Semiquanduqtol

Semi - QUantum Algorithm for Neuromorphic Design
Using QuanTum Optimization and Learning

Team Qbrain
김자은 Jaeun Kim (Presenter)
박경근 Kyoung Keun Park
이경민 Kyoungmien Lee
이정훈 Jeonghoon Lee

Content

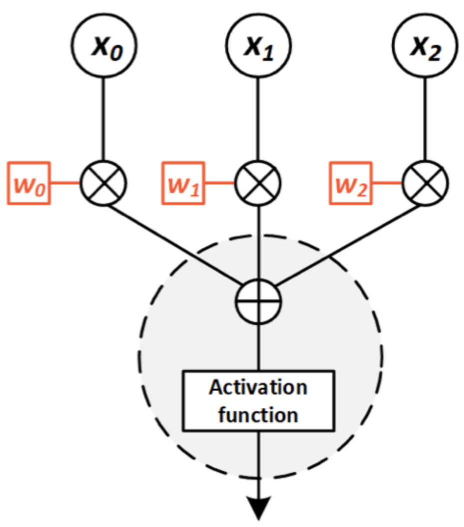
Motivation

Those pictures are from https://snntorch.readthedocs.io/en/latest/tutorials/tutorial_2.html

Artificial Intelligence has been very successful in the fields such as computer vision.
Its success relies on the idea of imitating neurons, a powerful computing mechanism.
Can we make neural networks even more powerful by building models that have a better resemblance to neurons?
Spiking Neural Network (SNN) is a novel approach to building neural networks. It closely resembles how our brain cells work.
While conventional neural networks represent data in matrices of numbers, SNN represents data in a time sequence of pulses.
It is anticipated to be more energy-efficient than conventional neural networks.

References: (we recommend reading the "Introduction" paragraph on this page)
[*https://snntorch.readthedocs.io/en/latest/*](https://snntorch.readthedocs.io/en/latest/)

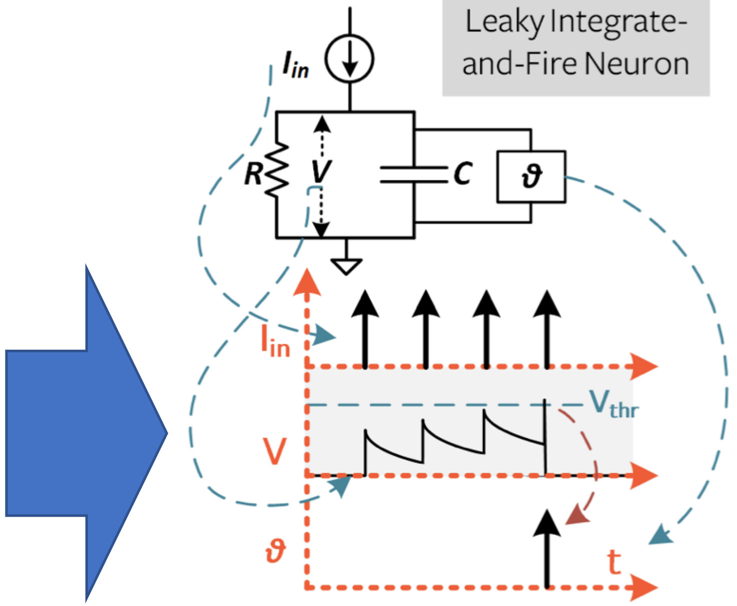
Artificial Neuron



Takes the weighted sum of inputs to produce an output (or activation). No time-varying dynamics or spiking.

Conventional neural networks

Leaky Integrate-and-Fire Neuron



When an input current is applied, the membrane voltage increases with time until it reaches a constant threhsold V_{th} at which point a delta function spike occurs, and the membrane is reset. ϑ is used as the output signal; membrane potential is an internal state.

SNN

Project Goal

End to end Neuromorphic (AI semiconductor) designing using quantum Optimization algorithm and hybrid quantum machine learning

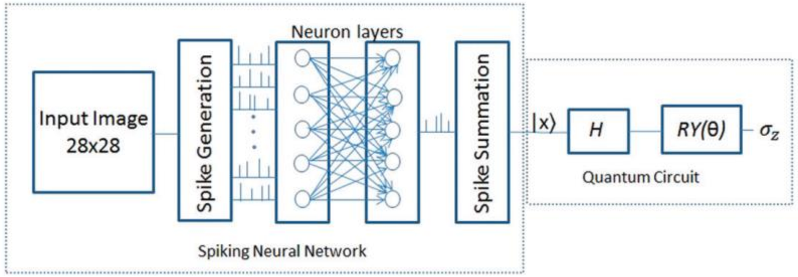
- Part 1 : Training Hybird Quantum Spiking Neural Network

