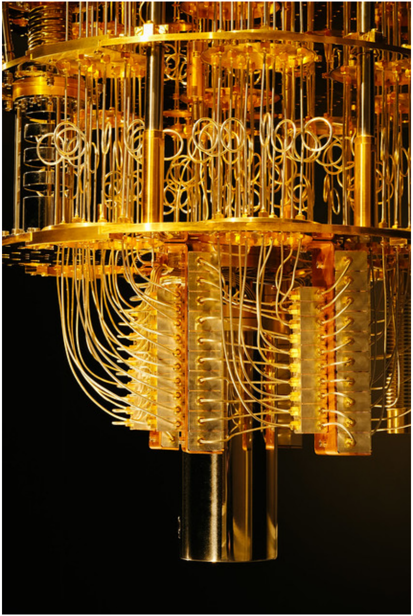
An Artificial Spiking Quantum Neuron

Matthias Degroote
Work with Lasse Bjørn Kristensen, Peter Wittek, Nikolaj Zinner, Alán Aspuru-Guzik

Aspuru-Guzik Group Meeting 08/08/2019



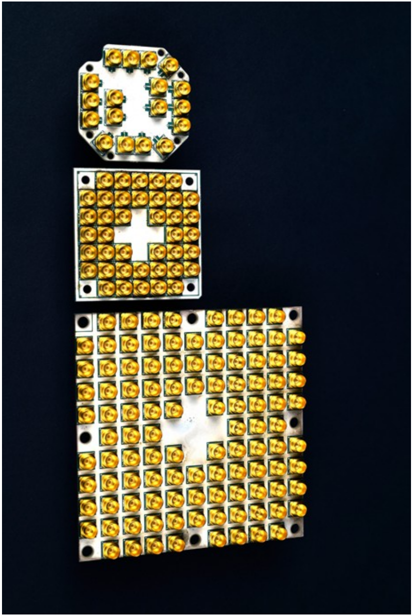
The big promise

Universal quantum computers can do anything classical computers can do and more...

Speedups

Algorithms with proven speedup

- Shor's factoring
- Grover search
- Boson Sampling



The disappointment

You need

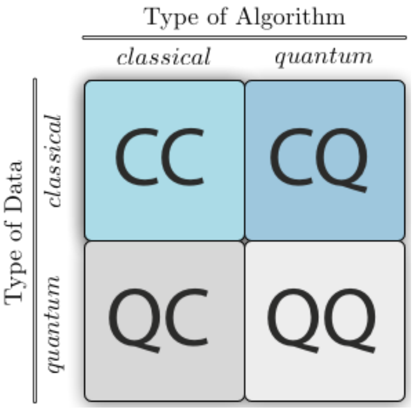
- very large
- perfectly performing
- well-programmed

quantum computers.

In the meantime

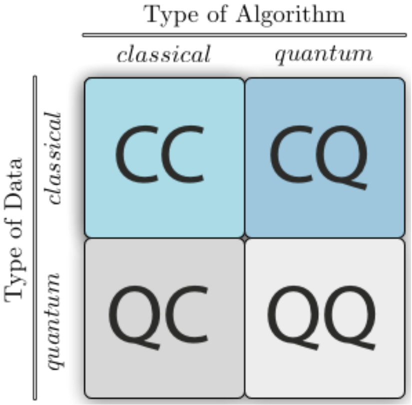
Do what they are good at

- discrete optimization
- quantum chemistry
- **machine learning**



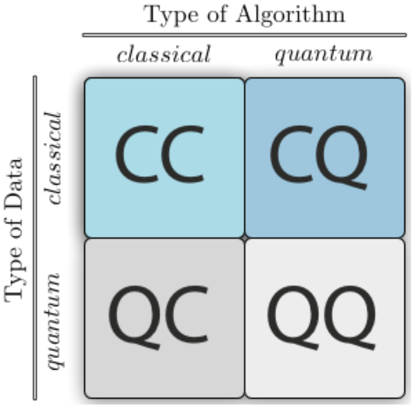
QC

- Quantum phase transition identification by adversarial autoencoders
- Machine learning for tensor networks
- Learning of density functionals
- Recurrent neural networks learning parameters for quantum circuits
- Materials design in the group



