Quantum Convolutional Neural Networks (QCNN) for High-Energy Physics Analysis at the LHC

Google Summer of Code @ ML4SCI Machine Learning for Science

Eraraya Ricardo Muten (Edo)
Quantum Technology Lab
Institut Teknologi Bandung
Indonesia

Mentors: Prof. Sergei V. Gleyzer, Dr. Emanuele Usai, Raphael Koh

eraraya-ricardo.me

github.com/eraraya-ricardo/qcnn-hep

Background

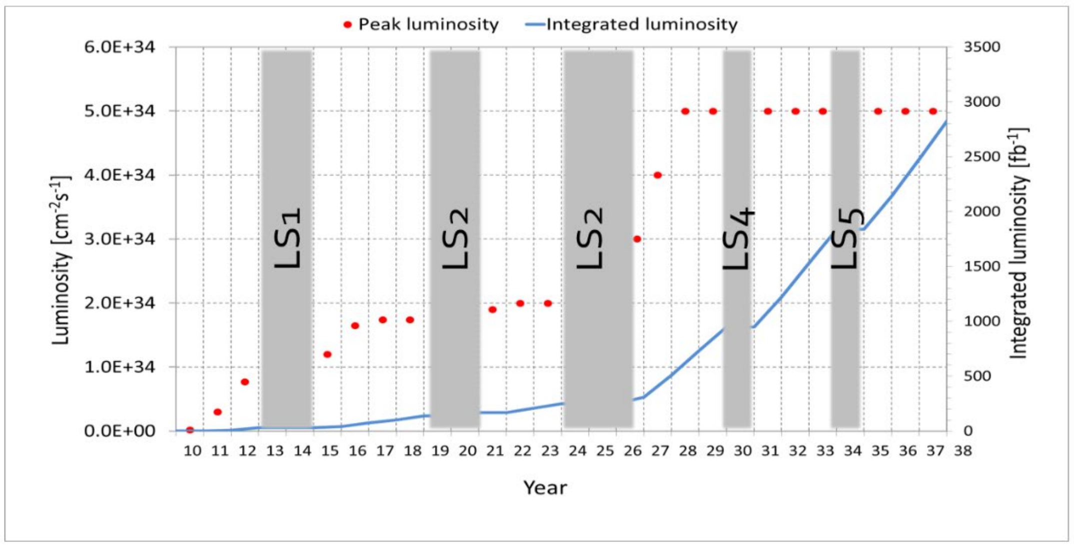
01

HL-LHC upgrades at CERN will **require enormous computing resources**[1]

02

Quantum computing has potential in improving performance of data processing and ML[2]

Can it improves HEP data analysis?



Projected LHC performance through 2038, **more luminosity = produce more data**[1]

[1] Burkhard Schmidt 2016 *J. Phys.: Conf. Ser.* **706** 022002.
[2] Biamonte J, et al. *Nature* 2017;**549**.