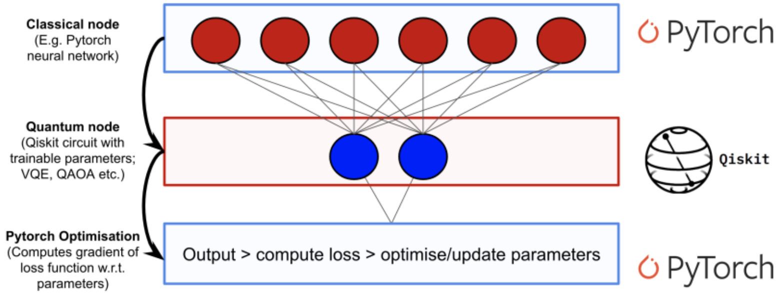
- Part 2 : Mapping SNN to Neuromorphic Hardware
    - Part 2a : Cluster SNN to smaller graph
    - Part 2b : Map Cluster to Neromrophic Crossbar

Part 1 [Ref #1]

Training Hybird Quantum Spiking Neural Network

- The overall structure basically follows the qiskit's Hybrid QCNN network (https://qiskit.org/textbook/ch-machine-learning/machine-learning-qiskit-pytorch.html) & the structure proposed in the reference [2] (They are very similar)
- The main difference is actually the classical model part (CNN vs SNN)
- SNN samples data from image (Data with time information)
- We feed-forwards the spike data to SNN
- The output of SNN is also spike data
  ➔ (T, 2) tensor where T is the number of time steps
- We sum up these spike data ((T,2) -> (1,2) by summing up)
- Use the output as a single qubit rotation angle (Ry)
- Use Z expectation value to classify MNIST digits
- Data : MNIST with 2 classes(3 & 6)





- We want to properly train this QSNN network
  & Plan to use IBM's real quantum device with QEC(e.g. CSS code)
- Possible advantage of QSNN : SNN also has probabilistic nature (Sampled data / Can treat sequence data
  ➔ Can generate sequence like RNN / …). Therefore, we expect that there combination is somewhat natural and quantum computer will handle such property of SNN well.

Part 2a [Ref #2]

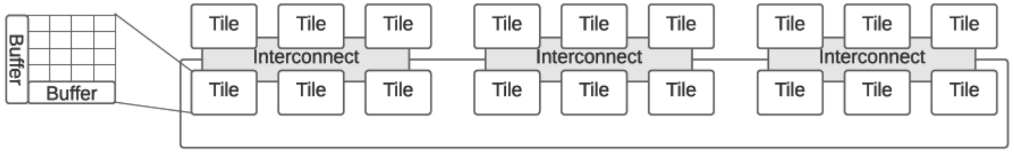
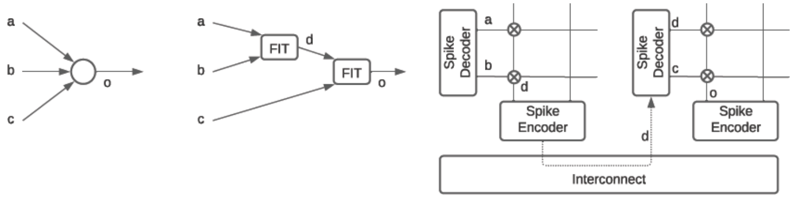


Figure 1

Cluster Spiking Neural Network graph to smaller graphs

- Cross bar based neromorphic is consist of crossbar and interconnect like upper figure 1
- Crossbar based neromrophic has limited amount of hardware, full SNN graph have to be divided(clustered) to several small graph like figure 2(a) to (b) which works like (c)
- Problem is divide graph to smaller graph like figure 3



(a) Original Neuron (b) Decomposed Neuron (c) Mapping the decomposed neuron to two crossbars

Figure 2

Problem formulation

objective

$$x^T L_G x = \sum_{(i,j)\in G} w(i,j)(x_i - x_j)^2$$

$$\begin{aligned}
\text{subject to} \quad & \sum_{i=1}^{n} x_{i,j} = \frac{n}{k} && j = 1,\ldots,k \\
& \sum_{j=1}^{k} x_{i,j} = 1 && i = 1,\ldots,n \\
\text{with} \quad & x_{i,j} \in \{0,1\}, && i = 1,\ldots,n.\ j = 1,\ldots,k
\end{aligned}$$

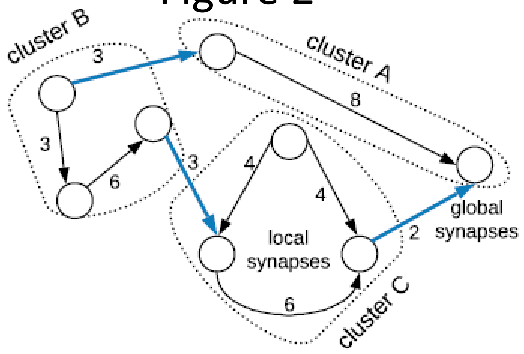


Figure 3

Optimization Method : Quantum Approximate Optimization Algorithm (QAOA)

