Quantum sea - Classifying water molecules and sodium ions in protein structures

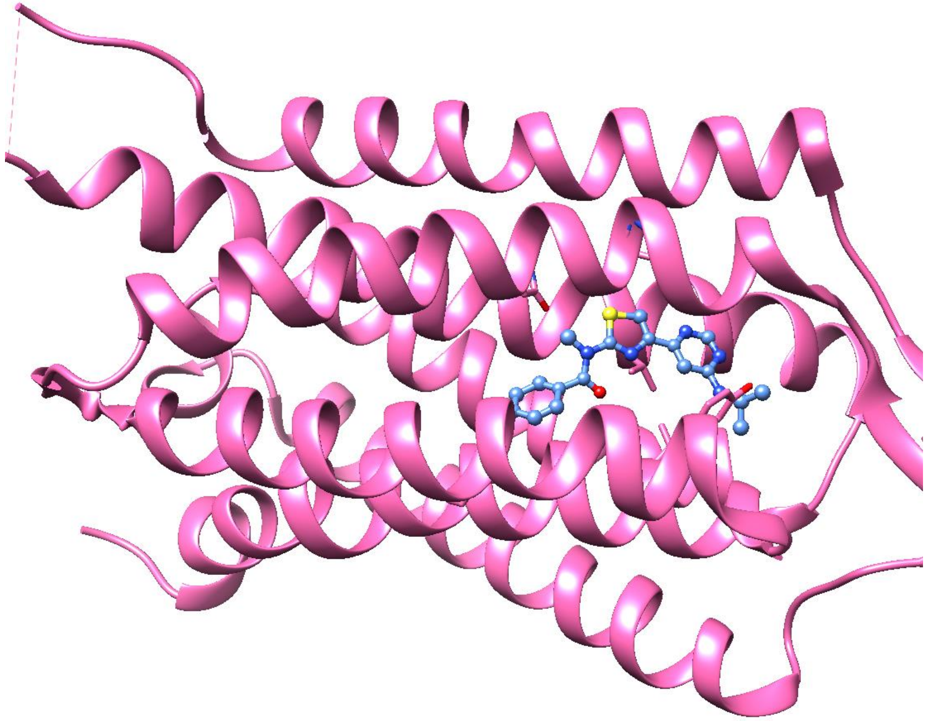
Team Name: Lindwurm (Sangwoo Park)

shadow1229@snu.ac.kr (or shadow1229@naver.com)

Protein is also a molecule.

- Even though protein shows incredible functions, but it is still a molecule.

- Understanding the physics behind protein would help to understand its function.

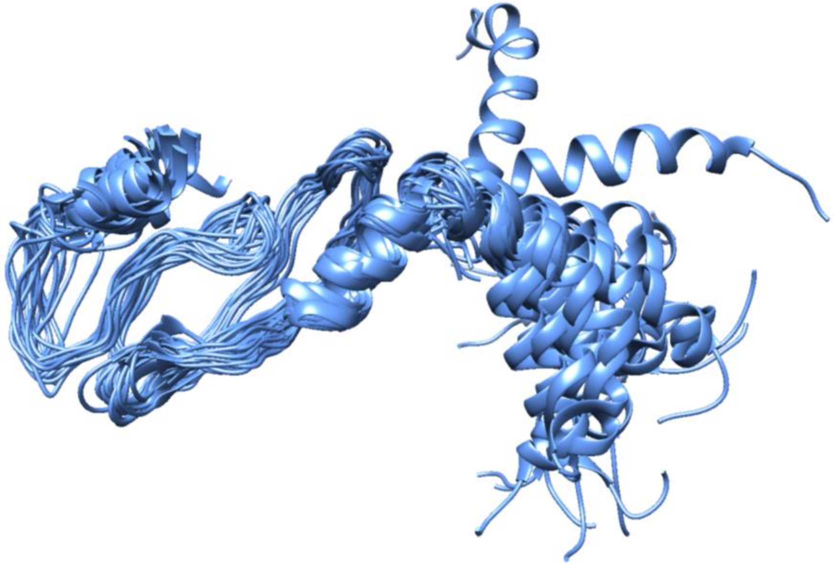


GPCR in complex with a negative allosteric modulator (PDB ID: 4OR2)

Understanding physics behind the protein would help to know interactions / dynamics of the protein.

Why we need quantum computer?

- Interaction of the protein is governed by Gibbs free energy.
  - To know Gibbs free energy, we should know the entropy term.
  - It means, we should know every possible state of the protein. Quite exponential!
- Maybe we need new method which is exponentially faster than classical ones!

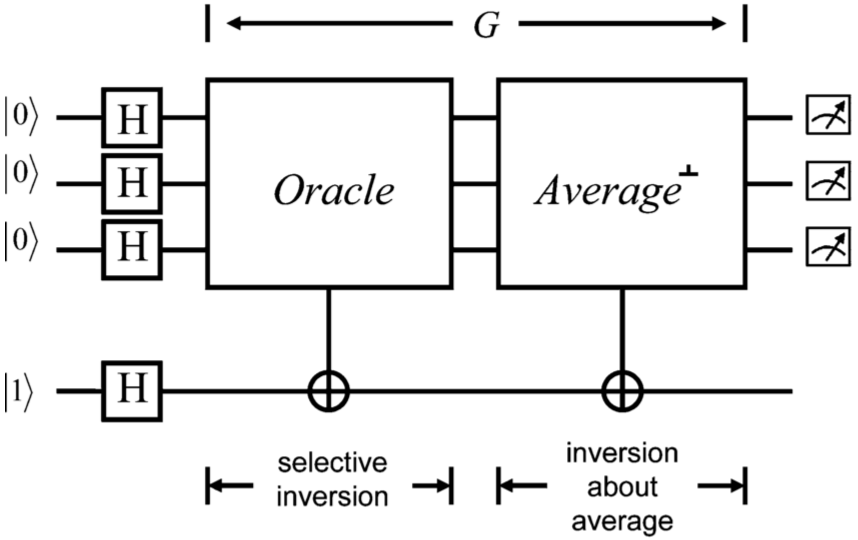


Protein could have a lot of structural states.
(PDB ID: 2L27, NMR structure)

