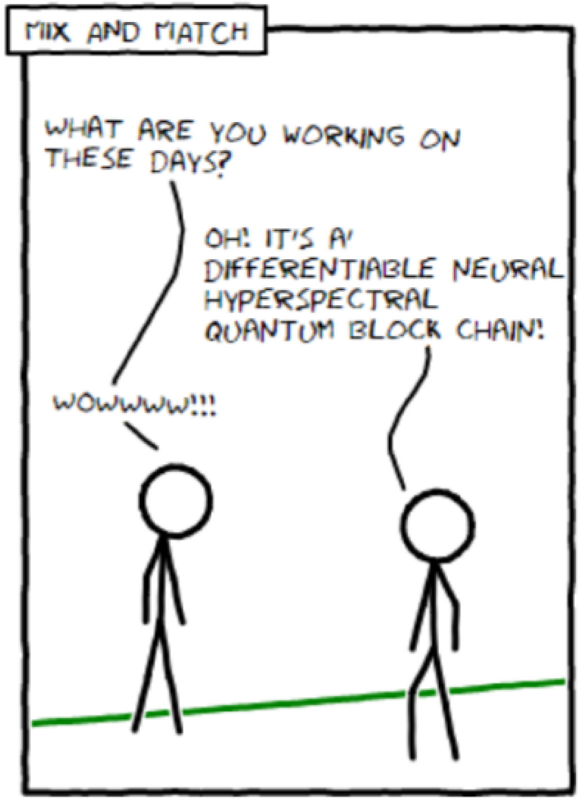
Reinforcement Learning and Quantum Computing

UncharTECH

William Clements
https://unchartech.io/



Source: Minhas, Fayyaz, Amina Asif, and Asa Ben-Hur. "Ten ways to fool the masses with machine learning." arXiv preprint arXiv:1901.01686 (2019).

1. The intersection of quantum and RL

Quantum computing and Reinforcement learning: what substance behind the buzzwords?

1) Introduction to quantum computing

2) Quantum computing is experimentally fiendishly hard. RL is good at solving hard problems. Can RL help build a quantum computer?

3) Quantum computing introduces new complexity classes. Could a quantum computer be used to achieve faster learning?

2. Quantum physics

Physics governing the properties of very small physical systems

Quantum systems obey different laws than classical systems

Classical physics	Quantum physics
Stochastic matrix: All real and positive values	Unitary matrix: Complex values with norm 1 eigenvalues
$\begin{pmatrix} s_{11} & \cdots & s_{1n} \\ \vdots & \ddots & \vdots \\ s_{n1} & \cdots & s_{nn} \end{pmatrix} \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix} = \begin{pmatrix} q_1 \\ \vdots \\ q_n \end{pmatrix}$	$\begin{pmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \ddots & \vdots \\ u_{n1} & \cdots & u_{nn} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_n \end{pmatrix}$
$p_i \geq 0, \quad \sum_{i=1}^{n} p_i = 1$	$\alpha_i \in \mathbb{C}, \quad \sum_{i=1}^{n} \lvert\alpha_i\rvert^2 = 1$
Transformations that maintain the l1 norm of probability vectors	Transformations that maintain the l2 norm of amplitude vectors

Credit : Scott Aaronson

3. Example: classical vs quantum coin flip

Coin: can be described by a vector of size 2

Heads: $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ Tails: $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Classical coin flip: $\begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$

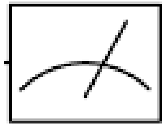
Two classical coin flips: $\begin{pmatrix} 1 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \rightarrow \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$

Quantum coin flip: $\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$

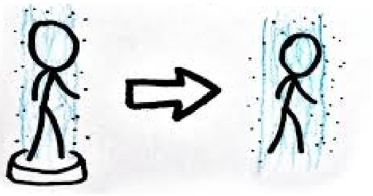
Two quantum coin flips: $\begin{pmatrix} 1 \\ 0 \end{pmatrix} \rightarrow \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

4. Properties of quantum systems

Superposition of states



Interaction with a classical environment: collapses the system to a single state, with probability given by amplitude squared



