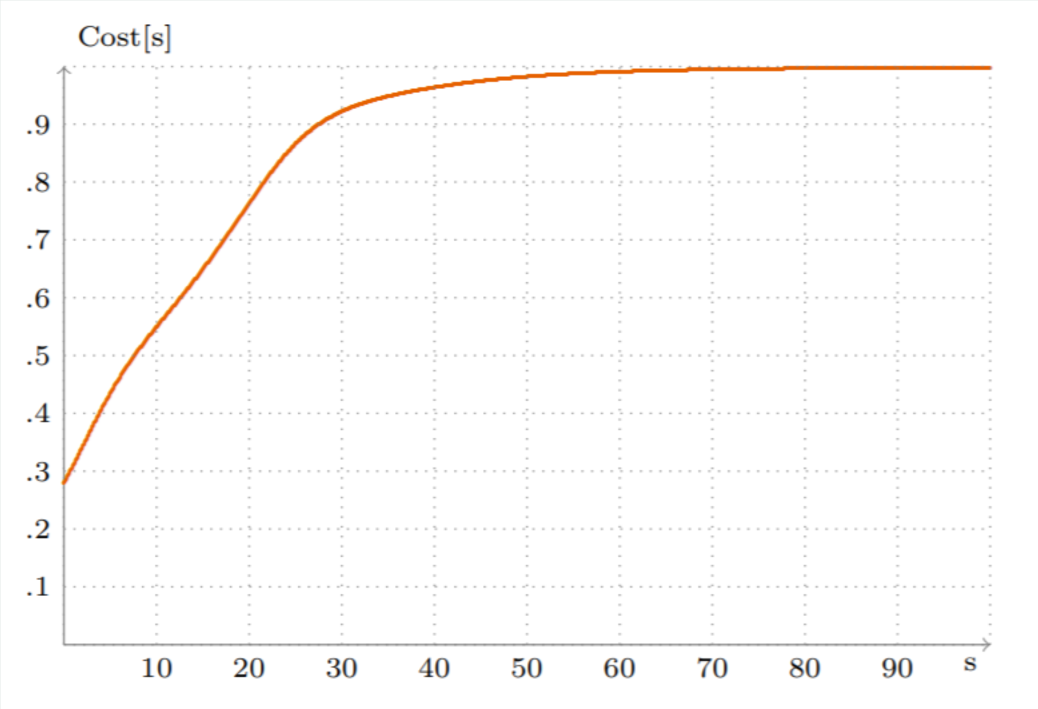


network, $\eta = \frac{1}{3}, \varepsilon = 0.1$



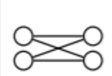


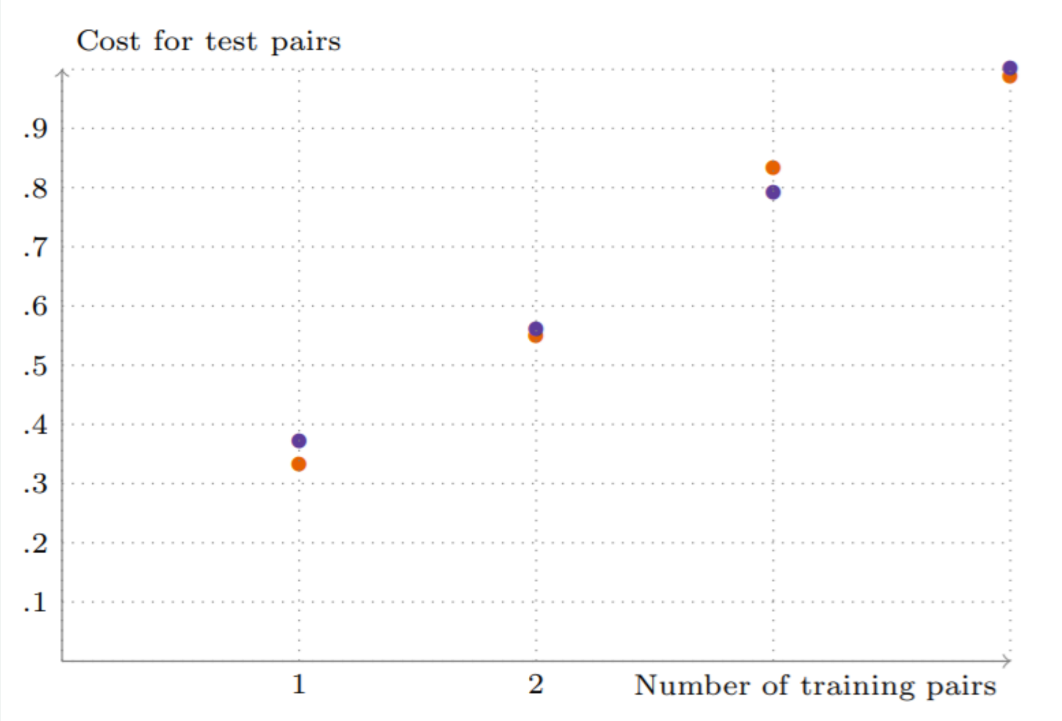
network, $\eta = \frac{1}{4}, \varepsilon = 0.1$