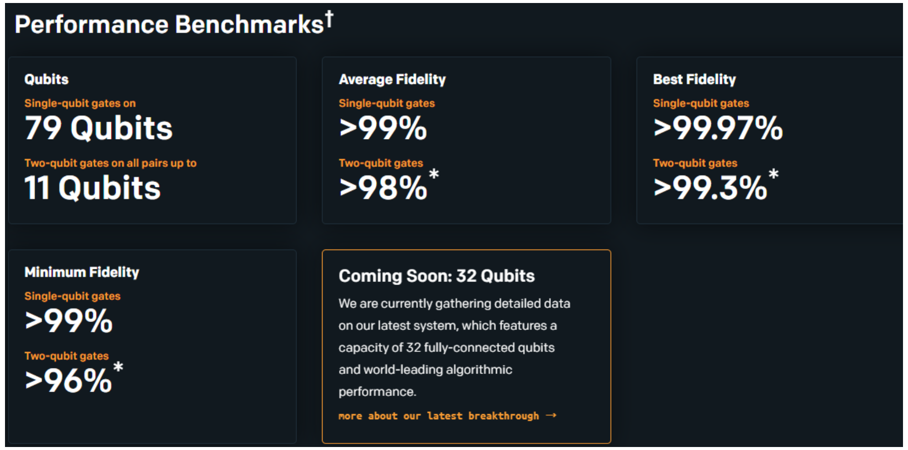
However, It’s bit early.

- Correct algorithm for predicting free energy of the protein…
  - Would use full superposition like Grover search…?
  - Actually we don’t have an algorithm yet!

- Fidelity / # of qubits are not enough for the prediction… yet!
  - Variational Quantum Eigensolver, Quantum Machine learning is available though…



Ju, Yi-Lin & Tsai, I-Ming & Kuo, Sy-Yen. (2007). 10.1109/TCSI.2007.907845.



Performance Benchmarks†

Qubits
Single-qubit gates on
79 Qubits
Two-qubit gates on all pairs up to
11 Qubits

Average Fidelity
Single-qubit gates
>99%
Two-qubit gates
>98%*

Best Fidelity
Single-qubit gates
>99.97%
Two-qubit gates
>99.3%*

Minimum Fidelity
Single-qubit gates
>99%
Two-qubit gates
>96%*

Coming Soon: 32 Qubits
We are currently gathering detailed data on our latest system, which features a capacity of 32 fully-connected qubits and world-leading algorithmic performance.
more about our latest breakthrough →

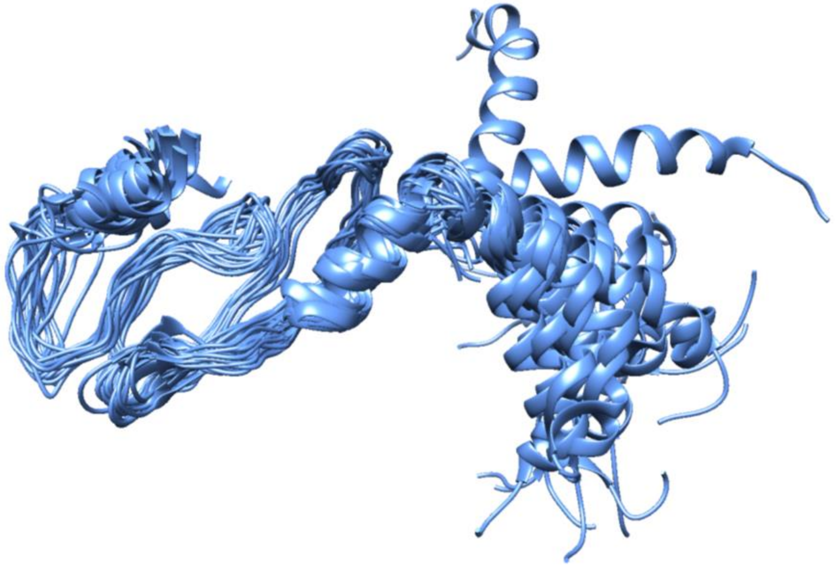
It’s state-of-the-art, but we need more!
(From IONQ’s website.)

Lots of limitations

- Limitations of quantum computer
- Limitations of time (5 days)
- Limitations of myself
  - I think I’m still a beginner in QC area. (sorry!)
  - Scoring 7500 pts in QHack coding challenge might not enough…
- In this presentation, I’ll try to solve toy problem.
  - With Quanvolutional Neural Network

The toy problem: Classifying H_2O / Na^+

- X-ray crystallography:
  - The method for obtaining high-res structure of a protein
  - Uses diffraction of X-ray due to electrons in the protein
  - small molecules, atoms or ions with the same number of electrons are likely to produce similar peaks.
    - H_2O : 10 electrons, Na^+ : 10 electrons
- However…
  - H_2O: 0 net charge / Na^+ : positive net charge
  - H_2O binding structure != Na^+ binding structure

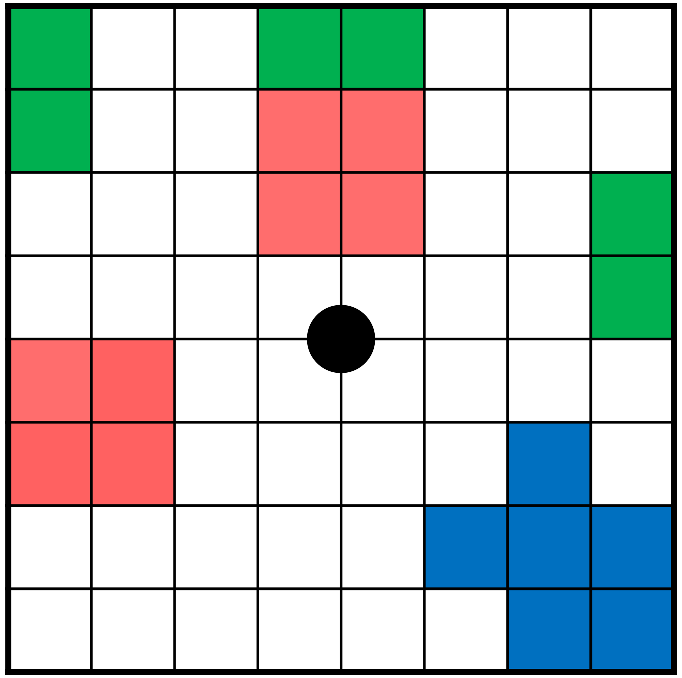


Please forget this NMR model. I will use X-ray crystallographic models only.