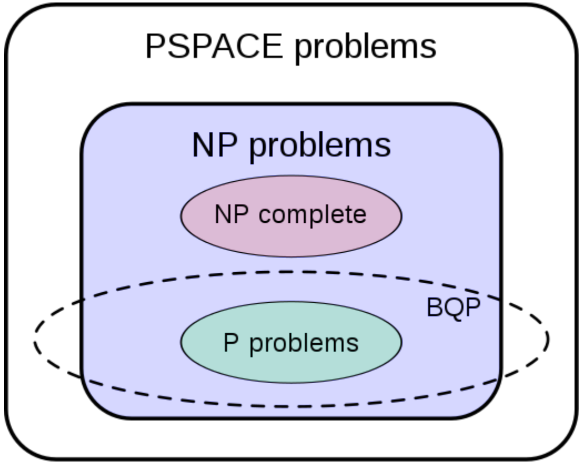
Entanglement: a measurement on one system can perturb another system

5. Why quantum computing?

Computers are physical objects: limited by the laws of physics

- Landauer's principle: erasing a bit of information requires energy

- Complexity classes derive from physics: imagine a computer that could do all possible calculations simultaneously



The use of different physical laws means different complexity classes can be reached

Quantum physics is more general than classical physics: more general complexity class "BQP"

6. Quantum bits

Qubits: arbitrary superposition of 0 and 1, which we note $|0\rangle$ and $|1\rangle$

The state of a qubit is written as: $\alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$

- Amplitudes are continuous

- Quantum operations change the amplitudes

- Measurement/interactions with a classical environment collapse the qubit

The state of two qubits is:

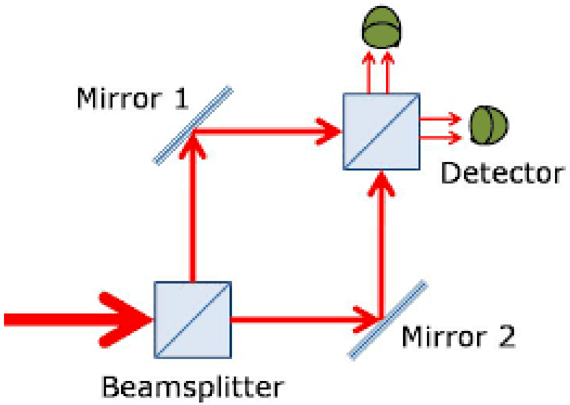
$$\begin{bmatrix} v_{00} \\ v_{01} \\ v_{10} \\ v_{11} \end{bmatrix} = v_{00}|00\rangle + v_{01}|01\rangle + v_{10}|10\rangle + v_{11}|11\rangle$$

first qubit second qubit

7. Recipe for making a qubit

Example: the position of a photon can encode a qubit

Photon



Photon arrives from left: $|0\rangle$ or $\begin{pmatrix}1\\0\end{pmatrix}$

Photon arrives from bottom: $|1\rangle$ or $\begin{pmatrix}0\\1\end{pmatrix}$

Each beamsplitter acts as a quantum coin toss

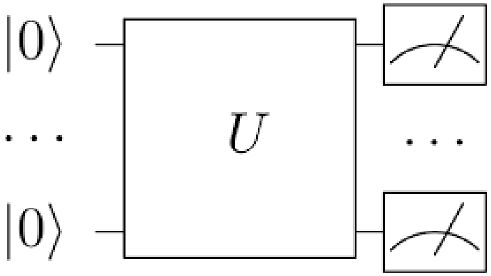
Two beamsplitters: the photon always exits the same port $\begin{pmatrix}1\\0\end{pmatrix} \rightarrow \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} \rightarrow \begin{pmatrix}1\\0\end{pmatrix}$

Multiple photons interfering with each other: quantum operations on multiple qubits

8. Quantum information processing

N classical bits are in only one state at any given time. N qubits can be in a superposition of 2^N states.

A computation may follow the following scheme: preparation, unitary operation, measurement



The objective is to manipulate amplitudes to increase probability that measurement yields correct result

I will discuss three examples of quantum computation

9. Example of a quadratic speedup: Grover search

Problem: find an item ω in an unsorted database of size N

Given a function f such that f(x) = 1 iff x is ω and 0 otherwise, the best classical algorithm requires O(N) calls of f.

Grover's quantum algorithm solves this problem with O(sqrt(N)) calls of f.

Requirement: quantum oracle U_ω for f that can tag the correct answer

$$\begin{cases} U_\omega|x\rangle = -|x\rangle & \text{for } x = \omega, \text{ that is, } f(x) = 1, \\ U_\omega|x\rangle = |x\rangle & \text{for } x \neq \omega, \text{ that is, } f(x) = 0. \end{cases}$$

10. Example of an exponential speedup: Simon's algorithm

Problem: given a black box function f: $\{0,1\}^N \rightarrow \{0,1\}^N$ for which for some unknown s and all x, $f(x) = f(x \oplus s)$, find s.