Dataset

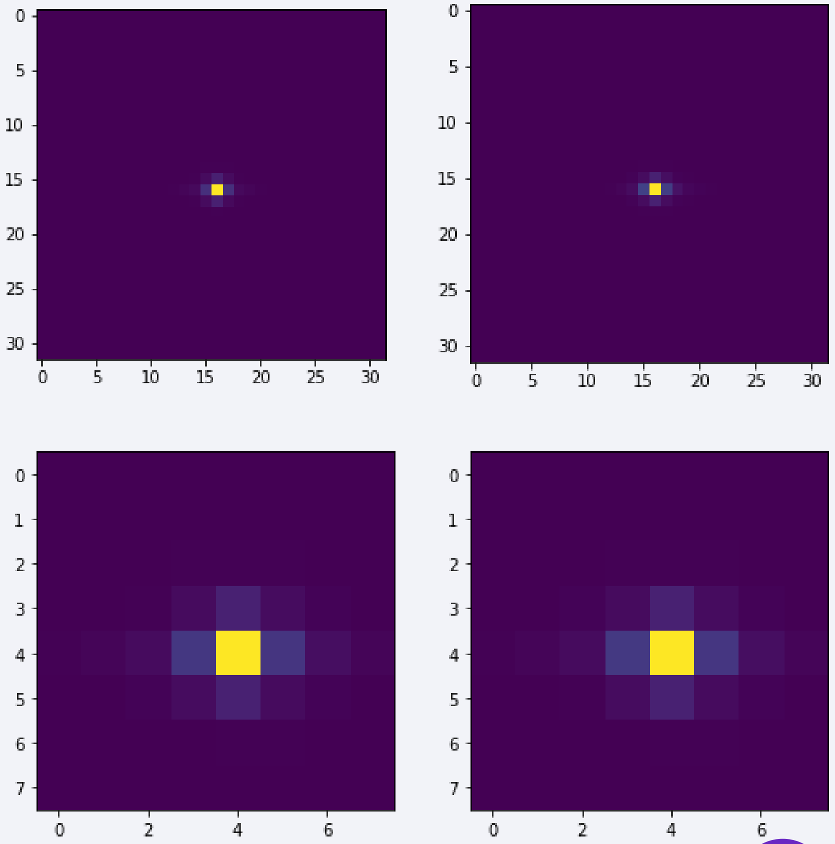
1. Photon-Electron ECAL Dataset

Images of electrons and photons captured by ECAL (Electromagnetic Calorimeter).

- A pixel = a detector cell
- Pixel's intensity = energy measured in that cell
- The dataset contains 32x32 images but cropped into 8x8
- The every pixel in the dataset is standard-scaled

$$x' = \frac{x - \mu}{\sigma}$$

Averages of image samples from the dataset.
Left: Photon, Right: Electron.
Top: Full 32x32, Bottom: After cropping 8x8.

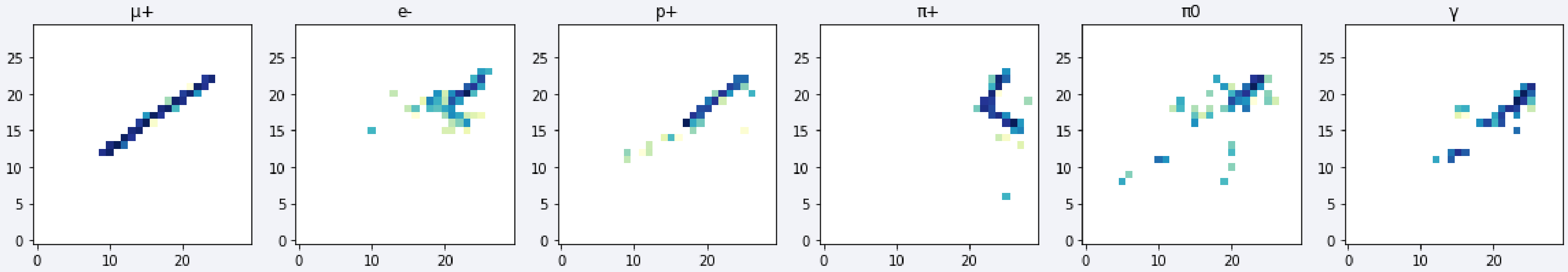


Dataset

2. LArTPC Dataset[3]

Images of simulated particle activities in a LArTPC (Liquid Argon Time Projection Chamber) detector.

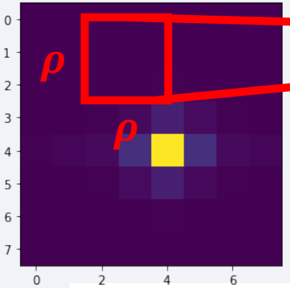
- Pixel's intensity = ionization energy loss along the particle trajectories
- The dataset contains 480x600 images, cropped to 30x30
- The every pixel in the dataset is log-scaled + MinMax-scaled



[3] Chen, S.Y., et al. 2020. **arXiv:2012.12177**

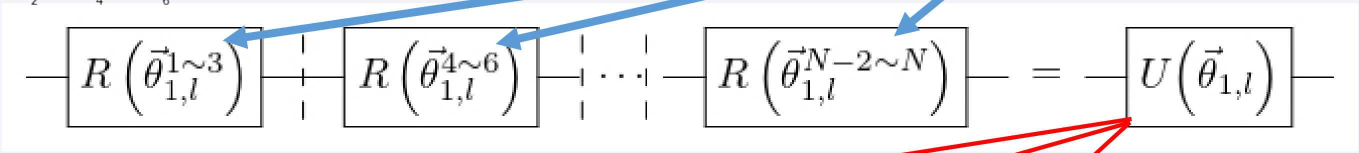
Quantum Convolution Layer with Data Re-uploading Circuit (QCNN-DRC)

