*Quantum Machine Learning for Quantum Data*

# **Author(s)**

· *Name : Koji Terashi*

· *Institution : ICEPP, the University of Tokyo*

· *Email address : koji.terashi@cern.ch*

· *Major : Experimental particle physics (ATLAS)*

· *Name : Ryu SAWADA*

· *Institution : ICEPP, the University of Tokyo*

· *Email address : ryu.sawada@cern.ch*

· *Major : Experimental particle physics (ATLAS)*

# **Challenge Advisor/Mentor**

· *Name : Koji Terashi*

· *Institution : ICEPP, the University of Tokyo*

· *Email address : koji.terashi@cern.ch*

· *Name : Ryu SAWADA*

· *Institution : ICEPP, the University of Tokyo*

· *Email address : ryu.sawada@cern.ch*

# **Summary**

*Machine learning is widely used nowadays in researches and industries. There is a high demand of computing architecture to perform it faster, e.g. GPU, TPU and FPGA. Quantum computer could bring a major breakthrough in speeding up machine learning process by utilizing QRAM - quantum random-access memory - to encode classical data with size N into quantum states with log(N) quantum bits. Most of studies of Quantum Machine Learning (QML) focus on learning classical data. However, QML could be used to learn quantum data, which is too complicated to be learned by classical machine learning. The goal is to demonstrate such an innovative idea with a simple toy model. As a more advanced goal, we could also try to classify simple processes of high-energy particle interactions as simulated in quantum particle showers.*

**1)** **Problem and Motivation**

*There are several implementation of quantum machine learning (e.g. quantum support vector machine in qiskit and quantum circuit learning in qulacs). Those implementations assume that their inputs are classical. Such input data are usually from a classical system, e.g. kinds of flowers etc. On the other hand, we could have much higher performance using quantum data as inputs and learning a quantum system. Using quantum machine learning, we could make the model to represent much more complicated system which may not be possible by classical models. For example, quantum many-body problems may be better to be learned by a quantum model rather than a classical mode.*

**2)** **Approach**

*We prepare a simple quantum circuit A, which is assumed to be a black box in the real world. The circuit A is classical in normal quantum machine learning problems, but we will try using a quantum circuit instead. We prepare another quantum circuit B using, e.g, a set of rotation gates with angles 𝜃 as parameters. For each measurement of output states from the circuit A and B, we compare the measurements and apply a feed-back to the circuit B by optimizing the 𝜃 parameters in a classical way. This process is iterated until the circuit B behaves like the original circuit A. By this way, we could make a quantum circuit B, which will simulate the circuit A without knowing the quantum logic in A. The proposed quantum circuits are schematically shown in the right figure. Mentors prepared a very simple circuit A with a randomly generated input numbers; participants can modify the circuit A to solve another problem. After training for regression, the circuit B may emulate the circuit A.  
  
  
Step 1*

1. *A simple toy-model is first considered for the circuit A, e.g, a combination of a few 1-qubit quantum gates. An example code of the circuit A is prepared by mentors. The input to the circuit is a (randomly generated) classical number. A classical number can be encoded in a quantum state to be an input to the circuit A. The circuit B is constructed using an algorithm called quantum circuit learning (QCL) - a quantum machine learning algorithm to learn input data by optimizing circuit parameters. Input quantum state to the circuit A and B is common; i.e. a classical number is encoded by a common quantum. The measurement of output states from the two circuits could be performed with, e.g, a simple Pauli-Z gate. The cost of the model is calculated classically from the measurements of the two circuits A and B.  In this Step A, we aim to make the circuit B to emulate the circuit A; we will therefore solve a regression problem. The cost function can be, for example, mean-squared-error. The results can be visualized by plotting on a 2D plane (the input and the output classical values on the horizontal and vertical axes respectively) from the both circuits.*
2. *An interesting extension of this step is to investigate whether we can figure out the input quantum state as well as the logic in circuit A. For this problem, the input is a quantum state instead of an encoding of a classical number. The input state (e.g, superposition of 0 and 1 states) is provided. The question in Step 1B is whether it is possible to extend the circuit B to learn the input state as well (e.g, the relative fractions of 0 and 1 states). This is probably challenging as the teacher data is now a quantum state. Participants will think how to define the cost, e.g. from the difference in the measurements like 1A or calculating it using another quantum circuit.*

*Step 2*

1. *A more interesting (and realistic) toy-model to learn is a simple quantum system composed of fermions and bosons. Based on arXiv:1904.03196, a quantum circuit has been prepared by mentors to simulate boson emission from fermion particle. This model considers two types of fermions (f1, f2) and one type of boson (𝜙). Starting with a single fermion state |f* 〉*(f1, f2 or their superposition), the circuit can simulate multiple 𝜙 emissions from the fermion (above figure shows the circuit with at most 10 emissions for 𝜙 boson). By allowing the coupling of 𝜙-f1-f2, the intermediate fermion state exhibits interference between f1 and f2, which is simulated in a classical approach only approximately. This fermion-boson circuit could be used as a circuit A, and another circuit B which emulates the circuit A could be modeled using QCL. The input of the circuit may be classical data (i.e, relative fractions of f1 and f2).*
2. *Along the same line in Step 1, the 2A could be further extended by learning the input state, e.g, how the f1 and f2 are superpositioned in the input state. It would be interesting if we can demonstrate that a quantum state can be learned directly.*

# **3)** **Resources**

*https://arxiv.org/abs/1606.02318*

*https://arxiv.org/abs/1612.02806*

*https://arxiv.org/abs/1810.10506*

[*https://arxiv.org/abs/1904.03196*](https://arxiv.org/abs/1904.03196)

*Mentors prepared example code,*

*https://github.com/rsawada/qc\_hackathon\_2019*

# **4)** **Tools**

*Qiskit: Open-source quantum computing framework*

*Qulacs: Super-fast quantum circuit simulator*

*Mentors prepared example code using Qulacs. But participants can try to code using Qiskit. A method of running the Qulacs example code is provided by mentors.*

# **5)** **Results**

*We expect to achieve implementing quantum-data machine learning by using existing tools as building blocks. A goal of Step 1A is to make a circuit B emulating the circuit A. Participants will try to think how to evaluate/visualize the performance of the model, and how to improve the performance. A goad of Step 1B is to show if input quantum states can be learned. After the Step 1A, participants may choose to try Step 2A (instead of 1B). The goal of Step 2 is the same as the Step 1, but with more realistic use-case. For the Step 2, mentors prepared an example application (i.e. parton-shower simulation), but participants can use or build own applications.*

# **6)** **Participants:**

*Graduate students [mathematician, physicist, computer scientist…], undergraduate [mathematician, physicist, computer scientist…], PHD [mathematician, physicist, computer scientist…], Postdoc physicist [mathematician, physicist, computer scientist…]*

# **7)** **References**

*https://qiskit.org*

*http://qulacs.org*