2023 Special Summer Internship

Variational Quantum Algorithms: Challenge or Excuse

July 1 – August 31, 2023

School of Quantum at Korea University

Together with ETRI, KIAS, POSCO Holdings

Learning Contents Guide

Last update: June 23, 2023

Note: This guide was originally suggested by the *Xanadu Communications Team*, and has since been edited by the Organizing Committee.

The content marked as "Explanation" corresponds to text and image explanations of the concepts. The content marked as "Codebook" includes interactive coding exercises. The content marked as "Demo" includes both explanation and code. In demos, the code cannot be executed directly on the page but it can be downloaded as a notebook or Python file. Any of this content can be used to prepare the lectures or as practice or reference material for the students. Each of the lectures is designed for a duration of around 1h30, plus individual practice time for the students to go through the coding exercises.

Lecture 1: Introduction to quantum computing

- What is quantum computing? (Explanation)
- Codebook module I.1: Qubits, bra-ket notation, superposition, and measurements.
- Codebook module I.2: Quantum circuits
- Codebook module I.3: Unitary matrices
- Practice:
    - Codebook codercises

Lecture 2: Quantum gates and variational quantum circuits

- Codebook module I.4: Quantum operations on single qubits
- Codebook module I.5: Global and relative phases
- Codebook module I.6: Rotation gates and Bloch sphere
- Codebook module I.10: Observables and expectation values
- Codebook module I.11: Tensor product and multi-qubit systems
- Codebook module I.12: Entanglement and controlled operations
- Variational circuits (Explanation)
- Practice:
    - Codebook codercises

Lecture 3: Embedding data into quantum circuits

- Embeddings (Explanation)
    - Basis embedding (Documentation)
    - Angle embedding (Documentation)
    - Amplitude embedding (Documentation)

- Ansatz, depth, layers (Explanation)
- How to embed data into a quantum state (Blog)
- Practice:
    - Try out PennyLane's embedding templates (Documentation)

Lecture 4: Optimizing quantum circuits

- Quantum gradients (Explanation)
- Differentiable programming (Explanation)
- Optimizing a quantum circuit with PennyLane (Video)
- Optimizing quantum circuits (Demo)
- Practice:
    - Optimize your own parametrized quantum circuit in PennyLane
    - Try out the RandomLayers, StronglyEntanglingLayers, and BasicEntanglerLayers templates in PennyLane (Documentation)

Lecture 5: Quantum Machine Learning (QML)

- What is Quantum Machine Learning? (Explanation)
- Variational classifier (Demo)
- Quantum Neural Network (Explanation)
- Learning to learn with quantum neural networks (Demo) (Optional)
- Practice:
    - Create a variational classifier for a different dataset

Lecture 6: The Variational Quantum Eigensolver (VQE)

- The Variational Quantum Eigensolver (Demo)
- The Variational Quantum Eigensolver (Video)
- Build molecular Hamiltonian (Explanation) (Optional)
- Practice:
    - Run the VQE algorithm for a different molecule

Lecture 7: The Quantum Approximate Optimization Algorithm (QAOA)

- Intro to QAOA (Demo)
- QAOA for MaxCut (Demo)
- A different perspective to QAOA (video)
- Practice:
    - Create your own QAOA program

[Optional] Advanced Variational Quantum Algorithms (VQA)

- Variational Quantum Thermalizer (Demo): Simulating the VQT for a 4-Qubit Heisenberg Model
- Perturbative Gadgets for Variational Quantum Algorithms (Demo)

Additional resources

- Discussion Forum: Where students can ask questions
- Xanadu Slack: Where students can join the community
- Codebook GitHub repository: Where you can find all of the content except for the answers to the coding exercises
- QHack 2023 coding challenge repository: For additional practice ideas
- **PennyLane installation instructions**

  It would be ideal for the students to have PennyLane installed. They can also use Google Colab if they prefer not to install PennyLane locally.

  If they do wish to install PennyLane locally they can run `python -m pip install pennylane` in their terminal**.** They can also watch this video for detailed instructions.

  I strongly recommend that they create a new virtual environment to prevent them from having any installation problems. They can create a virtual environment with Conda and install PennyLane as follows:

  1. Open their terminal (mac) or command line (Windows).
  2. Create a new Conda environment with: `conda create --name name_of_your_environment python=3.10`
  3. Activate the environment: `conda activate name_of_your_environment`
  4. Install the needed packages: `python -m pip install pennylane jupyter matplotlib`

  Note that they would be installing 3 packages here: PennyLane, Jupyter, and Matplotlib.
  Also, note that where it says "`name_of_your_environment`" the students can choose any name that they want.

  After this they can write `jupyter notebook` in their terminal and they will be ready to create programs using PennyLane (eg: watch this video).