The toy problem: Classifying H_2O / Na^+

- The problem in this project:
  - Input: Voxelized 3D image of the structure of C,N,O atoms from proteins or other compounds exclude water, where H_2O / Na^+ exists at the center.
  - Size of the input: A cube size of 16Å and grid spacing of 0.5Å.
  - Output: which one exists at the center, Water, or sodium ions?

Example



Carbon
Oxygen
Nitrogen
Water molecule or sodium ion

The toy problem: Classifying H_2O / Na^+

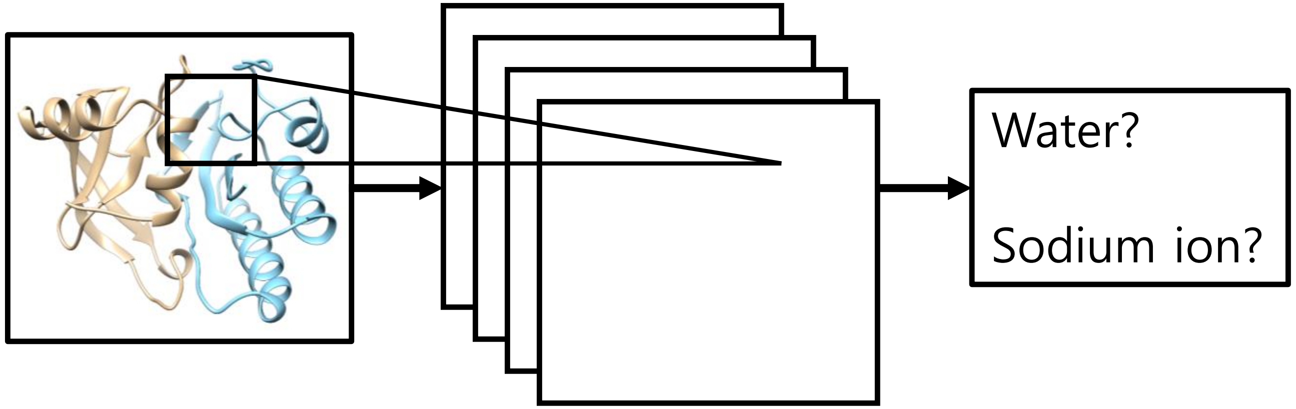
- Training / Validation set:
  - Training set: protein structures nearby 400 water molecules and 400 sodium ions, gathered from PDBBind v2019 refined set.
  - To train roatational invariance, 100 rotations were done for each structure (total 80000 structures. 800 structures per epoch)
    - (Inputs for epoch N == Inputs for epoch N+100)

The toy problem: Classifying H_2O / Na^+

- Training / Validation set (cont.):
  - Validation set: protein structures nearby 54 water molecules and 54 sodium ions, gathered from PDBBind v2019 refined set.
  - Yes, this set is not big enough and it contains some low resolution structures too.
    - No additional R-free, B-factor cutoff were applied
    - Need larger set for sodium ions. (This is the major reason of small set size.)

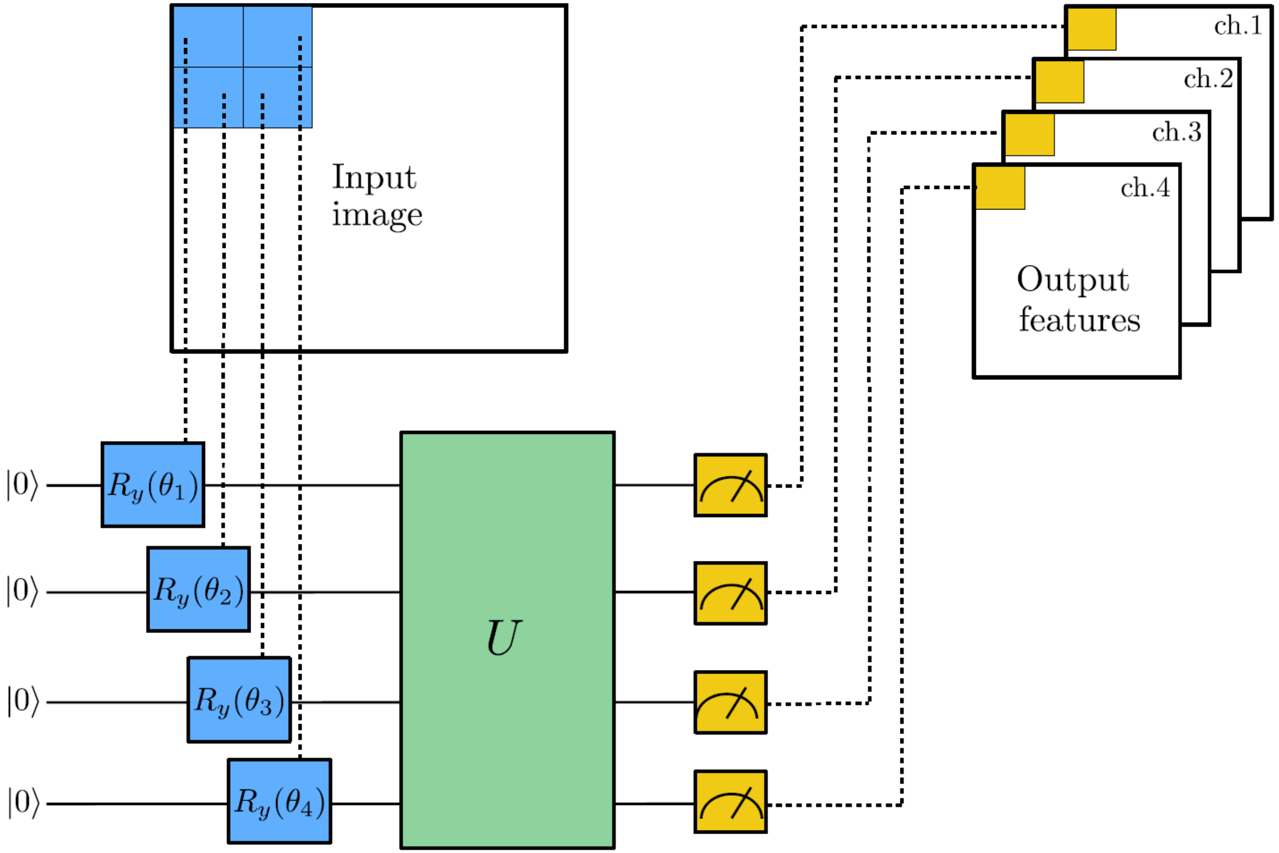
Pattern recognition for protein structures

- One simple way: using convolutional neural network



- And Quanvolutional neural network will be used for QML version!