The best classical algorithm requires $O(2^{N/2})$ calls of f. Simon's algorithm requires $O(n)$ calls of a quantum oracle for f.

A very artificial problem, but which is at the core of other quantum algorithms (such as factorization)

Caveat: in general, not clear how to build a quantum oracle

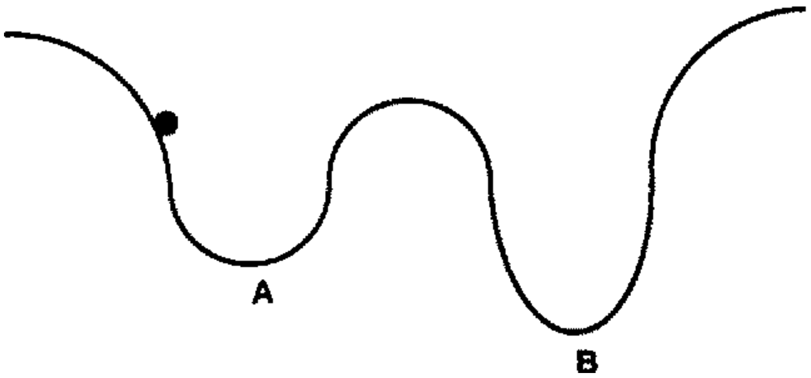
11. Analog quantum computing: quantum annealing

Problem: find $(x_1, \ldots, x_N)$ in $\{0,1\}^N$ that minimizes $\sum_{i,j} w_{i,j} x_i x_j$, where $w_{i,j}$ are real.

NP hard combinatorial optimization problem

Classically, heuristics are used to get out of local minima

Quantum effects can sometimes be used to prevent being stuck in local minima



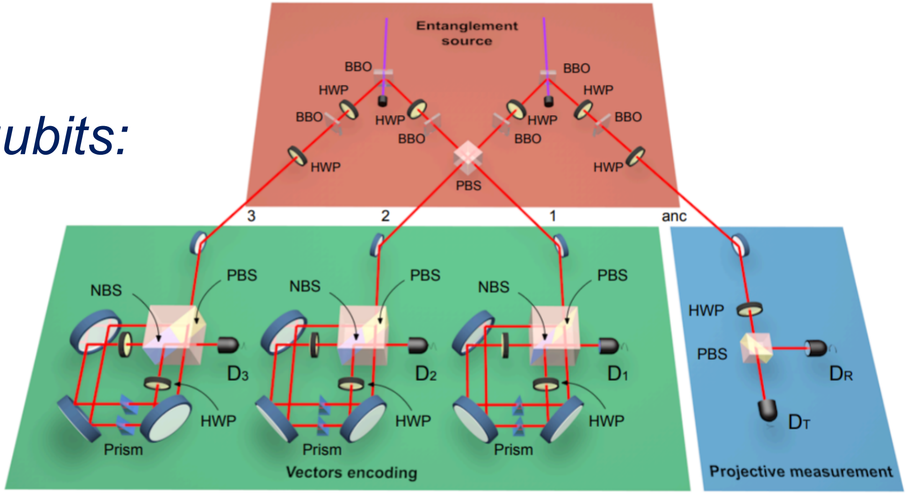
At the heart of a commercial solver: D-Wave

12. Quantum Machine Learning

Downstream tasks that can be accelerated by quantum computing: linear algebra, classification, dimensionality reduction

Example of classification: classifying *N* data points to clusters each with *M* samples takes O(log(MN))

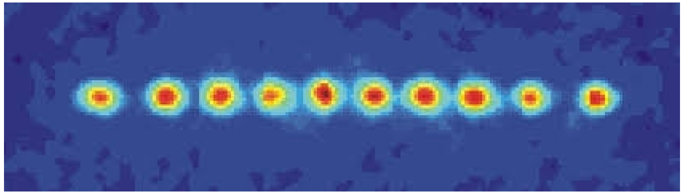
Experiment with 4 photonic qubits:



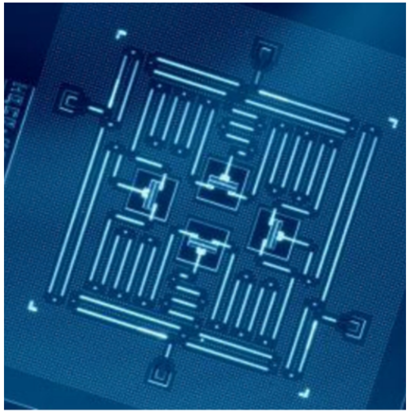
Cai, X-D., et al. "Entanglement-based machine learning on a quantum computer." Physical review letters 114.11 (2015): 110504.