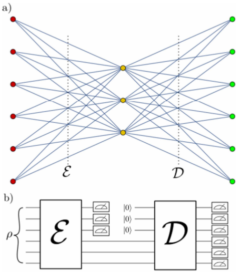
CQ

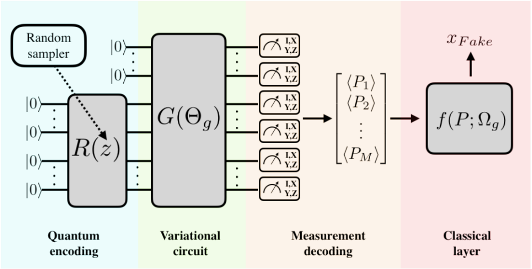
- Quantum enabled sampling
  e.g. Xanadu's Gaussian Boson Sampling
- Quantum Support Vector Machines
- Quantum Boltzman Machines
  e.g. D-Wave's samplers

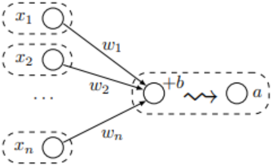


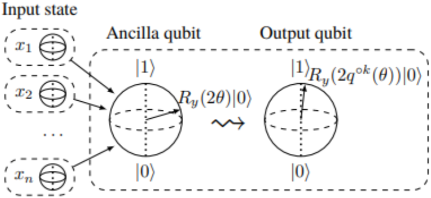
QQ

- Quantum Generative Adversial Networks
- Quantum Variatonal Autoencoder
- Quantum Classifiers
- Quantum Neurons


a)
$\mathcal{E}$
$\mathcal{D}$
b)
ρ
$\mathcal{E}$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$\mathcal{D}$


Random sampler
$|0\rangle$
$R(z)$
$G(\Theta_g)$
I,X Y,Z
$\langle P_1\rangle$ $\langle P_2\rangle$ $\vdots$ $\langle P_M\rangle$
$f(P;\Omega_g)$
x_{Fake}
Quantum encoding
Variational circuit
Measurement decoding
Classical layer


x_1
x_2
...
x_n
w_1
w_2
w_n
$+b$
a

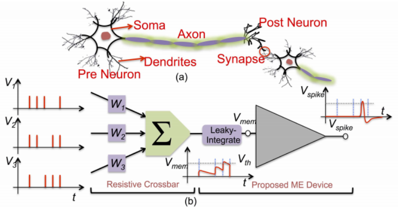

Input state
x_1
x_2
...
x_n
Ancilla qubit
Output qubit
$|1\rangle$
$|0\rangle$
$R_y(2\theta)|0\rangle$
$R_y(2q^{\circ k}(\theta))|0\rangle$

Power Consumption

Game	Computer	Human
Chess	900W	20W
Go	1000W	20W
StarCraft II	2000W	20W

How is this possible?

- sparse connectivity
- info transfer through spikes
- thresholding
- temporal character
  (firing rate, relative timing)