- The variational quantum circuit used for the quantum convolution layers is the data re-uploading circuit [4]

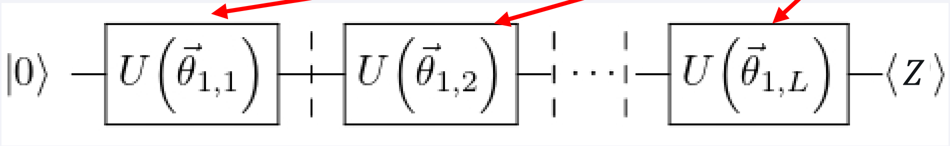


$\vec{x} = (x^1, x^2, \dots, x^\zeta)$, $\rho \times \rho = 3 \times 3$

$\theta^n_{q,l} = w^n_{q,l} x^n + b^n_{q,l} \rightarrow \vec{\theta}^{\,n\sim n+2}_{q,l} = (\theta^n_{q,l}, \theta^{n+1}_{q,l}, \theta^{n+2}_{q,l})$



$R(\phi, \theta, \omega) = RZ(\phi)RY(\theta)RX(\omega)$

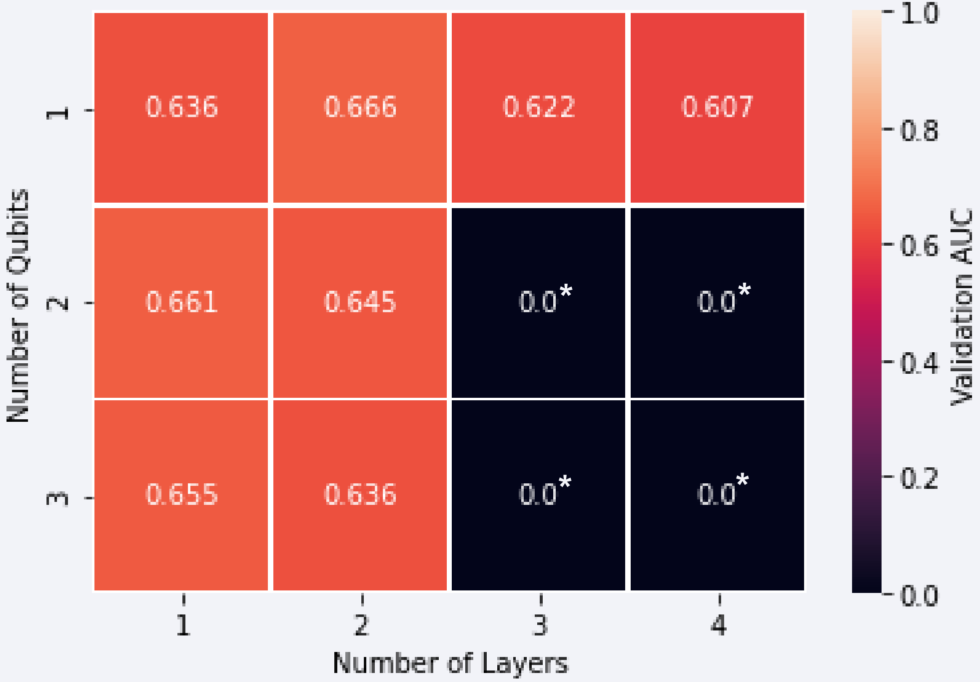


[4] Pérez-Salinas, A., et al. (2020). *Quantum*, 4, 226.