13. State of the art in quantum computing

Current general purpose quantum computers typically have tens of qubits, with trapped ions and superconducting circuits being the most promising



C. F. Roos/IQOQI



Gambetta, Jay M., et al. "Building logical qubits in a superconducting quantum computing system."

Quantum annealers have hundreds of qubits, with less stringent requirements

Not many qubits yet, but there is a rapid rate of increase

Reinforcement Learning for Quantum Computing

Quantum computing is a promising technology, but is very hard to achieve.

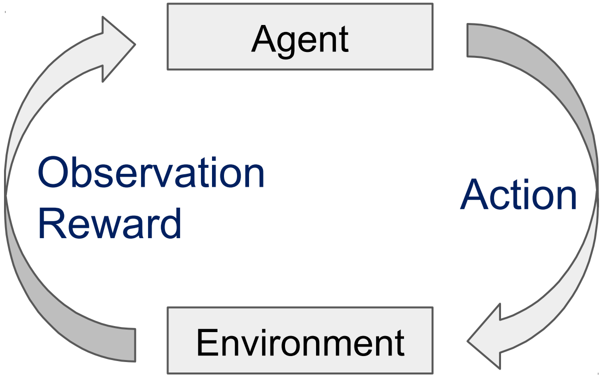
RL can solve hard problems. Can RL help build a quantum computer?

We will look at 3 ways in which RL can be applied to quantum computing:

- Experimental control
- Quantum error correction
- Design of quantum experiments

14. Reinforcement Learning

Overview of RL:



The agent's objective is to find a policy mapping observations to actions

$$\pi : X \to A$$

That maximizes the sum of future rewards

$$\sum_{t=1}^{\infty} \gamma r_t \qquad 0 < \gamma \leq 1$$

(discount factor)

Typical learning process:
1) Explore environment to identify rewards
2) Optimize actions to get the most rewards

15. Reinforcement Learning with Q functions

Several strategies exist to find an optimal policy. One strategy involves learning a “Q-function”:

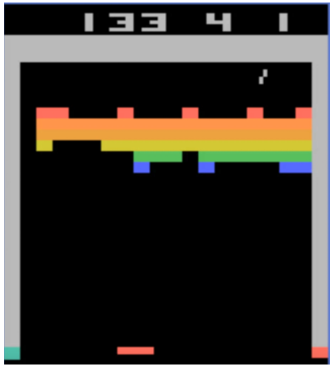
$$Q^{\pi}(a_t, s_t) = \mathbb{E}\left[\sum_{\tau=t}^{\infty} \gamma r_{\tau} | a_t, s_t\right]$$

“Expectation value of taking action a given observation s and then following policy π”

The Q function maps (observations x actions) to a real number: this function can be learned with a neural network

Example: “Deep Q Networks”

Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529.



Observation: pixels

Action: move paddle left or right

Reward: 1 whenever agent breaks a brick

16. Reinforcement Learning for Experimental Control

Problem: controlling a large scale experiment with many controls and measurement systems

Example: two superconducting qubits require 7 time-dependent control parameters (microwave pulse properties and qubit properties)

Solution: Use RL to find the best way to control these parameters to achieve high-quality logic gates

An order of magnitude reduction in gate error can be achieved!

Niu, Murphy Yuezhen, et al. "Universal quantum control through deep reinforcement learning." npj Quantum Information 5.1 (2019): 33.

