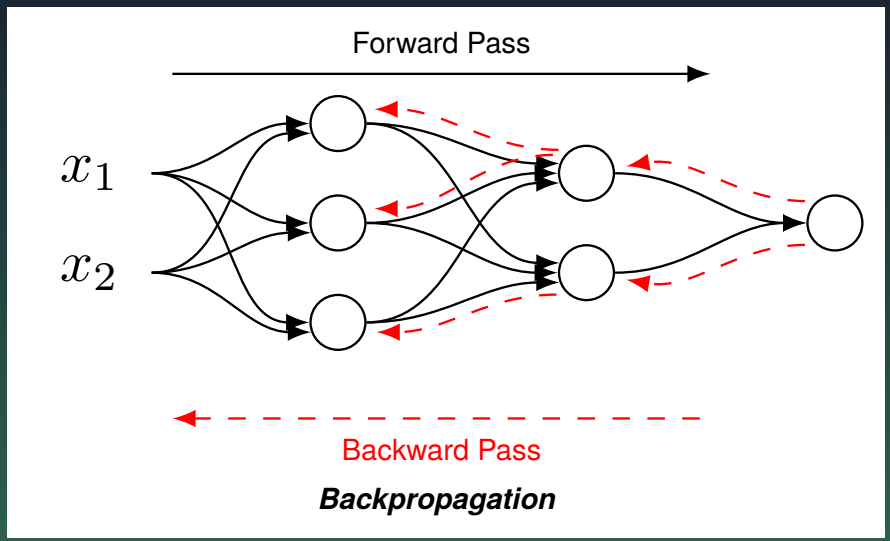
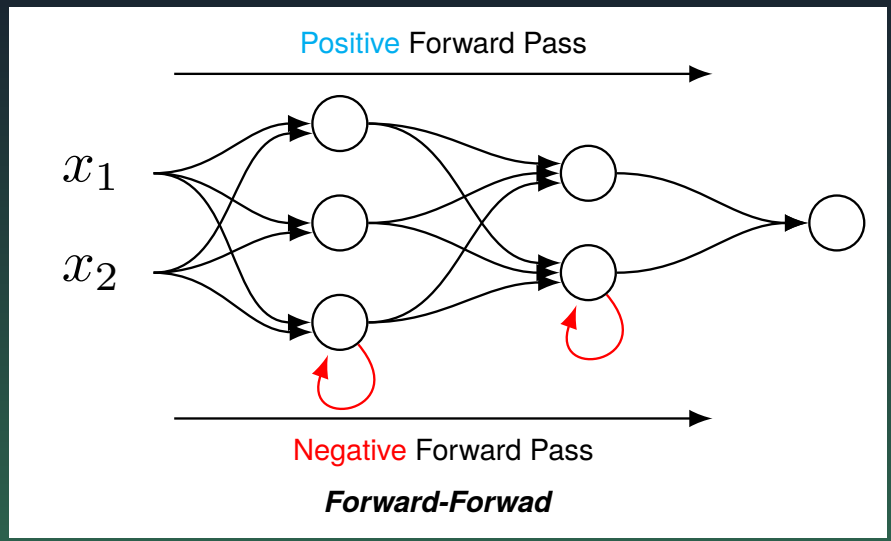
Deep Learning in Quantum Computers