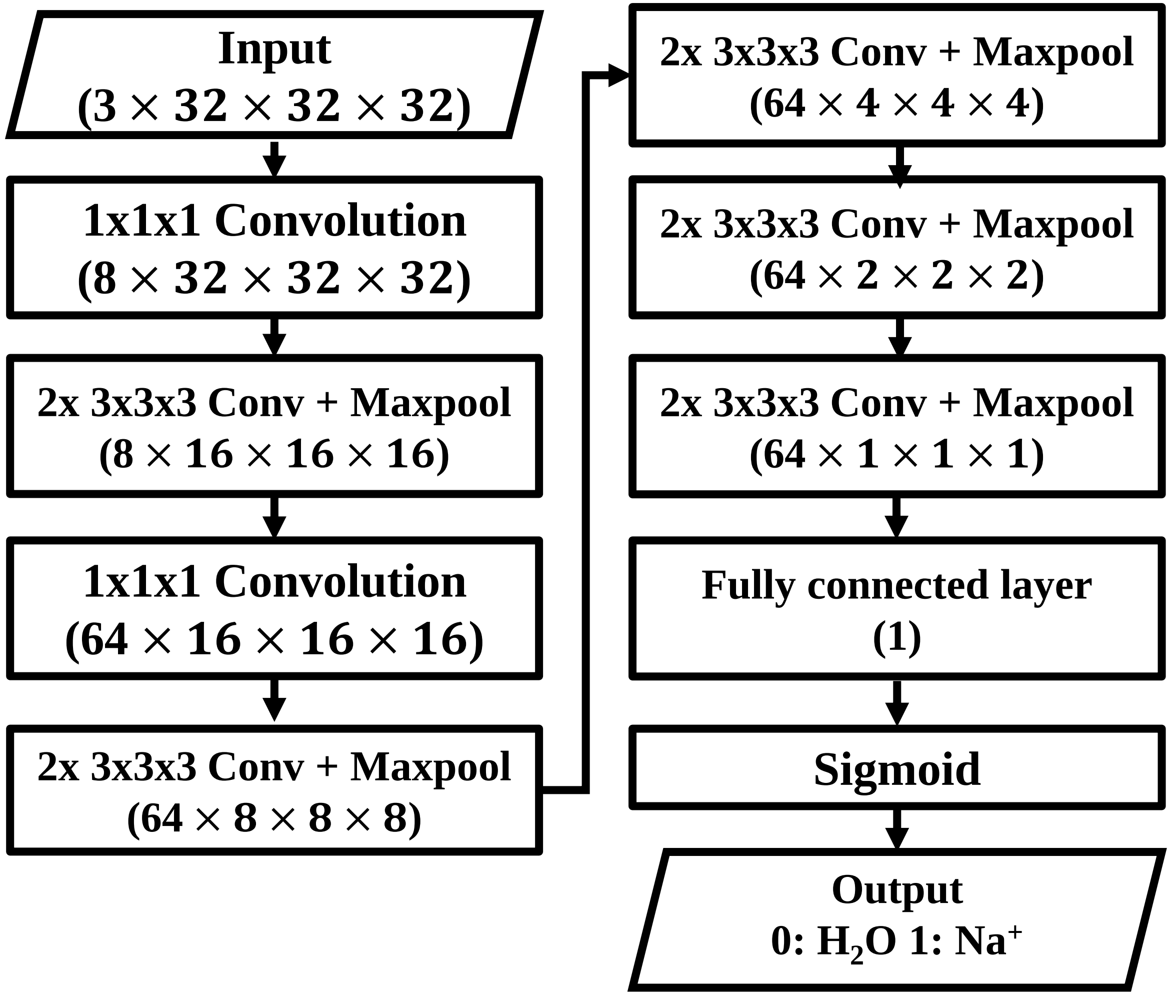
Quanvolutional neural network



From pennylane’s tutorial:
https://pennylane.ai/qml/demos/tutorial_quanvolution.html

Networks used in this project (1/3)

CNN:



Input
(3 × 32 × 32 × 32)

1x1x1 Convolution
(8 × 32 × 32 × 32)

2x 3x3x3 Conv + Maxpool
(8 × 16 × 16 × 16)

1x1x1 Convolution
(64 × 16 × 16 × 16)

2x 3x3x3 Conv + Maxpool
(64 × 8 × 8 × 8)

2x 3x3x3 Conv + Maxpool
(64 × 4 × 4 × 4)

2x 3x3x3 Conv + Maxpool
(64 × 2 × 2 × 2)

2x 3x3x3 Conv + Maxpool
(64 × 1 × 1 × 1)