17. Reinforcement Learning for Error Correction

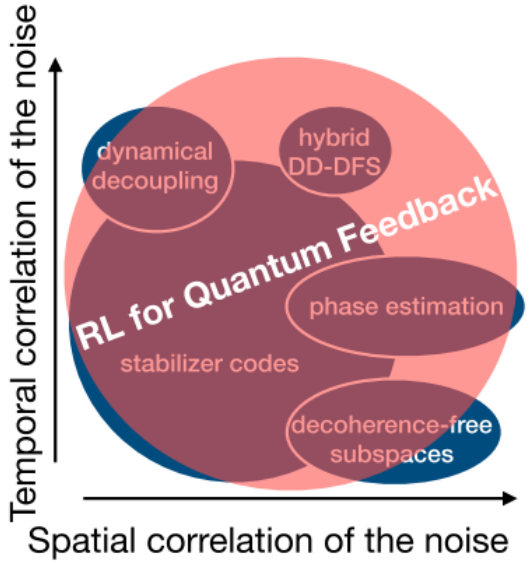
Problem: information stored in qubits is easily perturbed by noise

Much worse than classical bits due to continuous nature of qubits:

$$\alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Quantum error correction methods can be used to protect the information using measurement and feedback methods.

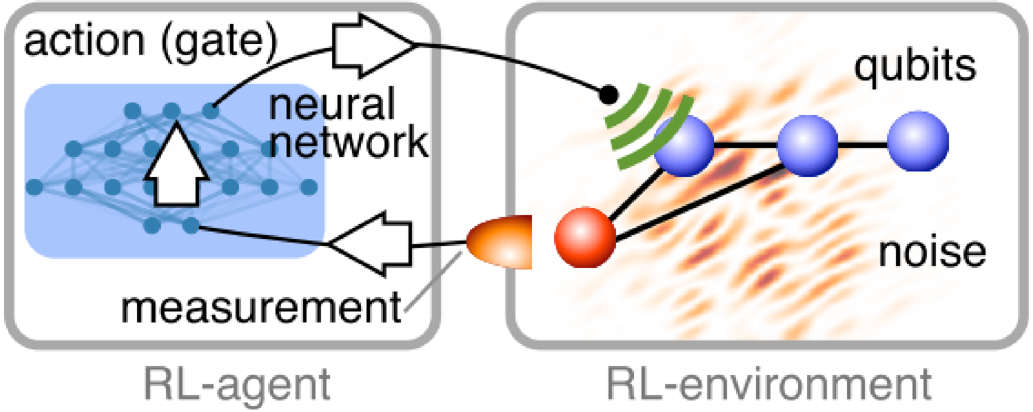
The choice of protocol currently depends on the type of noise: RL can be a more general solution



Fösel, Thomas, et al. "Reinforcement learning with neural networks for quantum feedback." Physical Review X 8.3 (2018): 031084.

18. Reinforcement Learning for Error Correction

Proposed Solution: Use RL to find error correction protocols given hardware specifications and noise model



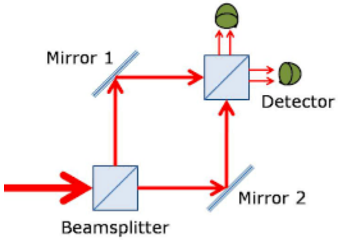
Fösel, Thomas, et al. "Reinforcement learning with neural networks for quantum feedback." Physical Review X 8.3 (2018): 031084.

Observations: partial measurements
Action: logic gate
Reward: “recoverable quantum information”

19. Reinforcement Learning for Quantum State Design

Problem: creating useful quantum states is not straightforward

Given mirrors and lasers, how to make a quantum state that satisfies some property X?



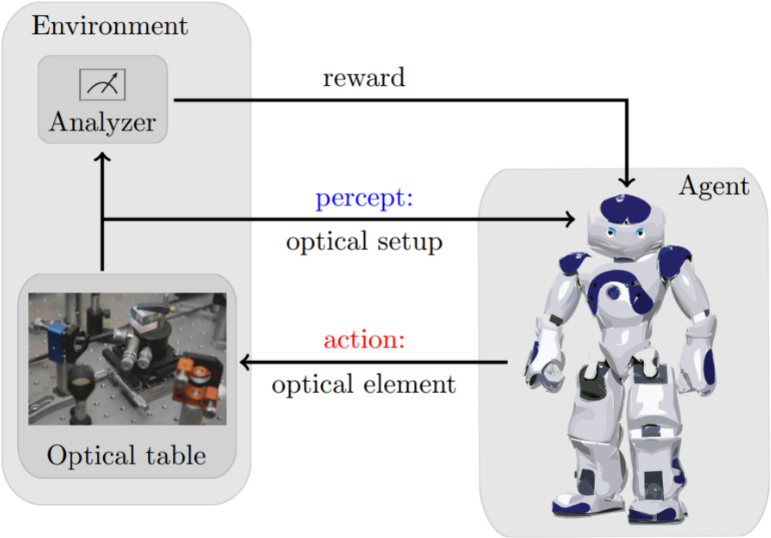
Previous attempts have for example used variants of brute force search:

Krenn, Mario, et al. "Automated search for new quantum experiments." Physical review letters 116.9 (2016): 090405.

