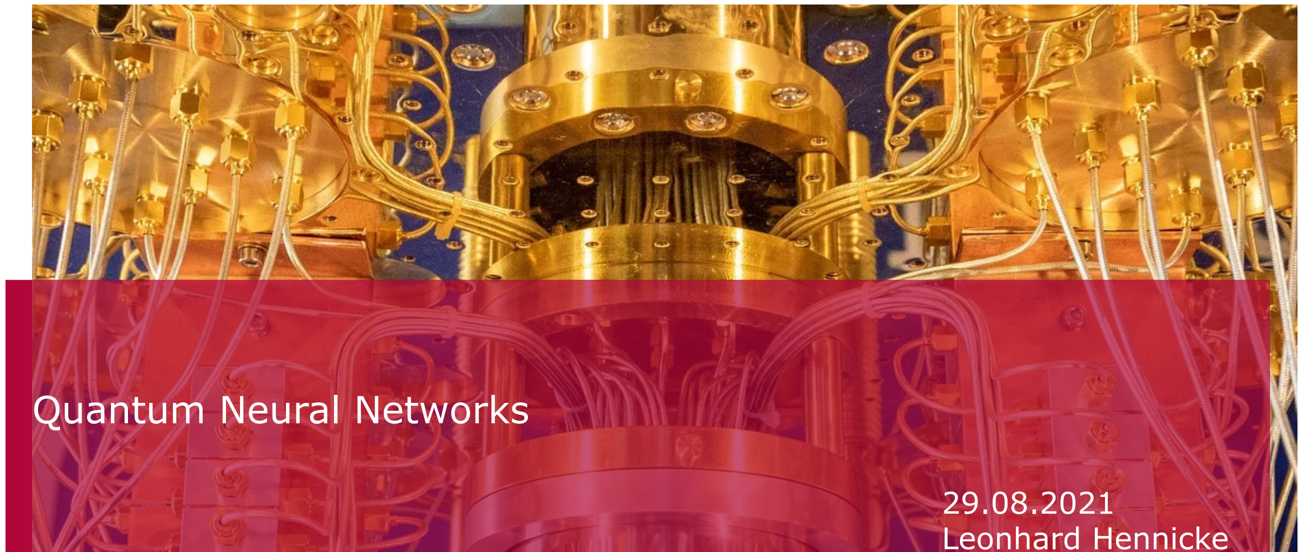
Hasso Plattner Institut
Digital Engineering · Universität Potsdam



Quantum Neural Networks

29.08.2021
Leonhard Hennicke

Test notes

https://www.cnet.com/a/img/5nXMNCjsK_b_Y-kF-Gh33cx78dM=/1200x675/2019/06/26/3f76e99d-8055-46f3-8f27-558ee276b665/20180405-ibm-q-quantum-computer-02.jpg

Agenda

1. Motivation
2. Enhancing Training and evaluation
3. Quantum Neural Networks
    1. Quantum Neurons
    2. Cluster State
    3. QCNNs (Example)

Motivation

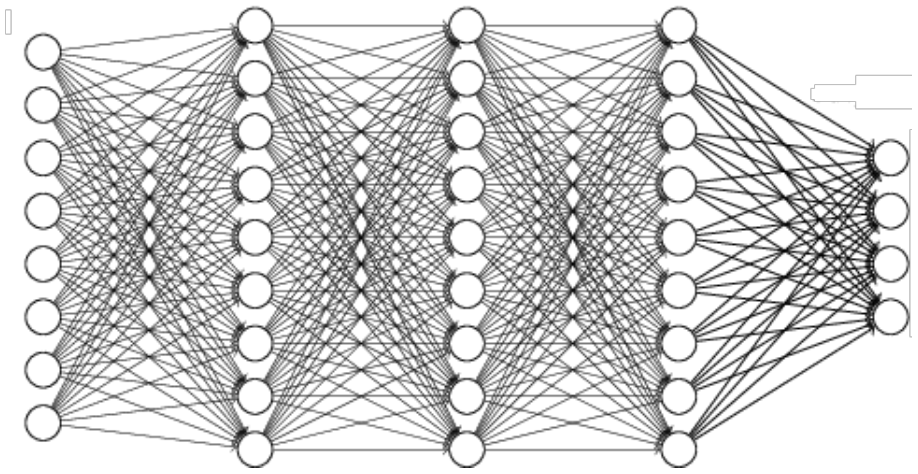
Functions can be modelled with fewer neurons (eg XOR can’t be captured by single layer NN, but possible with QNN)

Efficiently processing quantum data (process training data in superposition all at once) no sampling/tomography