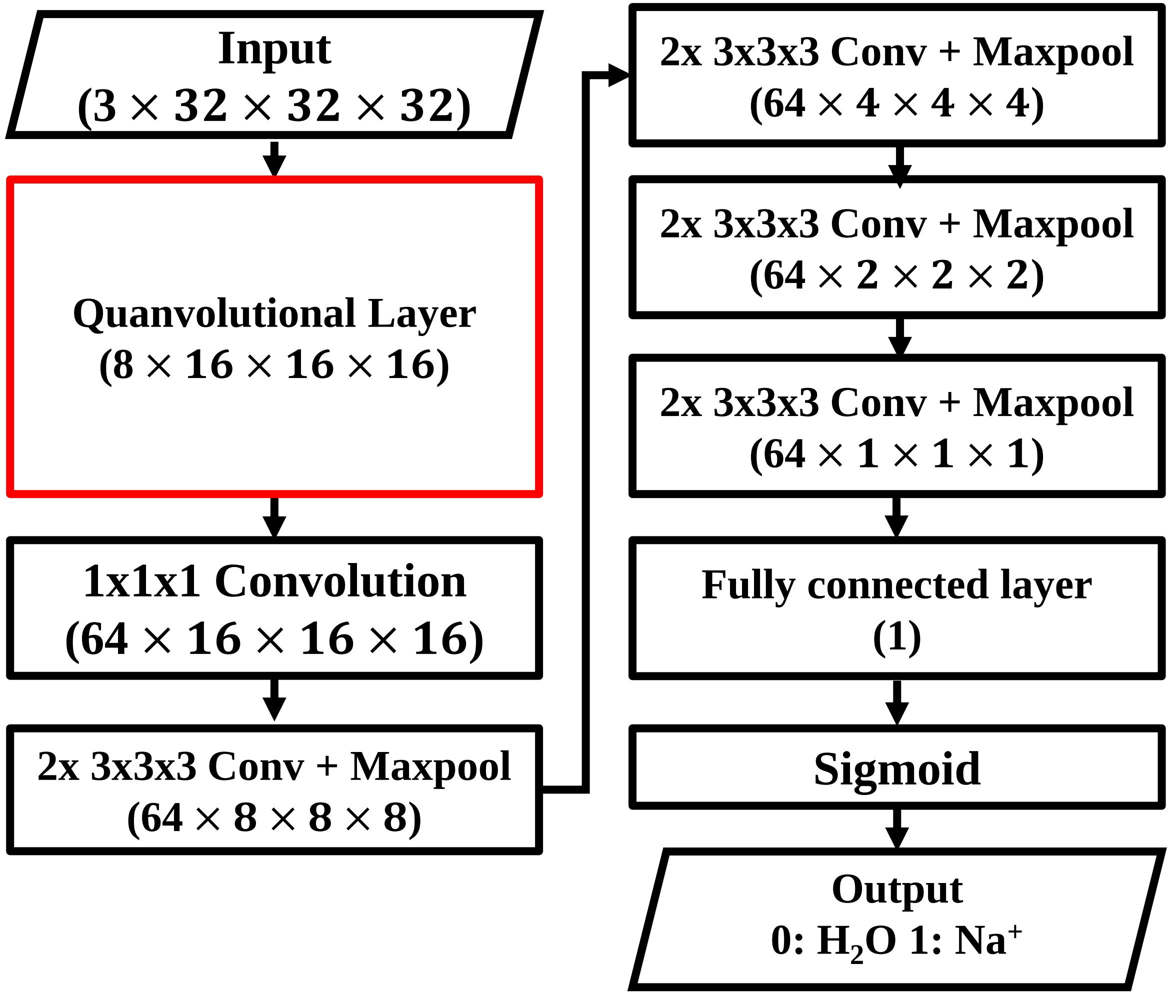
Fully connected layer
(1)

Sigmoid

Output
0: H_2O 1: Na^+

Networks used in this project (2/3)

Quanvolutional Neural Network:



Input
(3 × 32 × 32 × 32)

Quanvolutional Layer
(8 × 16 × 16 × 16)

1x1x1 Convolution
(64 × 16 × 16 × 16)

2x 3x3x3 Conv + Maxpool
(64 × 8 × 8 × 8)

2x 3x3x3 Conv + Maxpool
(64 × 4 × 4 × 4)

2x 3x3x3 Conv + Maxpool
(64 × 2 × 2 × 2)

2x 3x3x3 Conv + Maxpool
(64 × 1 × 1 × 1)

Fully connected layer
(1)

Sigmoid

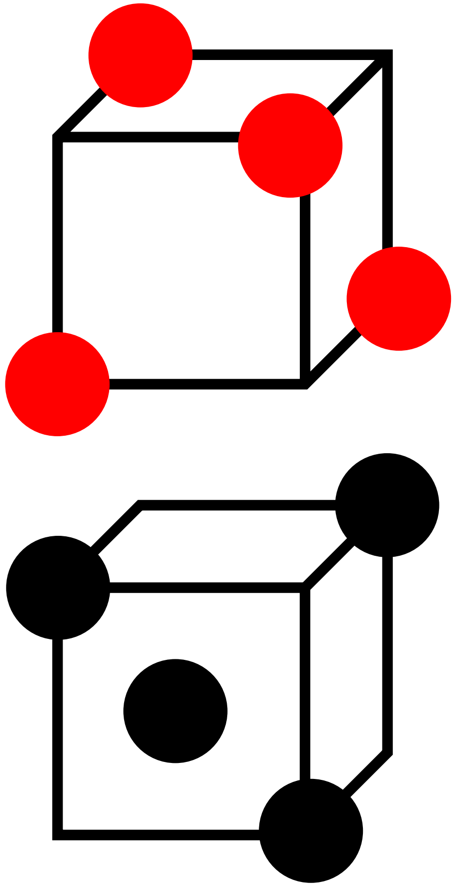
Output
0: H_2O 1: Na^+

Quanvolutional Layer:

# of wires: 4
(To reduce wires,
two quanvolutions
were done.)

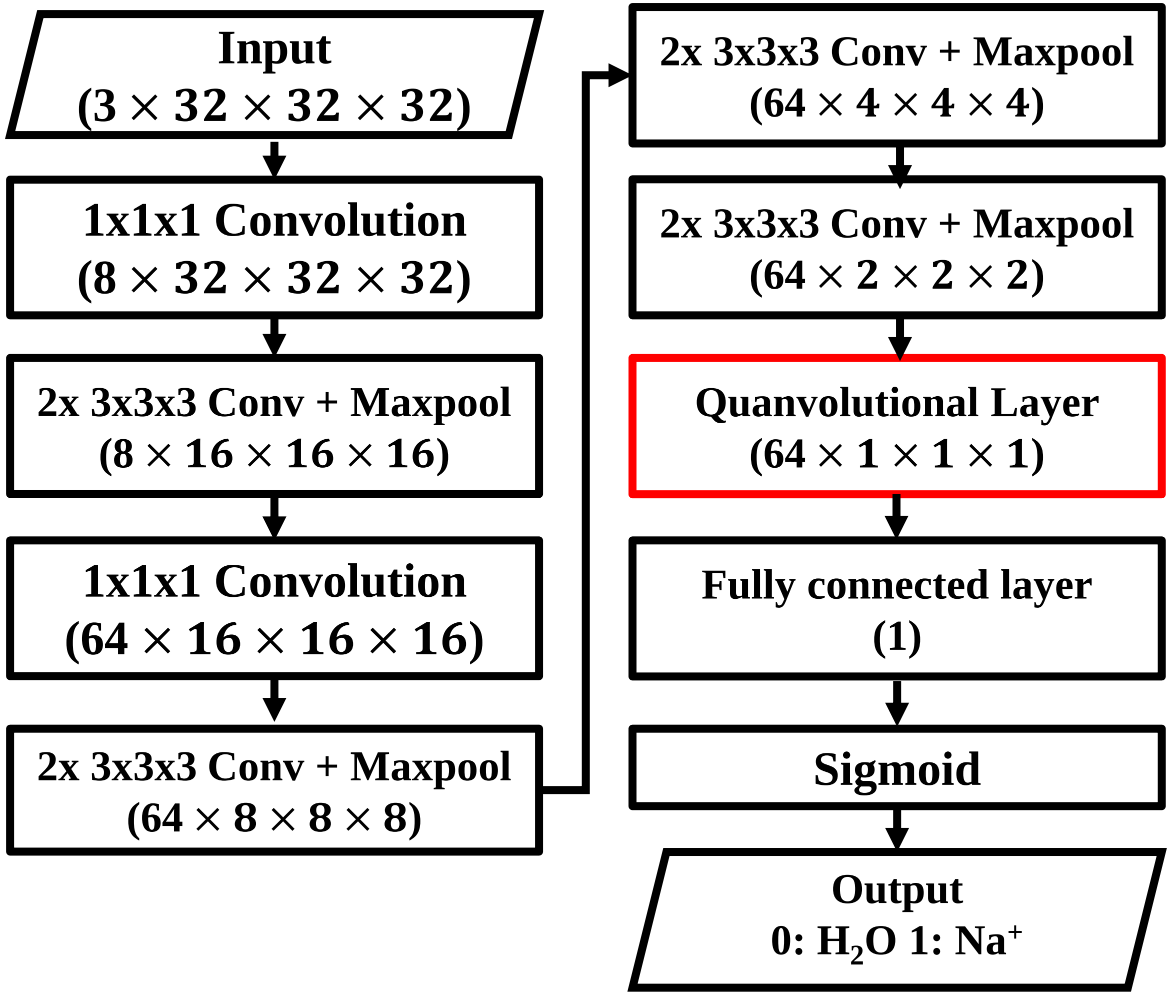
Quantum Circuit:
0. Starts with |0>
1. RX (π*O density)
2. RY (π*N density)
3. RZ (π*C density)
4. Randomlayer (
circuit layer:2.
rotations: 4)

Measurement: <Z>



Networks used in this project (3/3)

Trainable Quanvolutional Neural Network:



Input
(3 × 32 × 32 × 32)

1x1x1 Convolution
(8 × 32 × 32 × 32)

2x 3x3x3 Conv + Maxpool
(8 × 16 × 16 × 16)

1x1x1 Convolution
(64 × 16 × 16 × 16)

2x 3x3x3 Conv + Maxpool
(64 × 8 × 8 × 8)

2x 3x3x3 Conv + Maxpool
(64 × 4 × 4 × 4)

2x 3x3x3 Conv + Maxpool
(64 × 2 × 2 × 2)

Quanvolutional Layer
(64 × 1 × 1 × 1)

Fully connected layer
(1)

Sigmoid

Output
0: H_2O 1: Na^+

Quanvolutional Layer:

# of wires: 4

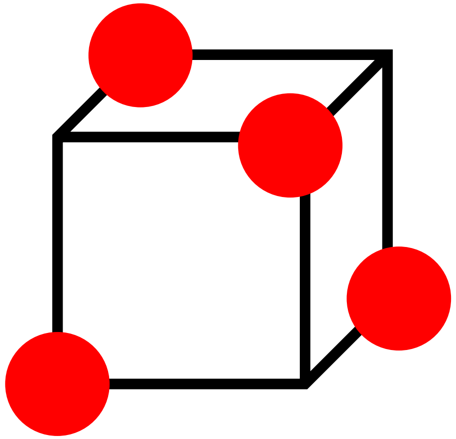
Quantum Circuit:
0. Starts with |0>

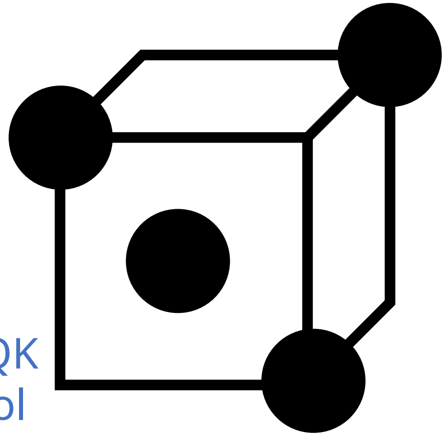
1. RY (π*input value)