Clements, William R., et al. "Approximating vibronic spectroscopy with imperfect quantum optics." Journal of Physics B: Atomic, Molecular and Optical Physics 51.24 (2018): 245503.

20. Reinforcement Learning for Quantum State Design

Proposed Solution: reinforcement learning to design experimental setups in quantum optics



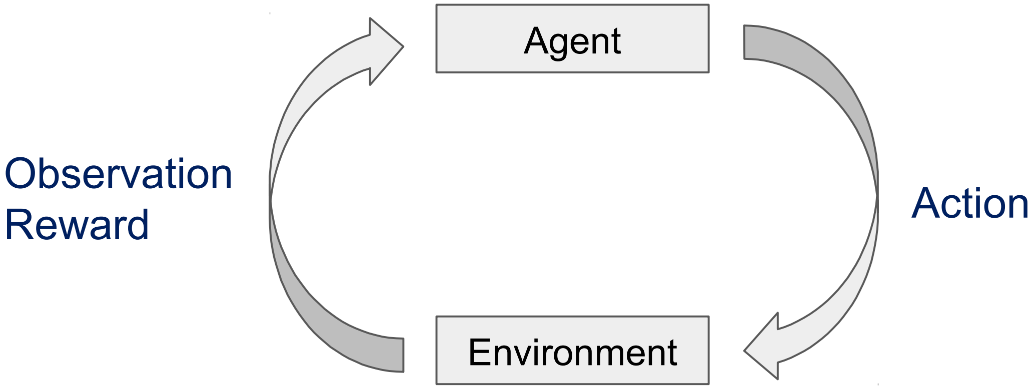
Melnikov, Alexey A., et al. "Active learning machine learns to create new quantum experiments." Proceedings of the National Academy of Sciences 115.6 (2018): 1221-1226.

Reward: a measure of "desirability" of the quantum state – complexity of the experiment

May allow for "smarter" search in the space of possible experiments

Quantum Computing for Reinforcement Learning

Can quantum computing help an agent learn?



Two options: give the agent a quantum computer and use standard RL, or introduce a quantum formulation of RL

Disclaimer: this is a very new field, with quite tentative work but interesting ideas

21. Quantum agent

Option 1: Does giving a quantum computer to a robot help?

Yes: quantum computing can be used for

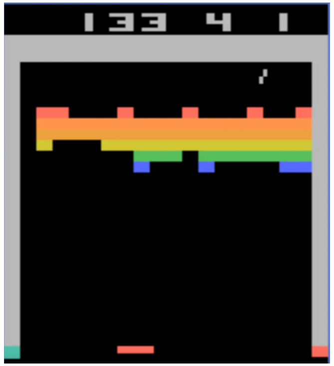
- Improved function approximation
- Faster planning

We will see how function approximation can leverage quantum annealing

22. Agents with classical computers

RL agents are equipped with classical computing tools

Example: DQN agent uses a feedforward neural network and backpropagation to learn to play Atari games. Is this optimal?



Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529.

No:

- Convergence takes time, depends on hyperparameters, and gets stuck in local minima

- Feedforward neural networks may be too simple (think of human brain)

23. Boltzmann machines for RL agents

Alternative to feedforward network: Boltzmann machine

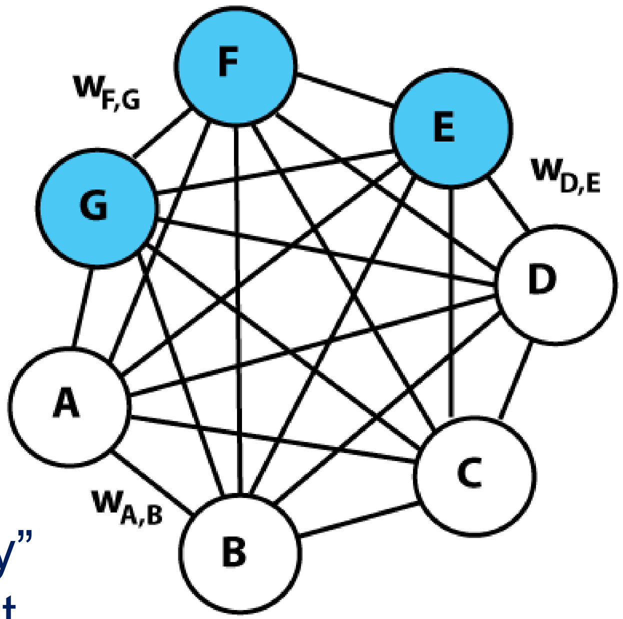
Each node x_i can take values in {0,1}, and to each configuration of nodes we associate an energy:

$$\sum_{i,j} w_{i,j} x_i x_j$$

Where $w_{i,j}$ are the weights of the network

We aim to learn weights such the “equilibrium energy” of the system can be mapped to a quantity of interest