Results & Discussion (ECAL Dataset)

QCNN Validation AUC (8500 training samples, 1500 testing samples)

AUC = Area under the ROC (Receiver Operating Characteristic) Curve

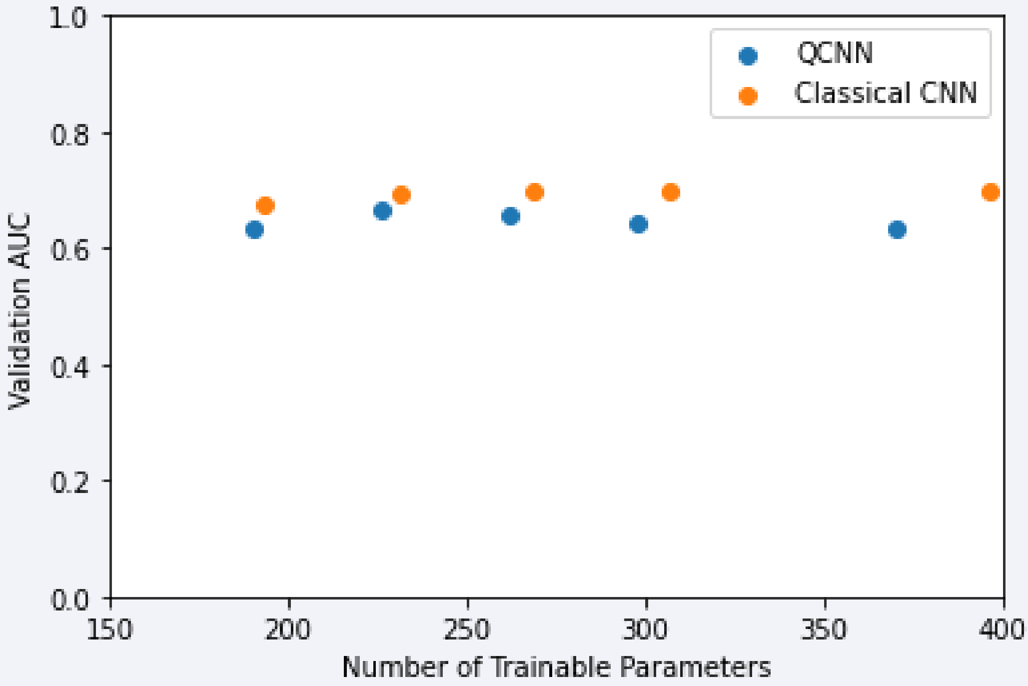


- Original paper shows more layers = higher accuracy as a classifier for 1-D dataset
- Not always the case if used as convolution filter for 2-D dataset

***0.0 (black) = not tested**

Results & Discussion (ECAL Dataset)

Validation AUC of QCNN vs Classical CNN (8500 training samples, 1500 testing samples)



With 423300 training samples & 74700 testing samples:

- QCNN (190 parameters): **0.730**
- Classical CNN (193 parameters): **0.738**

- With similar number of parameters, the QCNN is a little bit worse than Classical CNN

Both QCNN and Classical CNN:

- Increasing the number of training samples increases performance
- Only increasing the number of trainable parameters not necessarily increases performance (overfitting)

Results & Discussion (LArTPC Dataset)

Comparison of Best Models (160 training samples, 40 testing samples)