Quantum forward-forward algorithm

Forward-forward algorithm

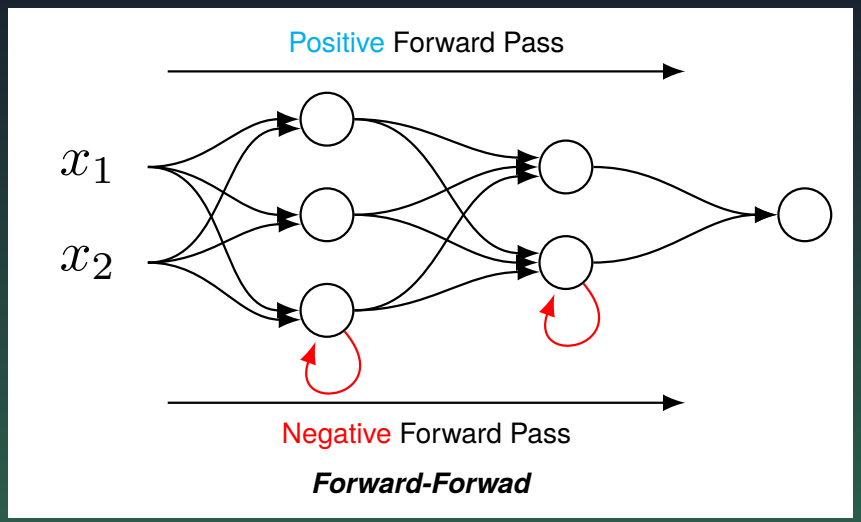
- Forward-forward algorithm (Hinton, 2022)
  - Biological implausibility of backpropagation
  - No explicit propagation of error derivatives
  - Information of neural activities are not stored
- Replaces the forward and backward passes of backpropagation with two forward passes (i.e., positive and negative pass) that operate in the same way as each other
- Applies a contrast learning scheme with different data with opposite objectives





Forward-forward algorithm cont.

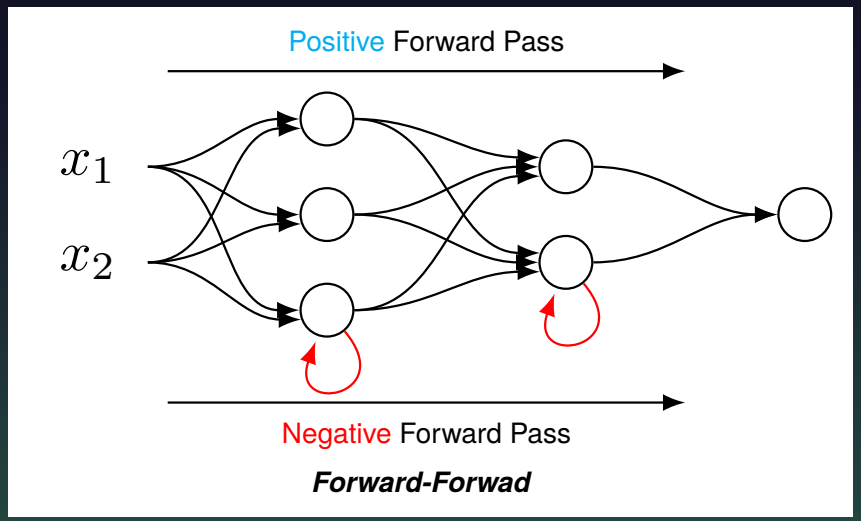
- The positive pass operates on real data and adjusts the weights to increase the goodness in every hidden layer. The negative pass operates on "negative data" and adjusts the weights to decrease the goodness in every hidden layer.
- The probability of the score can be represented as $p(positive\ data) = \sigma(\sum_j y_j^2 - \theta)$, where y_i is the activity of hidden unit j before layer normalization, and θ is the threshold, and σ is the activation function.



Forward-forward algorithm cont.

Advantages:

- Works when the precise details of the forward computation are unknown
- Not required to store the neural activities or stopping to propagate error derivatives