- However, learning these weights is hard



Blue: Visible nodes
White: Hidden nodes

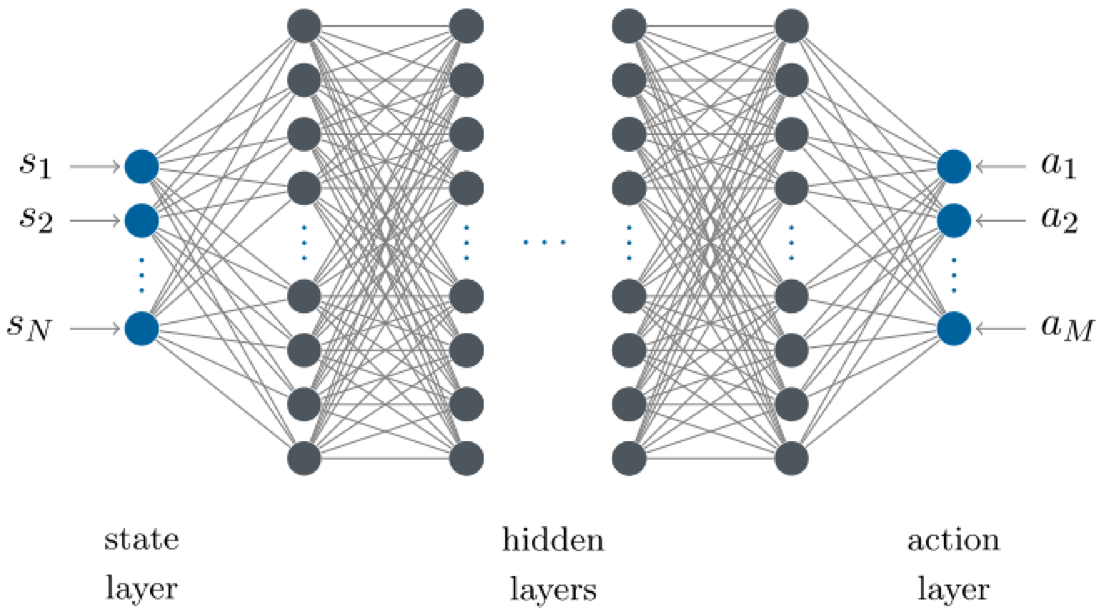
24. Quantum annealing for RL agents

Deep Boltzmann machines can be mapped to quantum annealers

- Reminder: designed to solve optimization problems of the form: $$\sum_{i,j} w_{i,j} x_i x_j$$

- Learn weights such that for each (s,a), the equilibrium energy of the network yields the Q-value

- Use quantum annealing to train these networks



Crawford, Daniel, et al. "Reinforcement learning using quantum Boltzmann machines." arXiv preprint arXiv:1612.05695 (2016).

25. Quantum formulations of RL

Option 2: quantum formulation of RL

Can an agent be entirely quantum?

Interactions with a classical environment would cause a "purely quantum" agent to become classical

- No quantum speedup in terms of environment interactions

What if both agent and environment are quantum?

26. Quantum environment

The environment can now accept a superposition of actions, and can be in a superposition of states

Strange things happen:

- An optimal policy can involve superpositions of states or actions

- Limited access to information about the state of the agent/environment

Is any of this useful? Yes! We can consider training an agent in simulation in a quantum computer (a quantum Atari emulator for example), or model-based RL

27. Quadratic acceleration in sparse reward settings

Problem: In an unknown environment with sparse rewards, finding the first sequence of actions that leads to a reward can be challenging

Solution: use Grover's algorithm to find a rewarding sequence of actions quadratically faster

Requires a quantum oracle for environment rewards:

$$|a_1, \ldots, a_M\rangle \xrightarrow{E^q_{oracle}} (-1)^{\Lambda(a_1,\ldots,a_M)} |a_1, \ldots, a_M\rangle$$ with Λ=1 if actions yield a reward

Limitation: brute-force search for a good sequence of actions

Reference:

Dunjko, Vedran, Jacob M. Taylor, and Hans J. Briegel. "Quantum-enhanced machine learning." Physical review letters 117.13 (2016): 130501.