Using speedup in dependencies to improve runtime -> transformer, more hyperopt

Enhancing Training and Evaluation

Enhancing Training and Evaluation



Quantum Neural Networks

Leonhard Hennicke
29.09.2021

quantum algorithms for training & evaluating feedforward NNs based on canonical classical feedforward & backpropagation algorithms

efficient quantum subroutine for approximating the inner products between vectors in a robust way

implicitly storing large intermediate values in qRAM for fast retrieval at later stages

quadratically faster in the size of the network, since they depend linearly on the number of neurons, not the number of connections

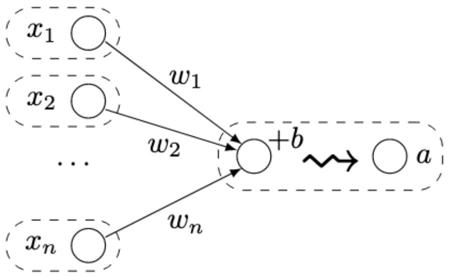
Not as prone to overfitting since they mimic regularisation techniques such as dropout

Challenges:
Sequential process, but quantum comp good at parallel
Storing at intermediate steps mat destroy quantum coherence
non-linearity with qbits
Quantum state preparation as bottleneck

https://i.stack.imgur.com/MBlhW.png

Quantum Neural Networks

Quantum Neurons



- Neuron as seen in standard Neural Networks
- Weighted sum of input values, combined with a bias as input for a non-linear activation function to produce output
- **How to produce non-linearity with QBits?**

Several approaches:
Using quantum measurements,

exploiting quadratic form of kinetic term,

dissipative quantum gates,

reversible circuits

Quantum Neurons

Using Pauli-Gate:

$$R_y(a\frac{\pi}{2}+\frac{\pi}{2})\,|0\rangle = \cos(a\frac{\pi}{4}+\frac{\pi}{4})\,|0\rangle + \sin(a\frac{\pi}{4}+\frac{\pi}{4})\,|1\rangle, \text{ where } a \in [-1,1]$$

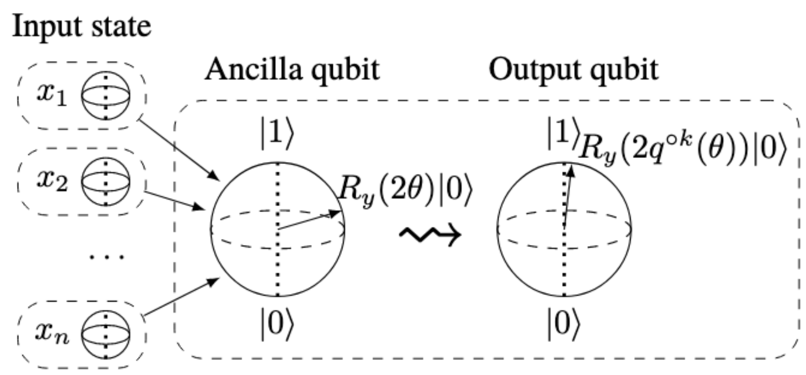
- *a* = 1 and *a* = -1 correspond to $|1\rangle$ and $|0\rangle$
  (similar to classical neuron with binary output state)
- $a \in (-1,1)$ represents neuron in superposition
  (no classical analogy)
- However, this is still linear

use the state $|x\rangle = |x_1 \cdots x_n\rangle$ as a control state and apply $R_y(2w_i)$

onto an ancilla qubit conditioned on the i-th qubit,

followed by $R_y(2b)$ on the ancilla qubit.

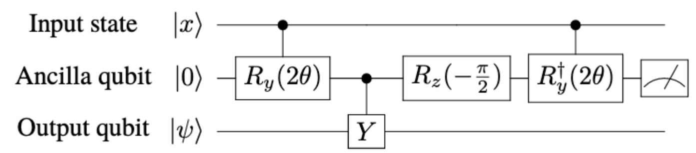
Quantum Neurons

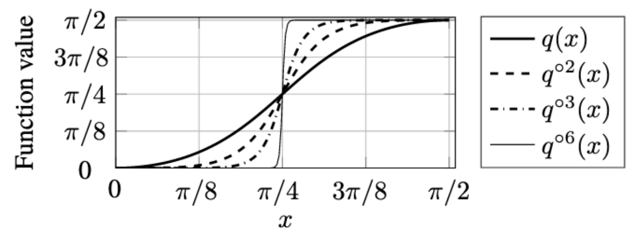


where q(φ) = arctan(tan² φ)

Two parts
1. Controlled pauli gates multiplied with weights, then pauli with bias on ancilla bit
2. Controlled pauli gate with non-linear activation function on output bit using RUS