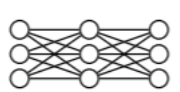


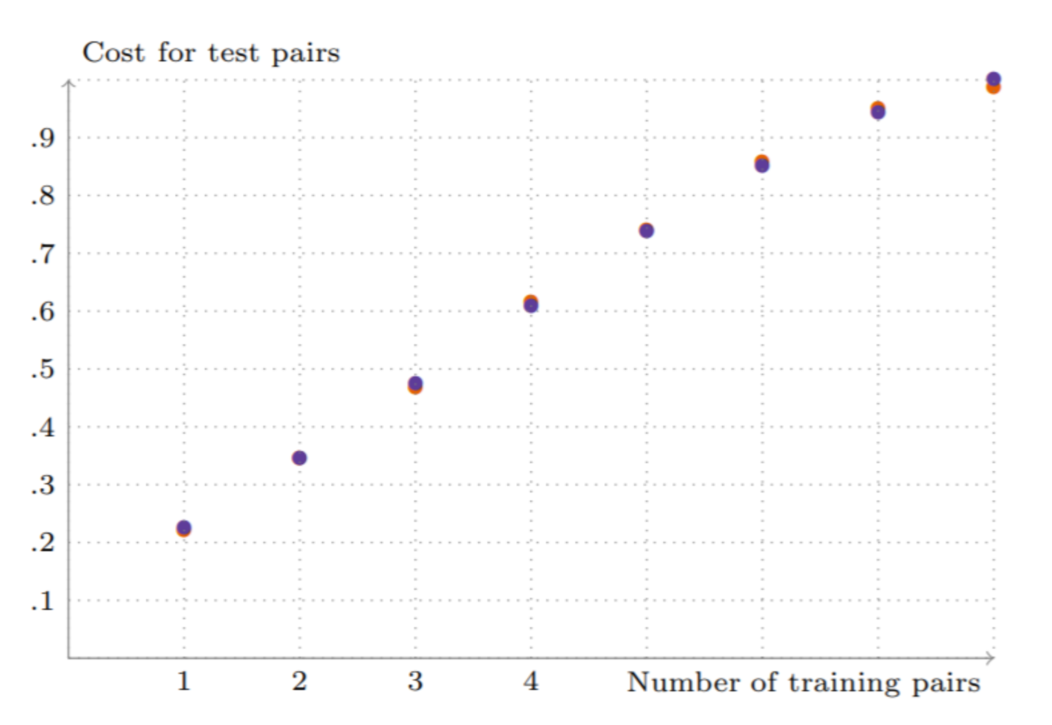
Training deep quantum neural networks [1]