28. Exponential speedups are theoretically possible

Simon's algorithm: Given a black box function f: $\{0,1\}^N \rightarrow \{0,1\}^N$ for which for some unknown s and all x, $f(x) = f(x \oplus s)$, find s.

Reformulation as a (contrived) RL problem:

- Action: send an input x to the environment OR guess a string s
- States: environment returns f(x) if last action was x, nothing otherwise
- Reward: 1 if last action was to guess s and s is correct, 0 otherwise

Classical agent: exponential number of environment interactions
Quantum agent: linear number of interactions

Reference: Dunjko, Vedran, et al. "Exponential improvements for quantum-accessible reinforcement learning." arXiv preprint arXiv:1710.11160 (2017).

29. Outlook

The intersection between RL and quantum is still relatively unexplored, but:

- Autonomously controlling and designing quantum experiments and protocols is a promising application of RL

- Research into how RL can be quantum-enhanced, despite being quite tentative at the moment, is only just beginning and may yield some useful insights

Quantum computers are still years away, but thinking about this now allows us to both prepare for their arrival and motivate future work

30. Summary and conclusion

I discussed three applications of RL to quantum computing:

- Experimental Control
- Design of error correction protocols
- Generation of quantum states

And different ways in which RL can be quantum-enhanced:

- Improved function approximation using quantum annealing
- Fundamental enhancements with access to a quantum environment

The intersection between the two fields, still largely unexplored, can be a source of inspiration for future work.

31. Who we are

Independent R&D lab focusing on RL and NLP, based in Paris

Our aim: work with academia and industry to bridge the gap between research and real world applications

UncharTECH

https://unchartech.io/

Questions?

UncharTECH

https://unchartech.io/

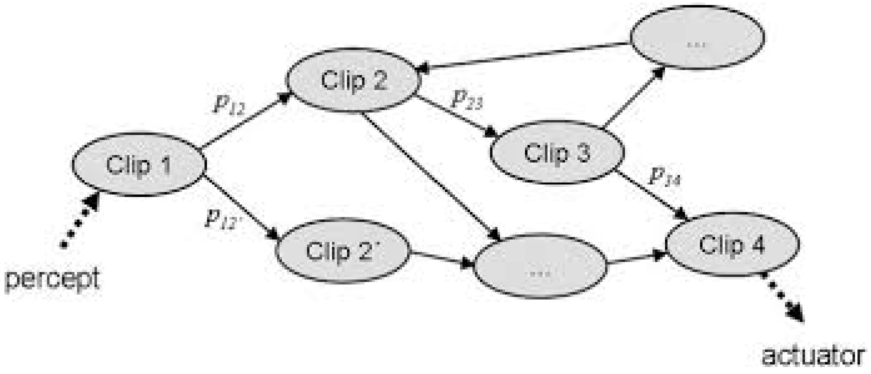
30. Faster planning

Problem: how best to leverage an agent's memory to select an action

Projective simulation framework: an agent's memory is represented by a graph of connected "clips", representing states, actions, sequences of past observations, prior knowledge, ...

Learning operates by changing the weights of the edges in the graph

Action selection is performed by sampling from the stationary distribution over actions induced by the Markov chain defined by the graph



Briegel, Hans J., and Gemma De las Cuevas. "Projective simulation for artificial intelligence." Scientific reports 2 (2012): 400.

31. Faster planning

Solution: use quantum machine learning to calculate the stationary distribution

Classically, the mixing time of a Markov chain is O(1/δ), where δ is the spectral gap between the largest and second-largest eigenvalues.

A Grover-like quantum algorithm can be used to obtain the stationary distribution in O(1/sqrt(δ)): quadratic speedup for action selection

Experimental demonstration with ionic qubits:

Sriarunothai, Th, et al. "Speeding-up the decision making of a learning agent using an ion trap quantum processor." Quantum Science and Technology 4.1 (2018): 015014.

And also with photonic qubits:

Flamini, Fulvio, et al. "Photonic architecture for reinforcement learning." arXiv preprint arXiv:1907.07503 (2019).