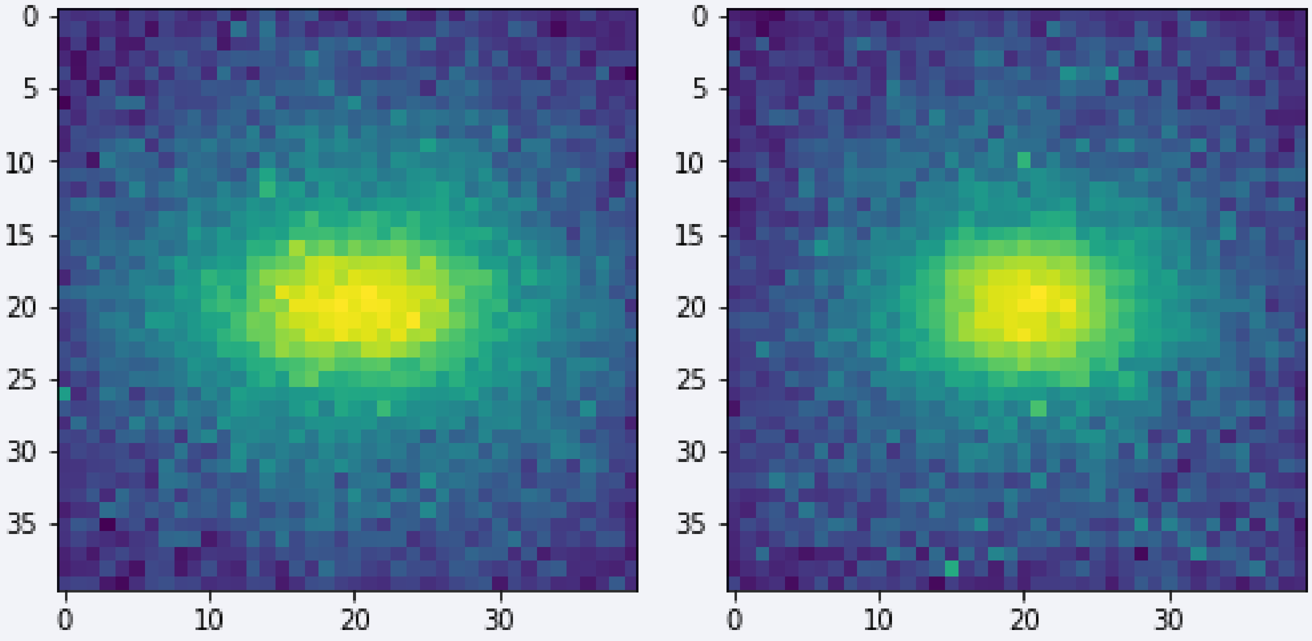
Classes	Model	Num. Qubits	Num. Trainable Params	Train Accuracy	Test Accuracy	Train AUC	Test AUC
e- vs μ+	QCNN-DRC*	2	220	1.0	0.950	1.0	0.996
	QCNN†	9	472	1.0	0.925	-	-
	CNN†	(classical)	498	0.9938	0.950	-	-
p+ vs μ+	QCNN-DRC*	1	130	1.0	0.950	1.0	0.980
	QCNN†	9	472	1.0	0.975	-	-
	CNN†	(classical)	498	0.9125	0.80	-	-
π+ vs μ+	QCNN-DRC*	2	220	1.0	0.950	1.0	0.977
	QCNN†	9	472	0.9688	0.975	-	-
	CNN†	(classical)	498	0.975	0.825	-	-

* = from this project
† = from reference [3], no AUC scores reported

- With fewer qubits, fewer parameters, and simpler circuit design, QCNN-DRC produced similar performance with QCNN from reference [3]
- Both QCNN models outperformed the classical CNN with fewer parameters.

[3] Chen, S.Y., et al. 2020. **arXiv:2012.12177**

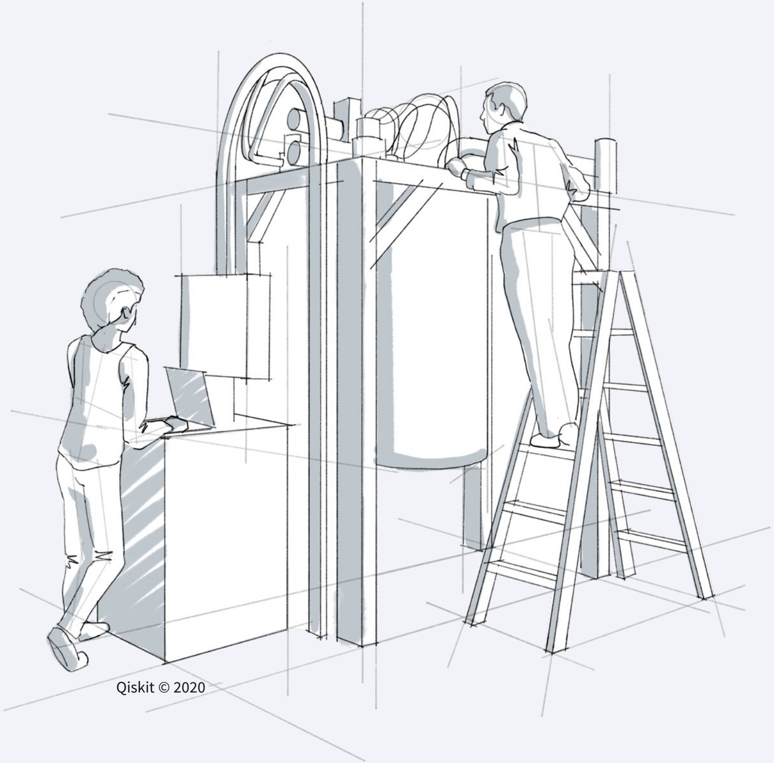
Future Works



Averages of image samples from the subdataset of 10k samples after cropping to 40x40.
Left: Gluon, Right: Quark.