2. Randomlayer (
circuit layer:2.
rotations: 4,
Trainable)

Measurement: <Z>

https://github.com/PlanQK/TrainableQuantumConvolution





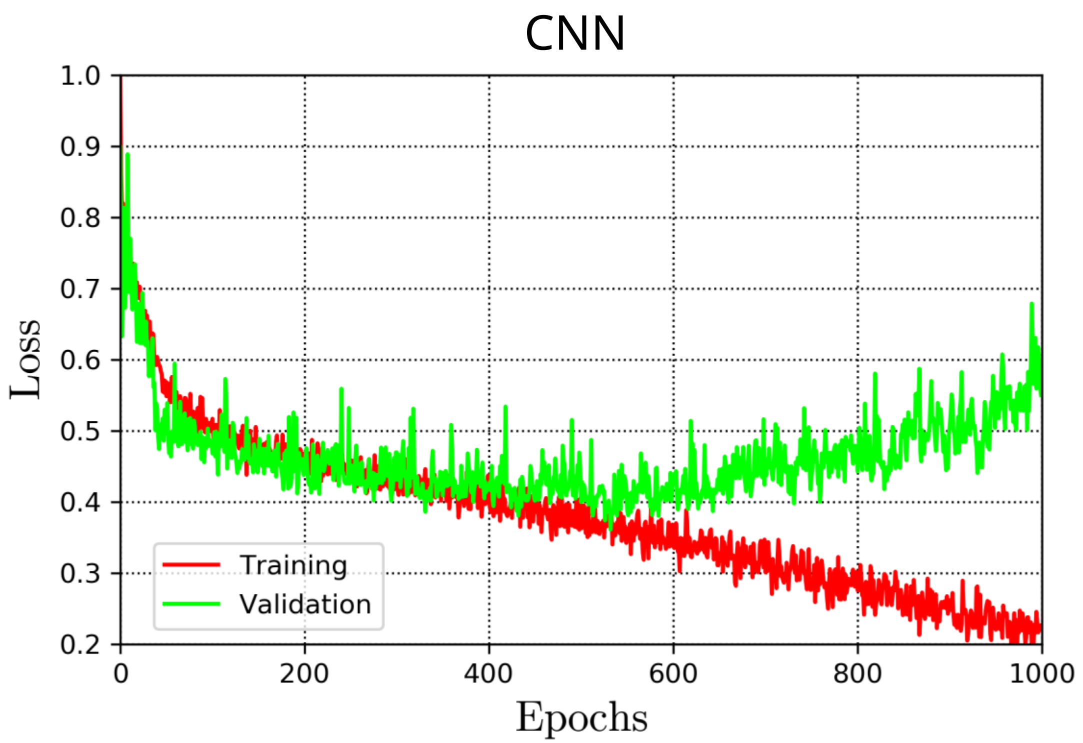
Loss functions / Optimizers

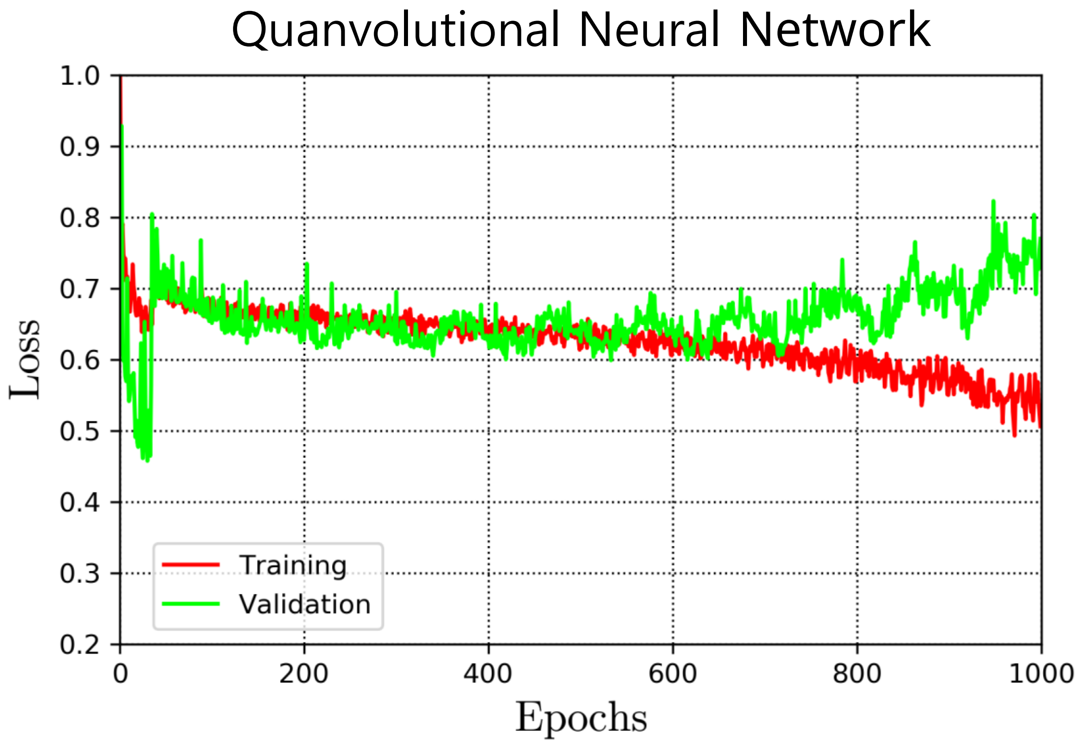
- Loss function: Cross Entropy loss (Batchsize = 4)

- Optimizer: Adam optimizer
  - CNN / QUANV : learning rate = 0.0001

  - Trainable QUANV : learning rate = 0.03

Results – Loss (1/2)