Quantum Neurons





Quantum Neural Networks

Leonhard Hennicke
29.09.2021

Do rotation, measure if successful and then reapply arbitrarily often if not
This gives non-linearity:
closer to threshold, more repetitions until result close enough to extrema
This introduces stochasticity though

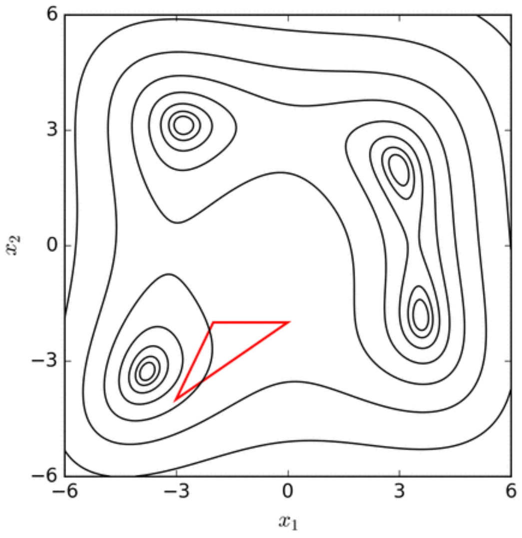
The action of an RUS circuit on the output qubit depends on the measurement outcome of the ancilla qubit. If the measurement returns |0〉, this indicates that the rotation by $2q^{\circ k}(\theta)$ has been successfully applied to the output qubit. Otherwise if the ancilla qubit measures |1〉, this indicates that the circuit has implemented a rotation $R_y(\pi/2)$ onto the output qubit. In this case we correct the operation by applying $R_y(-\pi/2)$ and then repeat the circuit until |0〉 is measured in the ancilla qubit, hence the name repeat until success.

Quantum Neurons

- Training of NNs usually using gradient descend
- Repeat-until-success circuits are stochastic, depending on measurements
- Hard to calculate gradient of objective function

- Nelder-Mead algorithm for gradient-free local optimization

Quantum Neurons



Nelder-Mead algorithm on Himmelblau's function

Heuristic:

- may converge to non-stationary points
- Trial and error

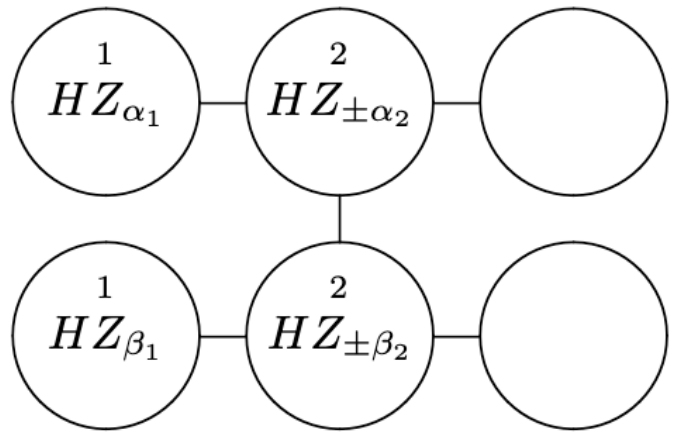
Simplex of the n-dimensional space given by number of variables of the objective function
Choose worst point and replace with better point

Finding better point:
Reflect worst point through centroid
if much better, stretch out
if worse, contract
Methods using constant size simlplex exist, but worse performance

https://en.wikipedia.org/wiki/Nelder%E2%80%93Mead_method#/media/File:Nelder-Mead_Himmelblau.gif

Cluster State

Cluster State



Prepared, special entangled many-qubit quantum state

1. each qbit in |+⟩ superposition I.e. H |0⟩
2. controlled phase gates between qbits that are connected in the graph (commute, so order irrelevant)

Adaptive sequence of single qbit measurements which process the cluster

single qbit measurements
basis may depend on results of earlier measurements
-> constrains complexity if processed classically

Read out of remaining qbits (either as quantum state or as read-out measurements)

Example: n is order of measurements, HZ which basis (single qbit unitary denoting rotation followed by basis measurements) and +- indicates that choice of sign depends on earlier results

QCNNs (Example)