Preliminary training with 1k samples showed promising results with Train AUC of 0.992.
Need to train with much more samples since the model was overfitted to the training set.

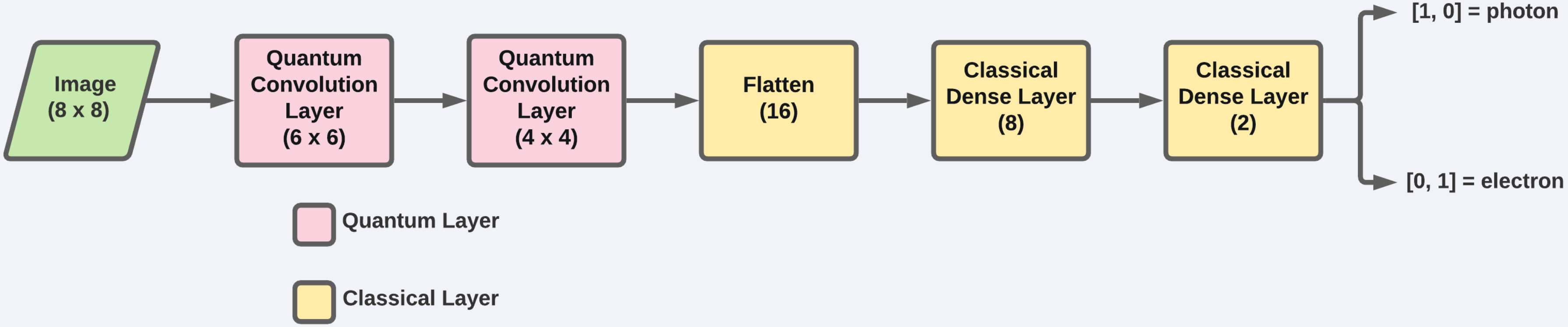
Thank You!
Any Questions/Comments?



APPENDIX

Algorithm

Overall Architecture of QCNN



Model creation & training: TensorFlow Quantum [6]
Quantum Simulator: Google's Cirq [7] noiseless analytic simulator

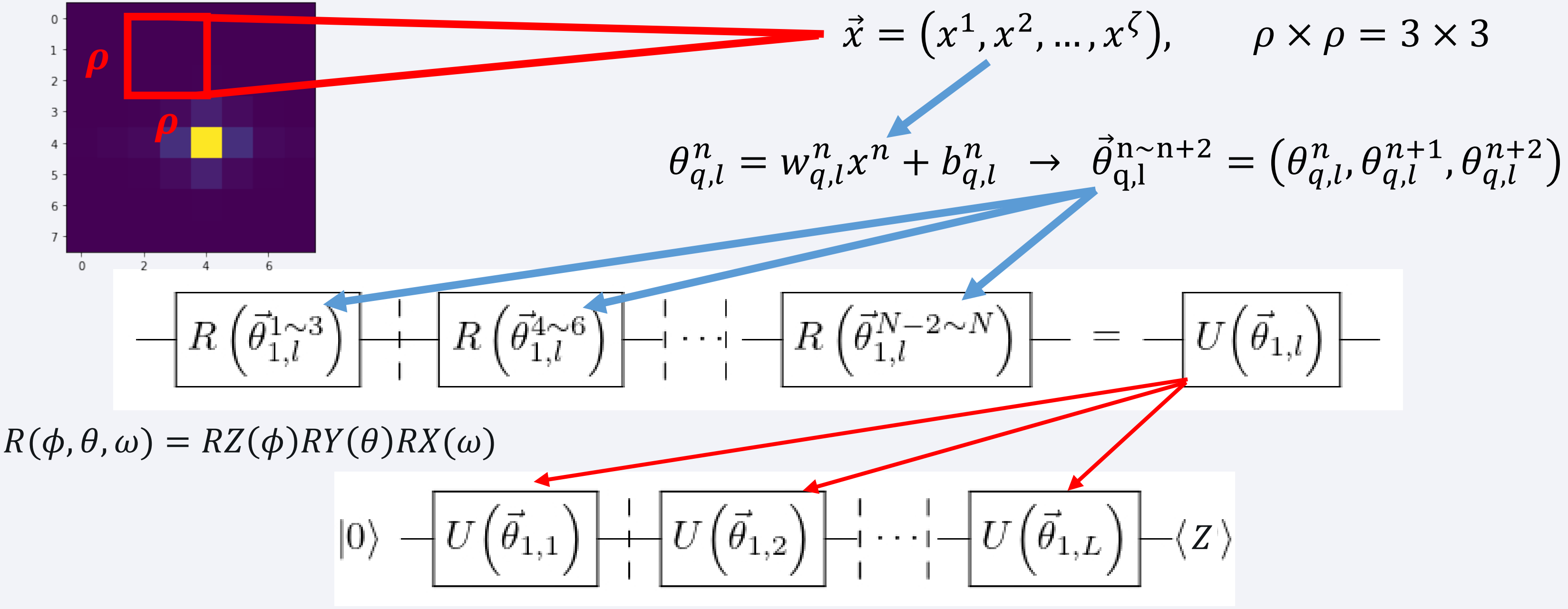
Quantum Convolution Layer:

- Transform input features to another feature map via a convolution-like operation
- The transformation is done by a trainable variational ansatz instead of a classical filter

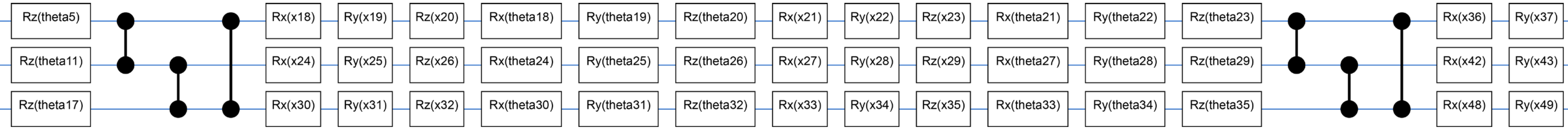
[6] Broughton, M., et al. 2020. **arXiv:2003.02989**
[7] Cirq Developers. **doi: 10.5281/zenodo.4586899**.

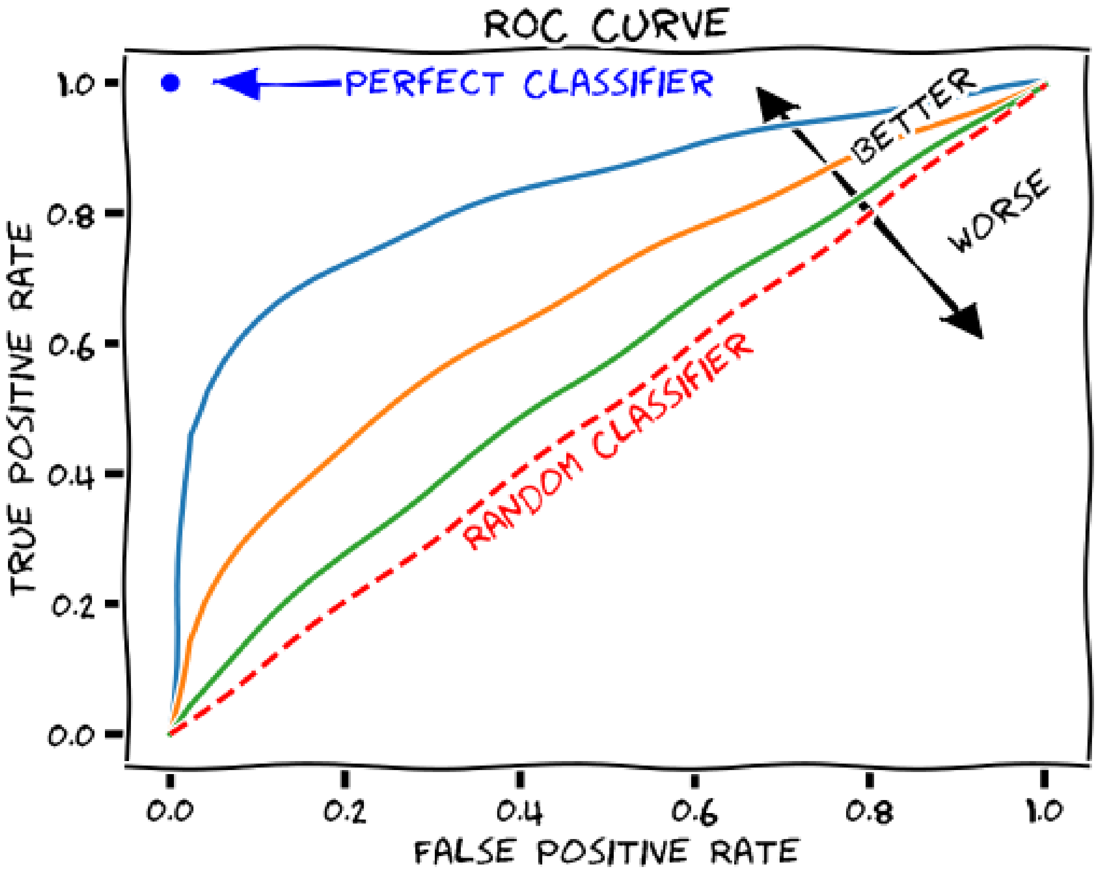
Quantum Convolution Layer with Data Re-uploading Circuit (QCNN-DRC)