Requirements

- **Integrate and fire**
- synaptic plasticity
- only feed-forward

	System overview		On-chip learning	Inference of deep SNNs					
Name	**Type of hardware**	**Neuron and synapse count**	**Plasticity rules**	**Network type**	**Network size**	**Dataset**	**Test accuracy**	**Throughput (in img/s)**	**Energy consumption (per image)**
TrueNorth (Merolla et al., 2014)	digital	single chip: 4096 cores, 1M neurons, 256M synapses; up to 8 chips	None	Deep CNNs: (a, b) Esser et al. (2015) (c) Esser et al. (2016)	(a) 1920 cores (b) 5 cores (c) 8 chips	(a, b) MNIST (c) CIFAR10 and many more	(a) 99.4% (b) 92.7% (c) 89.3%	(a, b) 1000 (c) 1249	(a) 108 μJ (b) 0.268 μJ (c) 164 μJ
SpiNNaker (Furber et al., 2013)	digital	single chip: 18 ARM cores, approx. 1k neurons and 1k synapses per core for real-time simulations; up to 576 chips	flexible, e.g., unsupervised (Jin et al., 2010) and supervised (Mikaitis et al., 2018) STDP	DBN: 2 hidden layers with 500 neurons each (Stromatias et al., 2015)	1 chip	MNIST	95%	91	3.3 mJ
BrainScaleS (Schemmel et al., 2010; Brüderle et al., 2011)	mixed-signal	wafer with 384 cores, 200k neurons, 45M synapses	STDP (Schemmel et al., 2006; Pfeil et al., 2013b)	MLP: 2 hidden layers with 15 neurons each (Schmitt et al., 2017)	14 cores	downscaled MNIST	95%	10000	7.3 mJ

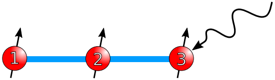
Spike communication in all of these systems is asynchronous. The SpiNNaker system is the only system listed in this table that allows events with payload (see section 1). Note that none of these systems natively support batching of inputs as commonly used in conventional deep learning.

Functions

- measure excitations
- measure relative phases

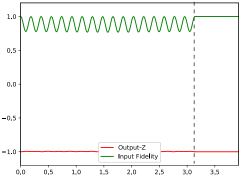
Requirements

- have distinct temporal character
- no specific hardware paradigm
- low qubit overhead
- leave input unchanged



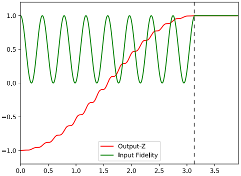
$e^{-\frac{iHt}{\hbar}}$ PHASE

$\left|\Psi^{\pm}\right\rangle\left|\downarrow\right\rangle \longrightarrow$



$\longrightarrow \left|\Psi^{\pm}\right\rangle\left|\downarrow\right\rangle$

$\left|\Phi^{\pm}\right\rangle\left|\downarrow\right\rangle \longrightarrow$



$\longrightarrow \left|\Phi^{\pm}\right\rangle\left|\uparrow\right\rangle$

Hamiltonian

$$H = \frac{J}{2}\vec{\sigma_1}\cdot\vec{\sigma_2} + \beta\sigma_2^z\sigma_3^z$$

$$+A\cos\left(\frac{2\beta}{\hbar}t\right)\sigma_3^x$$

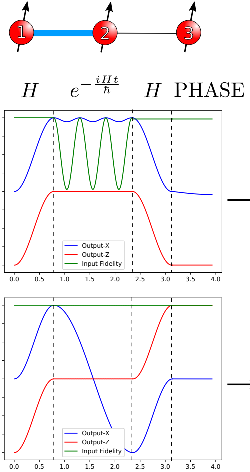
$$\left|2\beta \pm 2\sqrt{J^2+\beta^2}\right| \gg A$$

$$\beta = kA$$

$$J = \pm\sqrt{l^2 - k^2}A$$

$|\Phi^+\rangle|\downarrow\rangle$
$|\Psi^+\rangle|\downarrow\rangle$ $\longrightarrow$ $|\Phi^+\rangle|\downarrow\rangle$
$|\Psi^+\rangle|\downarrow\rangle$

$|\Phi^-\rangle|\downarrow\rangle$
$|\Psi^-\rangle|\downarrow\rangle$ $\longrightarrow$ $|\Phi^-\rangle|\uparrow\rangle$
$|\Psi^-\rangle|\uparrow\rangle$



