# Quantum machine learning using Hybrid quantum neural networks.

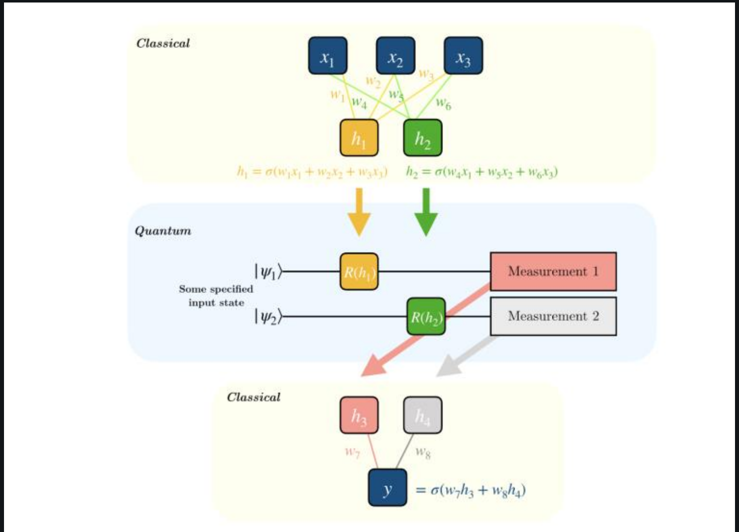
Juhyeok Lee, Jiwan Song, Younghun Ryu, Mingi Kim

Image classification by Quantum machine learning **#20**

Qiskit

# What we did

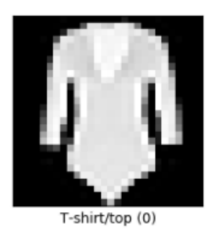
## Concept of hybrid QNN

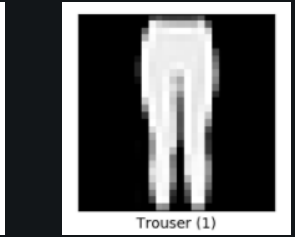


https://qiskit.org/textbook/ch-machine-learning/machine-learning-qiskit-pytorch.html

## Problem definition

1. Fashion_MNIST T-shirts & Trouser classification


T-shirt/top (0)


Trouser (1)

2. Quantum calculator (add function only)

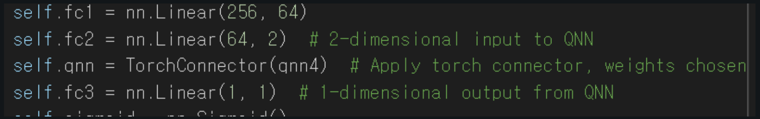
$$a, b \in \{x:\ x \in R, x \in [0,1]\}$$

Calculate $a + b$

# Results

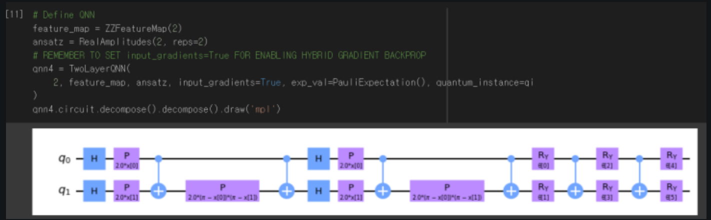
## Coding description

### TorchConnector from Qiskit



```
self.fc1 = nn.Linear(256, 64)
self.fc2 = nn.Linear(64, 2)  # 2-dimensional input to QNN
self.qnn = TorchConnector(qnn4)  # Apply torch connector, weights chosen
self.fc3 = nn.Linear(1, 1)  # 1-dimensional output from QNN
self.sigmoid = nn.Sigmoid()
```

### Quantum Neural Network (QNN)



```
[11] # Define QNN
     feature_map = ZZFeatureMap(2)
     ansatz = RealAmplitudes(2, reps=2)
     # REMEMBER TO SET input_gradients=True FOR ENABLING HYBRID GRADIENT BACKPROP
     qnn4 = TwoLayerQNN(
         2, feature_map, ansatz, input_gradients=True, exp_val=PauliExpectation(), quantum_instance=qi
     )
     qnn4.circuit.decompose().decompose().draw('mpl')
```

ZZFeatureMap (Encoding) RealAmplitude (Ansatz)

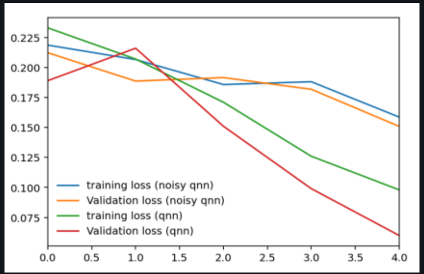
## Learning result

### 1. Fashion_MNIST classifier

```
ground truth: 0
prediction  : 0
```

```
ground truth: 1
prediction  : 1
```

# Aer qasm_simulator

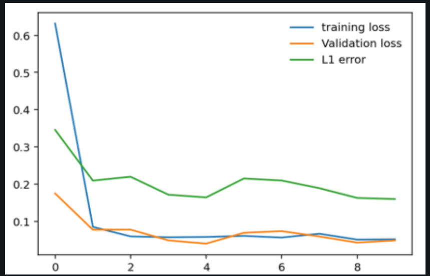


Epoch number

### 2. Quantum Calculator

input: tensor([0.8586,0.4095])
ground_truth: 1.2680
prediction: 1.1983
L1 error: 0.0697

# Aer qasm_simulator



Epoch number

# Problems

## 1. No quantum advantage (in fundamental way)

→ No use of entanglement / Use classical optimization

→ Imperfect quantum gate operations

## 2. Queue time issue

→ hundreds of learning cycle / 7-8min per one circuit job