- Express constraint binary quadratic optimization problem to quadratic unconstraint binary optimization
- Use Qiskit to express QAOA circuit
- Find Optimize value using that circuit

Part 2b [Ref #2]

Map Cluster to Crossbar based Neuromorphic hardware
- minimize spike congestion on shared interconnected
- to minimize energy consumption and spike latency

Problem formulation

Objective : minimize (total spike #) x (Spike moving distance)

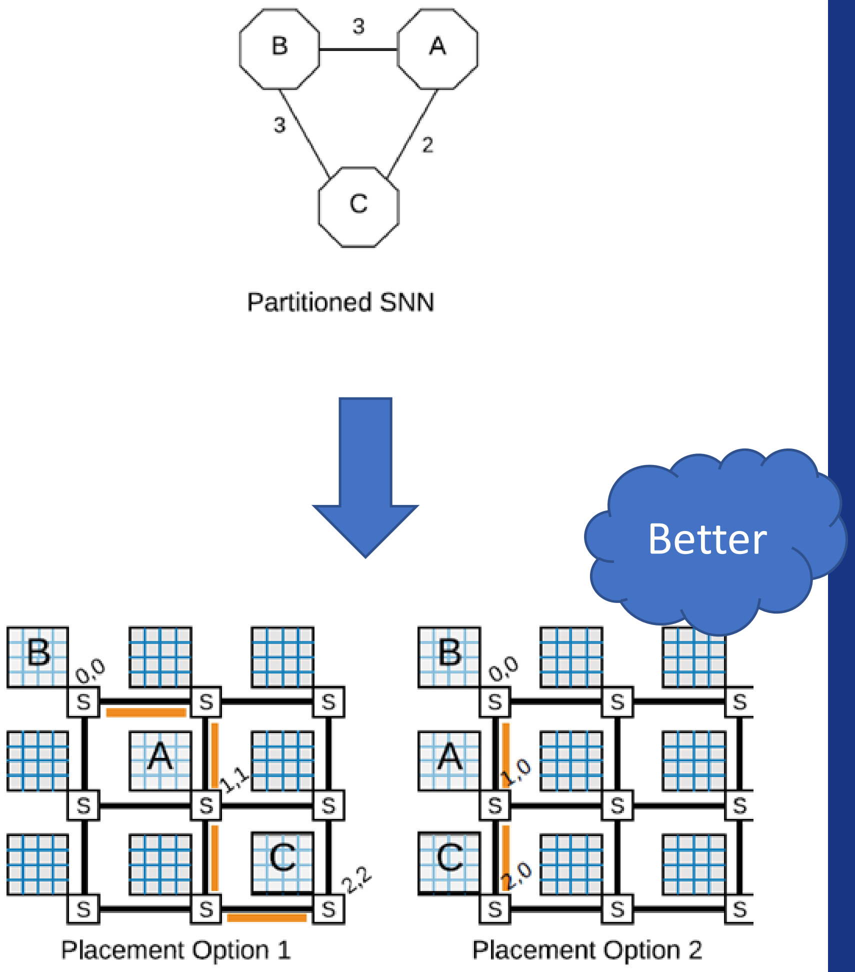
Subject to

$$\sum_i m_{ij} \leq 1 \quad \forall j$$

$$\sum_j m_{ij} = 1 \quad \forall i$$

$$m_{ij} = \begin{cases} 1 & \text{if cluster } c_i \in \mathcal{C} \text{ is mapped to crossbar } v_j \in \mathcal{V} \\ 0 & \text{otherwise} \end{cases}$$

Optimization Method : Durr –Hoyer algorithm [ref #3]
- Build quantum circuit that can execute Durr Hoyer using Qiskit
  - Build oracle that can express unconstraint objective function
  - Build Grover algorithm in Qiskit



Result and Conclusion

Simulation Result
- show Hybrid Quantum SNN can propose classification of MNIST
- show Quantum optimization algorithm can design neuromorphic of given SNN

Conclusion

- Quantum algorithm can design neuromorphic

Future work

In this Project : Topic is to design Classical Neuromorphic helped by quantum

Future : Quantum Neuromorphic

- Hybrid quantum network -> Fully quantum network

- Quantum Scale graph mapping using quantum algorithm

References

[1] A Ajayan and A P James, "Edge to quantum : hybrid quantum-spiking neural network image classifier"

[2] Adarsha Balaji et al, "Mapping Spiking Neural Networks to Neuromorphic Hardware"

[3] Christoph Durr and Peter Hoyer, "A quantum algorithm for finding the minimum"

Thank You