Hamiltonian

$$H = \frac{J}{2}\vec{\sigma_1}\cdot\vec{\sigma_2} + \delta\sigma_2^x\sigma_3^x + B\sigma_3^z$$

$$2\delta \gg B$$

$$\delta = 2m$$

$$J = 2n$$

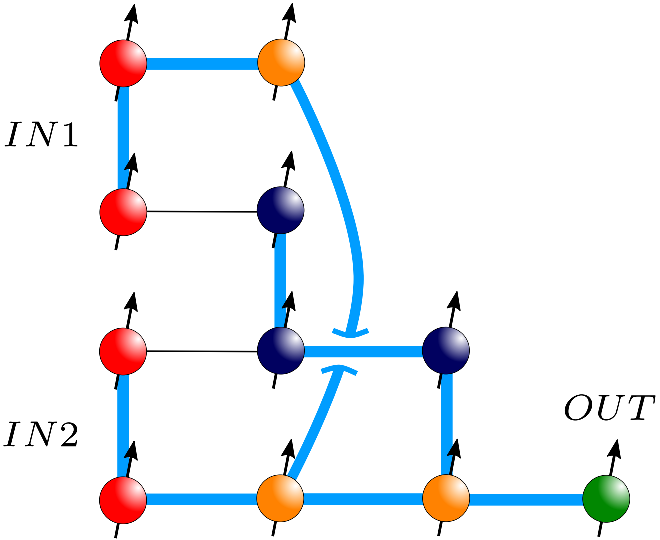
$$n \gg m$$

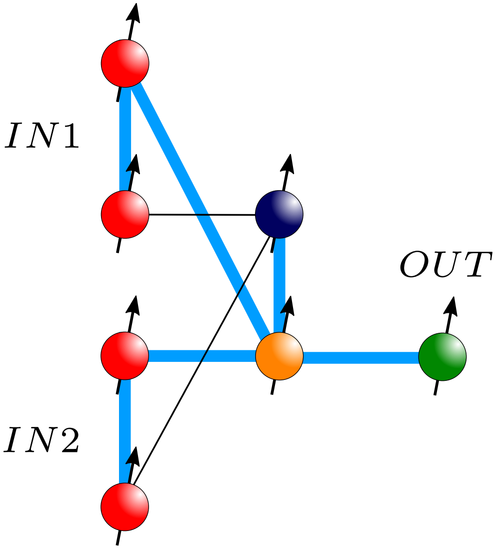
4 maximally entangled 2 qubit states

$$
\begin{aligned}
\left|\Phi^{+}\right\rangle &= \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \\
\left|\Phi^{-}\right\rangle &= \frac{1}{\sqrt{2}}\left(|00\rangle - |11\rangle\right) \\
\left|\Psi^{+}\right\rangle &= \frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right) \\
\left|\Psi^{-}\right\rangle &= \frac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right)
\end{aligned}
$$

Why are they special?

- correlations beyond classical
- used in quantum communication
  e.g. teleportation, superdense coding
- quantum cryptography


$IN1$
$IN2$
OUT



Differences

- Reduction of #qubit
- No longer strictly feed forward
- Complication for last layer

Measurement back-action:

$$
\begin{aligned}
& \text{Comparator} \frac{1}{\sqrt{2}} \left(\left| \Psi_1^+ \right\rangle + \left| \Phi_1^- \right\rangle \right) \left| \Psi_2^+ \right\rangle \left| \downarrow \right\rangle \\
= \quad & \frac{1}{\sqrt{2}} \left(\left| \Psi_1^+ \right\rangle \left| \Psi_2^+ \right\rangle \left| \uparrow \right\rangle + \left| \Phi_1^- \right\rangle \left| \Psi_2^+ \right\rangle \left| \downarrow \right\rangle \right)
\end{aligned}
$$

Defines an expected likelihood kernel:

$$
P_{\left| \uparrow \right\rangle} = \sum_{0}^{3} \left| a_i \right|^2 \left| b_i \right|^2
$$

$$
k\left(p, p'\right) = \int p(x) p'(x) \mathrm{d}x
$$

Can be used for quantum kernel learning methods

Thank you for your attention!

Questions are welcome

Further reading and self-promotion:
An Artificial Spiking Quantum Neuron
https://arxiv.org/abs/1907.06269

Slides:
https://mfdgroot.github.io/
Slack #group_meeting