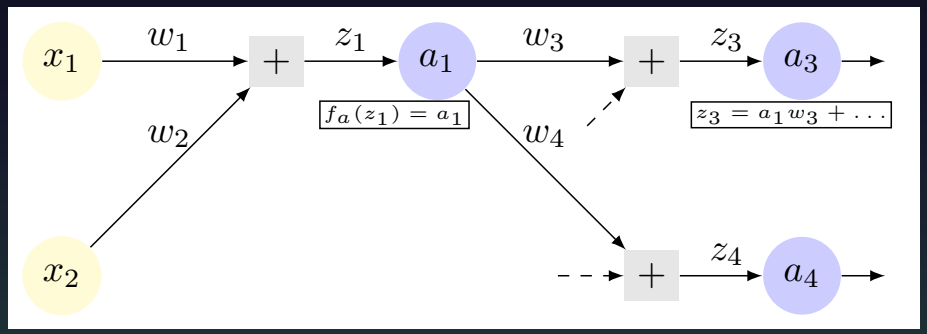
Quantum forward-forward algorithm

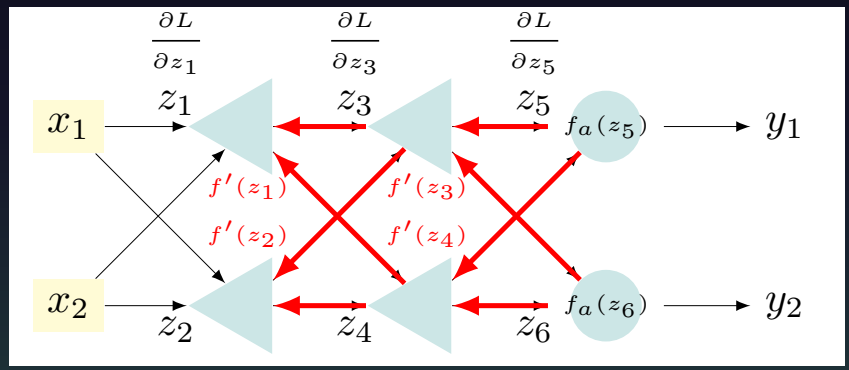
- Backbone: Quantum Boltzmann Machine (Wiebe & Wossnig, 2019)
- Object functions: Minimize the quantum KL-divergence of positive pass and maximize the quantum KL-divergence of negative pass
- The relative entropy is formulated as $S(\rho|\sigma_v) = \mathrm{Tr}(\rho\log\rho) - \mathrm{Tr}(\rho\log\sigma_v)$, where ρ represents data and σ_v represents models
- The preparation of the required Gibbs states and the evaluation of the loss function's analytic gradient can be realized by employing variational quantum imaginary time evolution (Zoufal et al., 2021)

Quantum backpropagation algorithm

Backpropagation algorithm

- Backpropagation algorithm (Rumelhart et al., 1986)
  - More generalized than forward-forward algorithm
  - Nice scaling properties: efficient reuse information and weight-sharing
  - $TIME(f') = 2 * TIME(f)$





$$\frac{\partial L}{\partial \mathbf{W}^{(l)}} = \frac{\partial L}{\partial \mathbf{a}^{(l)}} \frac{\partial \mathbf{a}^{(l)}}{\partial \mathbf{z}^{(l)}} \frac{\partial \mathbf{z}^{(l)}}{\partial \mathbf{W}^{(l)}}$$