- The variational quantum circuit used for the quantum convolution layers is the data re-uploading circuit [4]
- The number of layers and qubits can be increased (ring of CZ gate will be used if 2 or more qubits are used)
- The Z expectation of the last qubit is measured



[4] Pérez-Salinas, A., et al. (2020). *Quantum*, 4, 226.


Rz(theta5)
Rz(theta11)
Rz(theta17)
Rx(x18)
Ry(x19)
Rz(x20)
Rx(theta18)
Ry(theta19)
Rz(theta20)
Rx(x21)
Ry(x22)
Rz(x23)
Rx(theta21)
Ry(theta22)
Rz(theta23)
Rx(x36)
Ry(x37)
Rx(x24)
Ry(x25)
Rz(x26)
Rx(theta24)
Ry(theta25)
Rz(theta26)
Rx(x27)
Ry(x28)
Rz(x29)
Rx(theta27)
Ry(theta28)
Rz(theta29)
Rx(x42)
Ry(x43)
Rx(x30)
Ry(x31)
Rz(x32)
Rx(theta30)
Ry(theta31)
Rz(theta32)
Rx(x33)
Ry(x34)
Rz(x35)
Rx(theta33)
Ry(theta34)
Rz(theta35)
Rx(x48)
Ry(x49)


ROC CURVE
PERFECT CLASSIFIER
BETTER
WORSE
RANDOM CLASSIFIER
TRUE POSITIVE RATE
FALSE POSITIVE RATE
0.0
0.2
0.4
0.6
0.8
1.0

QCNN Settings

- 10k samples with 15% for test samples
- 200 epochs, 128 batch size
- *varying* qubits, *varying* layers
- filter size = [3, 3], stride = [1, 1]
- followed by classical head [8, 2] with activation [relu, softmax]
- classical preprocessing = crop to 8x8 + standard scaling
- optimizer: Adam, lr = 0.001 with decay, $\beta_1 = 0.9$, $\beta_2 = 0.999$, ε = 1e-07
- cross-entropy loss

- Simulator: noiseless
- Gradient calculation: analytic

Classical CNN Settings

- 10k samples with 15% for test samples
- 200 epochs, 128 batch size
- filter size = [3, 3], stride = [1, 1]
- conv activation = [relu, relu]
- use_bias = [True, True]
- followed by classical head [8, 2] with activation [relu, softmax]
- classical preprocessing = crop to 8x8 + standard scaling
- optimizer: Adam, lr = 0.001 with decay, $\beta_1 = 0.9$, $\beta_2 = 0.999$, ϵ = 1e-07
- cross-entropy loss

LR Decay

```
lr = 1e-3
    if epoch > 180:
        lr *= 0.5e-3
    elif epoch > 160:
        lr *= 1e-3
    elif epoch > 120:
        lr *= 1e-2
    elif epoch > 80:
        lr *= 1e-1
```