Generalisation

 network, $\eta = 1, \varepsilon = 0.1, 10$ pairs

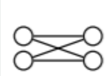


 network, $\eta = 0.1, \varepsilon = \frac{2}{3}$, 10 pairs

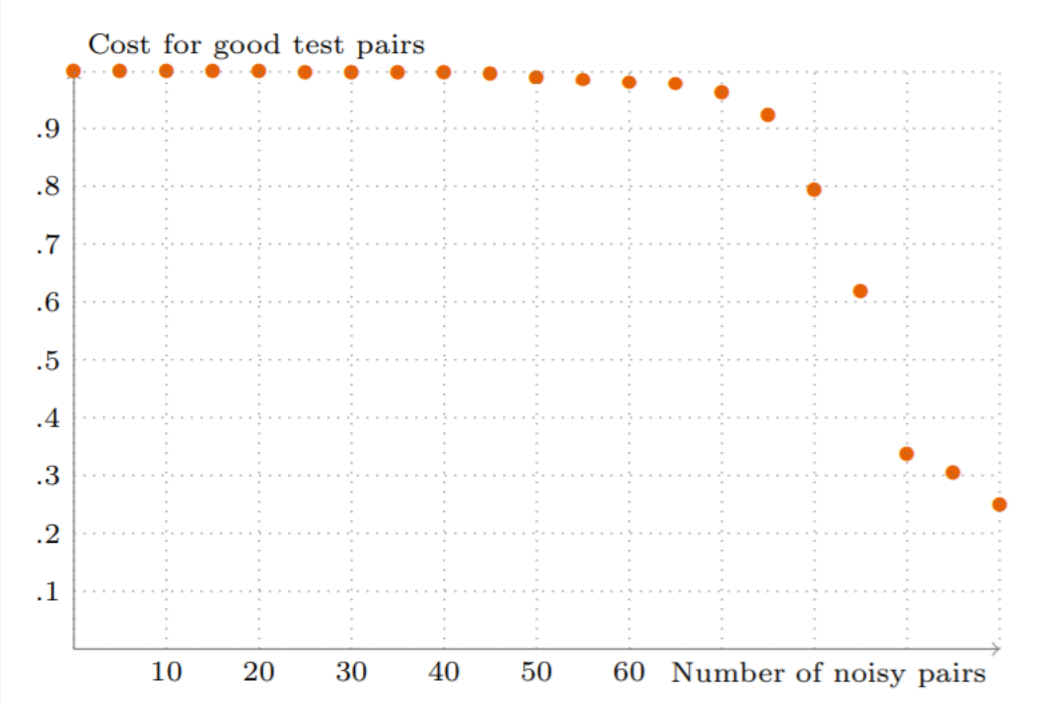


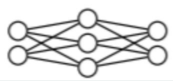
Training deep quantum neural networks [1]

Robustness to Noisy Data

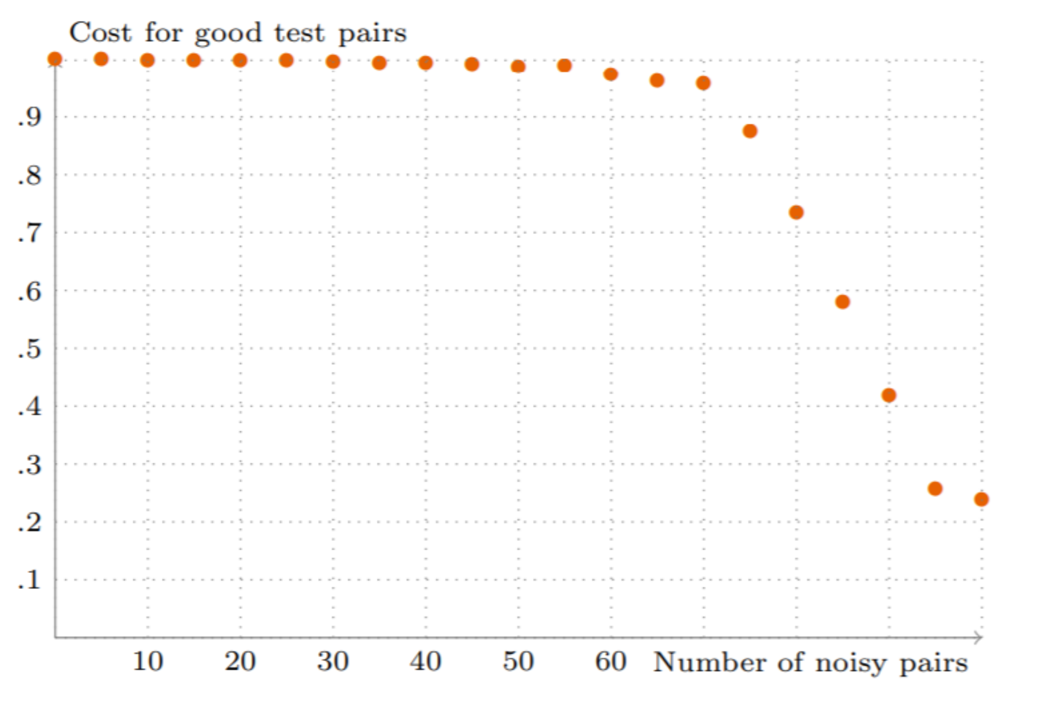


network, $\eta = 1, \varepsilon = 0.1, 100$ pairs





network, $\eta = 1, \varepsilon = 0.1,$ 100 pairs



Training deep quantum neural networks [1]

Takeaways:

- ✓ Transition to quantum architecture
- ✓ Training Algorithm
- ✓ QNN in classical simulation
- ✓ Generalisation and tolerance to noisy data

References

[1] Beer, K., Bondarenko, D., Farrelly, T., Osborne, T. J., Salzmann, R., Scheiermann, D., & Wolf, R. (2020). Training deep quantum neural networks. *Nature communications*, *11*(1), 1-6.

[2] Carleo, G., & Troyer, M. (2017). Solving the quantum many-body problem with artificial neural networks. *Science*, *355*(6325), 602-606.

[3] Tiersch, M., Ganahl, E. J., & Briegel, H. J. (2015). Adaptive quantum computation in changing environments using projective simulation. *Scientific reports*, *5*(1), 1-18.

[4] Lovett, N. B., Crosnier, C., Perarnau-Llobet, M., & Sanders, B. C. (2013). Differential evolution for many-particle adaptive quantum metrology. *Physical review letters*, *110*(22), 220501.

References

[5] Carleo, G., & Troyer, M. (2017). Solving the quantum many-body problem with artificial neural networks. *Science*, *355*(6325), 602-606.

[6] Beer, K., Bondarenko, D., Farrelly, T., Osborne, T. J., Salzmann, R., & Wolf, R. (2019). Efficient learning for deep quantum neural networks. *arXiv preprint arXiv:1902.10445*.