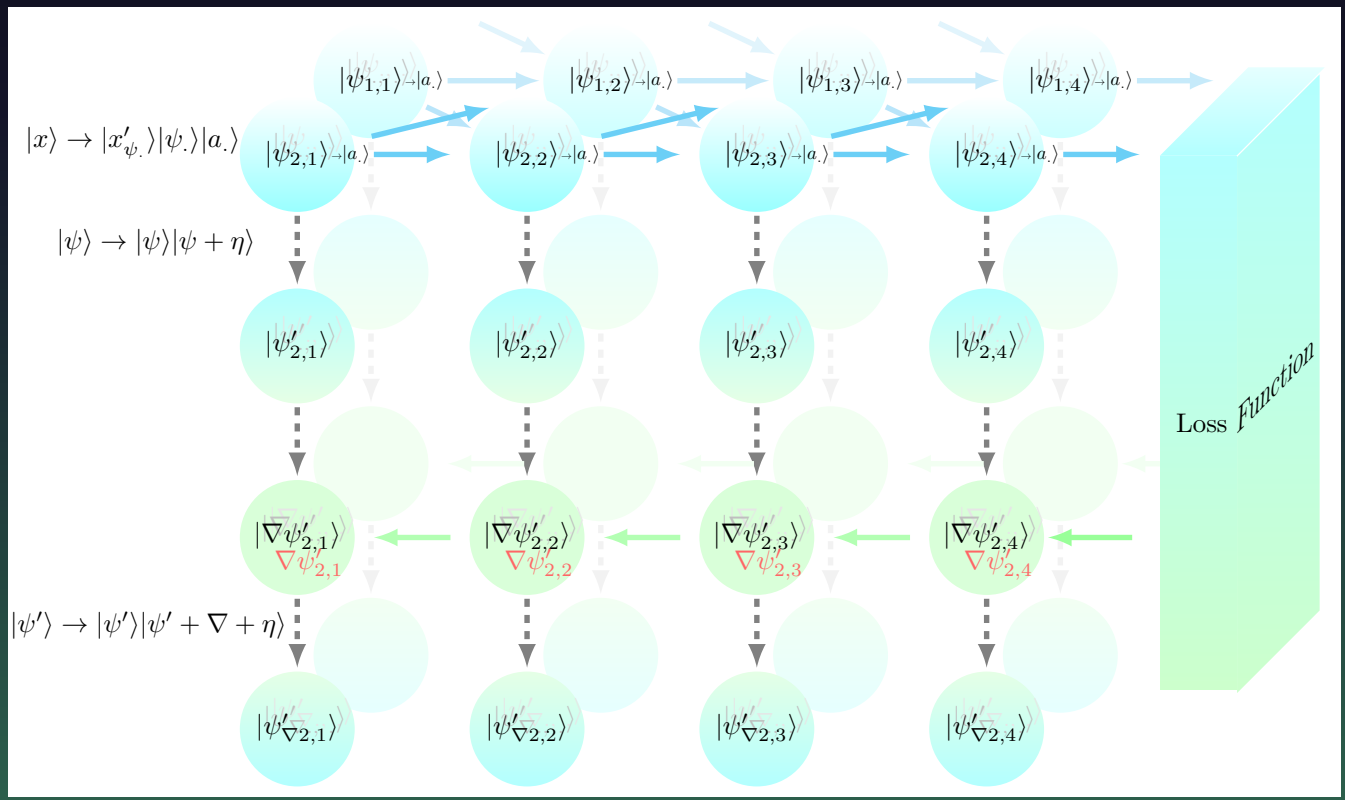
$$\frac{\partial \mathbf{z}^{(l)}}{\partial \mathbf{W}^{(l)}}: \sum x_i, \sum a_i$$

$$\frac{\partial \mathbf{a}^{(l)}}{\partial \mathbf{z}^{(l)}}: f'(z) = \sum f_{a^{(l+1)}}(z_i)$$

$$\frac{\partial L}{\partial \mathbf{a}^{(l)}}: \frac{a_i}{L}$$

Quantum backpropagation algorithm

- Approximate a clone quantum state by applying the formalism of differential privacy (Aaronson & Rothblum, 2019)
- $|\psi\rangle \rightarrow |\psi\rangle|\psi + \eta\rangle$, where $|\psi\rangle$ is the quantum state to be cloned, and η is a Laplace noise term.