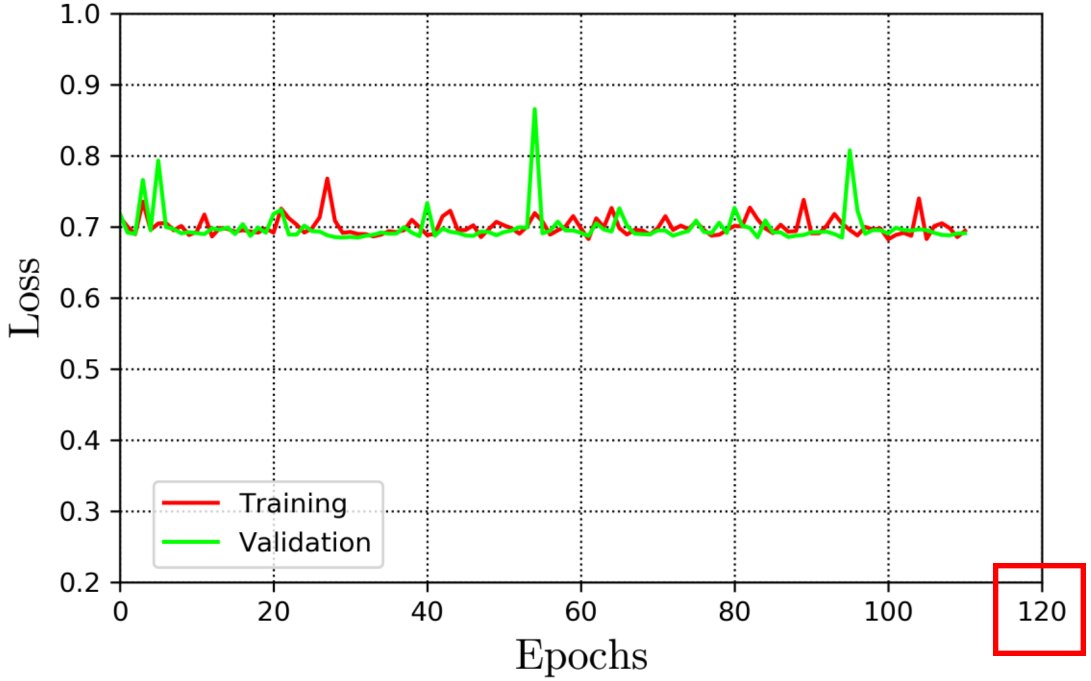




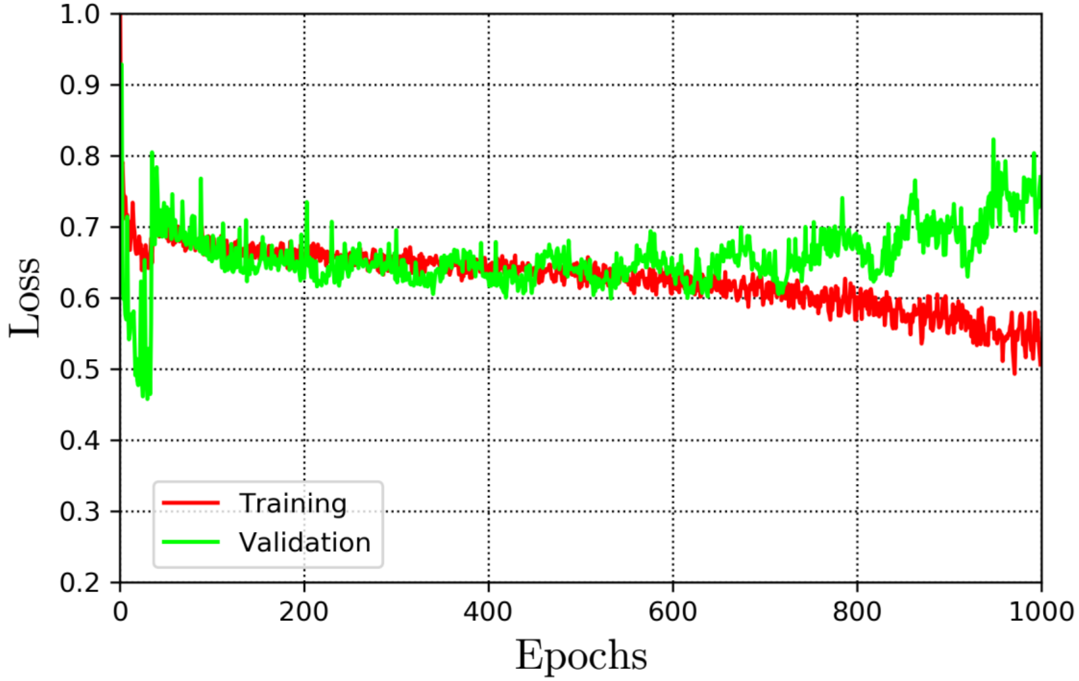
(maybe) Due to limited amount of rotations(=100) in training set,

Both CNN and QUANV shows overtraining after 400 epoch.

Results – Loss (2/2)



Trainable Quanvolutional Neural Network



Quanvolutional Neural Network

Trainable Quanvolutional Neural Network was not trained well,

even though learning rate was higher than other two methods.

(Maybe I need to learn more about optimizers in Quantum Machine Learning.)

Results – Validation set accuracy

	Overall accuracy	Water accuracy	Sodium ion accuracy
CNN - 400 epochs (Not overtrained yet)	**81.5%** (88 / 108)	**85.2%** (46 / 54)	**77.8%** (42 / 54)
CNN - 1000 epochs (Overtrained)	**77.8%** (84 / 108)	**87.0%** (47 / 54)	**68.5%** (37 / 54)
QUANV - 400 epochs (Not overtrained yet)	**66.7%** (72 / 108)	**68.5%** (37 / 54)	**64.8%** (35 / 54)
QUANV - 1000 epochs (Overtrained)	**61.1%** (66 / 108)	**64.8%** (35 / 54)	**57.4%** (31 / 54)
Trainable QUANV (111 epochs)	**55.6%** (60 / 108)	**27.8%** (15 / 54)	**83.3%** (45 / 54)

Conclusion

- Machine learning can specify H_2O / Na^+ !
    - From CNN's accuracy, which is about 80% !
- However, quanvolution based method showed worse accuracy.
    - 55.6% for trainable quanv, 61~67% for non-trainable quanv
    - Maybe different network design, loss function, optimization method would be necessary to show higher performance.
    - But, still better than random guessing!

What can be done further?

- Prepare better training / test set
  - Larger and high-resolution (low R-free, B-factor, etcetc) set
  - Use more atom types rather than using C, N, O only.
- Use better optimization method for quanvolution, or try to use QGCNN
  - Embedding will be done for nearby atoms, rather than nearby voxels

What can be done further? (cont.)

- Try to apply rotational equivariance concept in the network
  - Most of computation were consumed for learning rotation!
    - Building 100 rotated input structure for each target, training 1000 epochs…
  - SE(3)-CNN-like network would be helpful for this problem.
    - What could be QML-version of SE(3)-CNN?
- And the most important one:
  - Try to think about the method about free E estimation with quantum computer – This is the true problem!!

Thanks for reading this presentation!

Acknowledgements:

- Seoklab (https://seoklab.org)
- Pennylane (https://pennylane.ai)
- AWS Braket (https://console.aws.amazon.com/braket/)
- IBM Quantum (https://quantum-computing.ibm.com/)
- BEMANI SOUND TEAM