Outlook

- Using Quantum Neurons for fully connected layers
- Testing

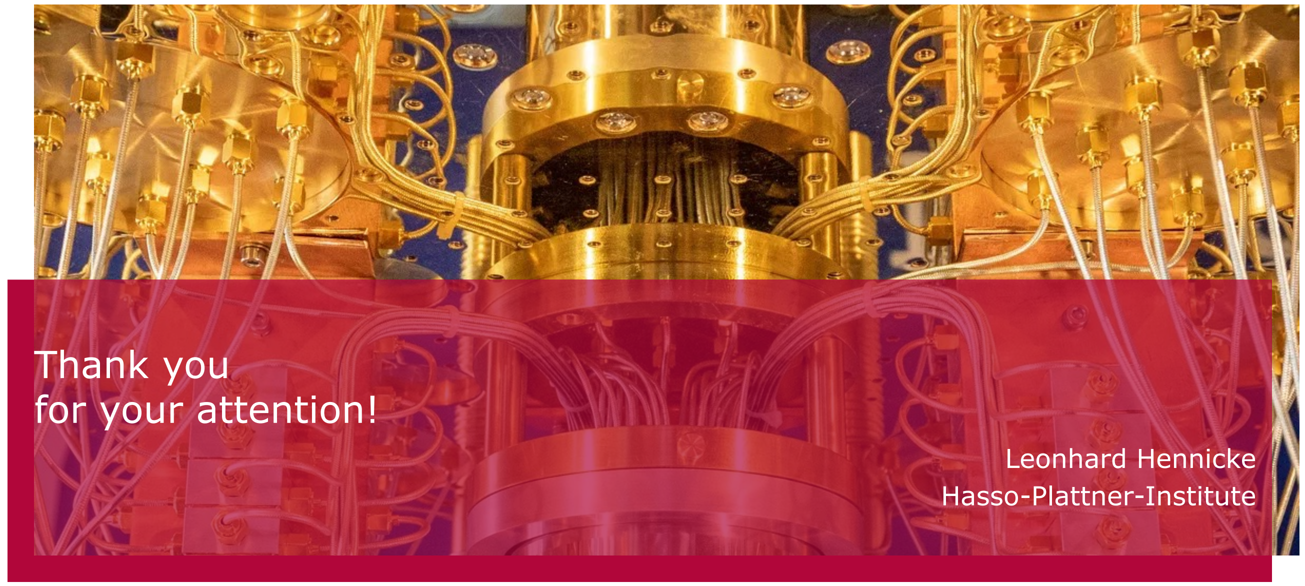
Not sure if testing makes a whole lot of sense. Only thing that should have defined behaviour a priori are the gates themselves and the input state.

Concrete gate behaviour is dependent on the parameters which are to be optmizied, so it changes.
So does the expected state at different points of circuit.
Could make assertions for entanglement if fitting places in circuit can be found.
As long as the implementation of the gates is correct, there is not much to be tested.
If there where errors in gate implementation it is questionable wether they would be found in training while parameters change.

Input state could be tested, but that would defeat the point of accelerating/estimating the processing of quantum many-body states.

End-to-end tests make sense, which is what the testing set is for, but still, 100% accuracy is not expected and testing correct implementation of the gates might be more efficient on a per gate basis.

Hasso Plattner Institut
Digital Engineering · Universität Potsdam



Thank you
for your attention!

Leonhard Hennicke
Hasso-Plattner-Institute

https://www.cnet.com/a/img/5nXMNCjsK_b_Y-kF-Gh33cx78dM=/1200x675/2019/06/26/3f76e99d-8055-46f3-8f27-558ee276b665/20180405-ibm-q-quantum-computer-02.jpg

References

1. Cao, Yudong, Gian Giacomo Guerreschi, and Alán Aspuru-Guzik. "Quantum neuron: an elementary building block for machine learning on quantum computers." *arXiv preprint arXiv:1711.11240* (2017).
2. Adedoyin, Adetokunbo, et al. "Quantum algorithm implementations for beginners." *arXiv preprint arXiv:1804.03719* (2018).
3. Cong, Iris, Soonwon Choi, and Mikhail D. Lukin. "Quantum convolutional neural networks." *Nature Physics* 15.12 (2019): 1273-1278.
4. Allcock, Jonathan, et al. "Quantum algorithms for feedforward neural networks." *ACM Transactions on Quantum Computing* 1.1 (2020): 1-24.
5. Nielsen, Michael A. "Cluster-state quantum computation." *Reports on Mathematical Physics* 57.1 (2006): 147-161.
6. Yanofsky, Noson S. "An introduction to quantum computing." *Proof, Computation and Agency*. Springer, Dordrecht, 2011. 145-180.
7. Tacchino, Francesco, et al. "An artificial neuron implemented on an actual quantum processor." *npj Quantum Information* 5.1 (2019): 1-8.
8. Paetznick, Adam, and Krysta M. Svore. "Repeat-Until-Success: Non-deterministic decomposition of single-qubit unitaries." *arXiv preprint arXiv:1311.1074* (2013).
9. Wiebe, Nathan, and Vadym Kliuchnikov. "Floating point representations in quantum circuit synthesis." *New Journal of Physics* 15.9 (2013): 093041.
10. Nelder, John A., and Roger Mead. "A simplex method for function minimization." *The computer journal* 7.4 (1965): 308-313.