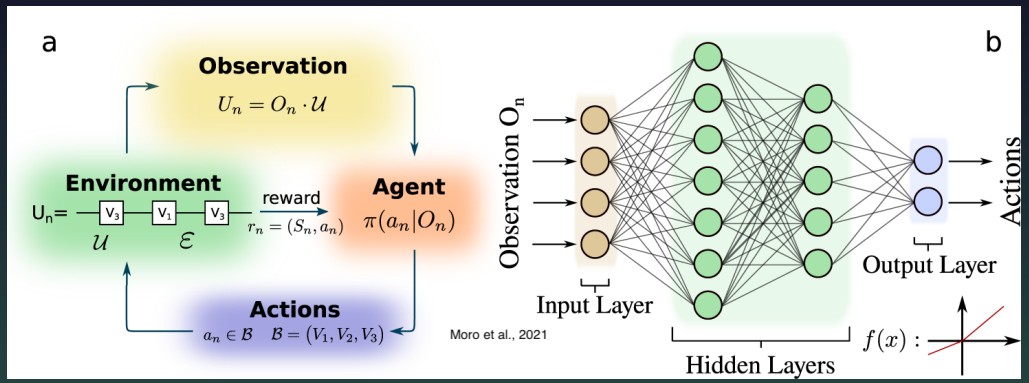
Quantum deep reinforcement learning algorithm

Quantum deep reinforcement learning algorithm

- The reinforcement learning framework (Sutton & Barto, 2018) is a powerful algorithm that helps to train agents to interact with a complex environment in an efficient way to reach given objectives
- Utilize reinforcement learning agents to search optimal quantum circuits (Kuo et al., 2021; Moro et al., 2021)
- Integration of reinforcement learning with variational quantum circuits (Skolik et al., 2022) and quantum Boltzmann Machines (Crawford et al., 2019; Jerbi et al., 2021)

Quantum deep reinforcement learning algorithm cont.

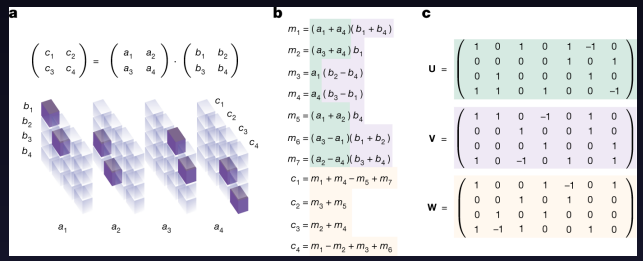
- Applying quantum deep neural network into deep Q-Netowrk algorithm (Mnih et al., 2015) to form a deep Q-Network



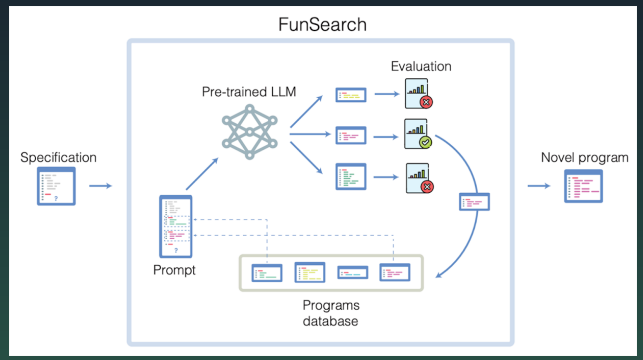
Learning Systems for Quantum Algorithm Design

DeepMind

- AlphaTensor: Discovering faster matrix multiplication algorithms with reinforcement learning (Fawzi et al., 2022)



- Funsearch: Mathematical discoveries from program search with large language models (Romera-Paredes et al., 2023)



Funsearch for quantum algorithm

- Major component of funsearch:
  - Code manipulator
  - Evaluator
  - Sandbox
  - Program database
- OpenAI Assistants API:
  - Tools: code interpreter, function calling
  - Implementation of funsearch:
  - Code interpreter works as sandbox and function calling for implementation of function template and Evaluator
  - The assistant can generate inputs (arguments) for the function template, realizing the evolution of code manipulation
- Specialized for QisKit code generation by looping in IBM's Watsonx.ai and tapping into their Granite foundation models. And running the sandbox in quantum computers
- OpenAI Assistant for Quantum Algorithms/Circuits design

Thank you and questions!