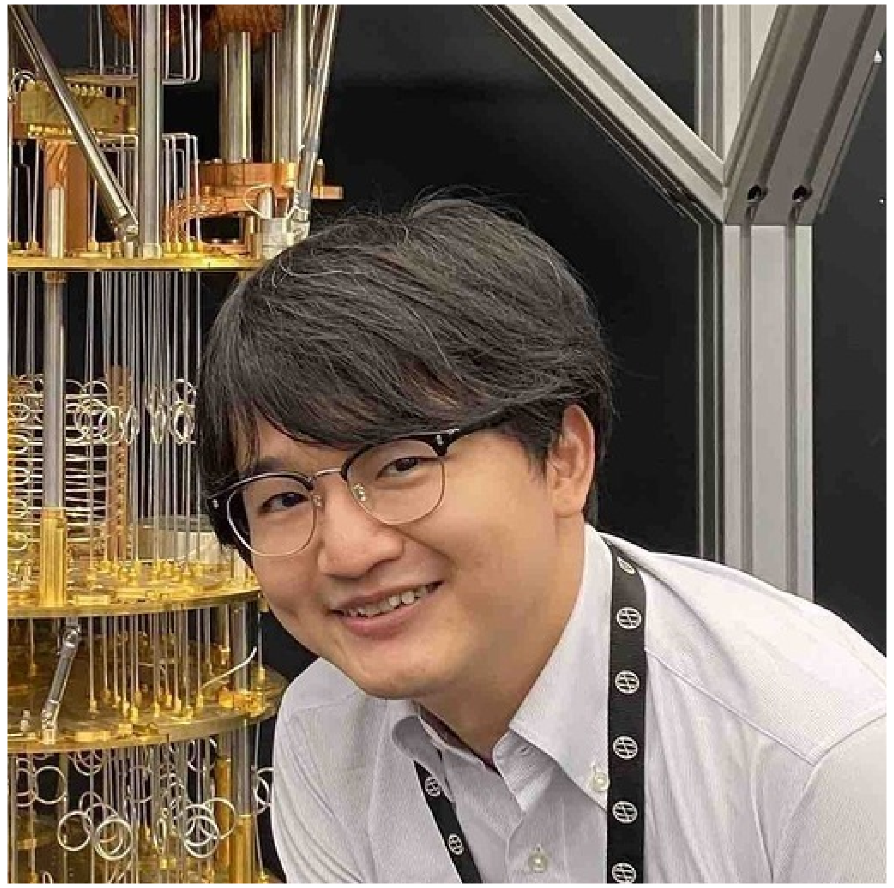
Qiskit

Qiskit Advocate Mentorship Program, Spring'23

***Project** : Implement Date-reuploading classifier in Qiskit Machine Learning #3*

Qiskit



Mentor: Atsushi Matsuo
Researcher at IBM , Japan

Eraraya Ricardo Muten

MS, Quantum Science & Technology, Technical University of Munich

Shivani Rajput

Research Associate, University of Delhi , India

Project Description

Data re-uploading is a recently proposed idea of quantum neural network, which uses a quantum circuit with a series of data re-uploading and processing layers. Unlike the conventional quantum circuit of quantum neural network, it has multiple layers of re-uploading input data. In this project, we will implement the data re-uploading quantum neural network in Qiskit Machine Learning. The goal of this project is to write code and create a pull request.

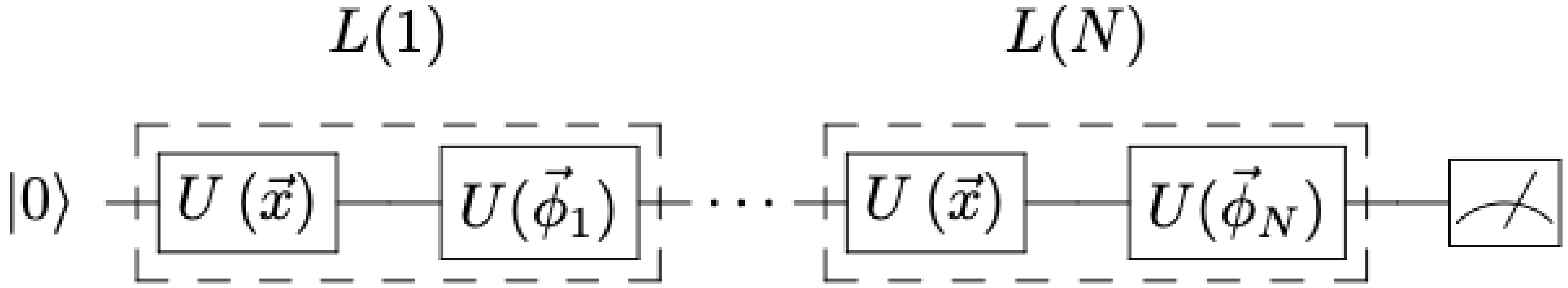


Figure from Pérez-Salinas et al. (2019)

Original Paper: https://quantum-journal.org/papers/q-2020-02-06-226/pdf/

Progress

Qiskit

Step 1 - Prepared Data-reuploading class

a) For Single Qubits: Only Rotational Gate layer (L)

$U(\varphi, x) = L(N)\, L(N-1) \ldots L(2)L(1)$

$$L(i) = U\left(\vec{\theta}_i + \vec{w}_i \circ \vec{x}\right)$$

$$L(i) = U\left(\vec{\theta}_i^{(k)} + \vec{w}_i^{(k)} \circ \vec{x}^{(k)}\right) \cdots U\left(\vec{\theta}_i^{(1)} + \vec{w}_i^{(1)} \circ \vec{x}^{(1)}\right),$$

$U(\vec{\theta}) = Rx(\theta x)Ry(\theta y)Rz(\theta z)$ $\vec{\theta} = (\theta x, \theta y, \theta z)$ $\vec{w} = (w_x w_y w_z)$ $\vec{x}$= input data (N) = no of layers

b) Multi qubits without entanglement : Stacking the same ansatz(U)for multiple qubits.

c) Multi qubits with entanglement (CZ gates). Alternating between (L) Rotational Layer and and (E)entanglement layer.

$U(\varphi, x) = L(N)\, E\, L(N-1)\, E \ldots E\, L(2)EL(1)$

[Note : followed Linear entanglement]

Original Paper: https://quantum-journal.org/papers/q-2020-02-06-226/pdf/

Progress

Qiskit

Step2: Integration with QNN implementation of Qiskit Machine Learning (EstimatorQNN and SamplerQNN) and used PyTorch with TorchConnector feature of Qiskit.

Step3: Optimization

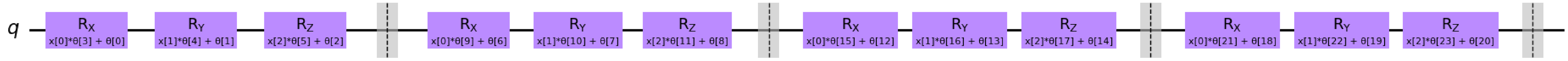
The optimization process is initiated using the optimizer and loss function.

Step4: After optimization, the trained model (**model1**) is used to make predictions on the input data (**X_**), and the accuracy of the predictions is calculated by comparing them to the ground truth labels (**y_**)

Original Paper: https://quantum-journal.org/papers/q-2020-02-06-226/pdf/

Progress

DRC = DataReuploading(num_qubits=1, num_features=2, num_layers=4)

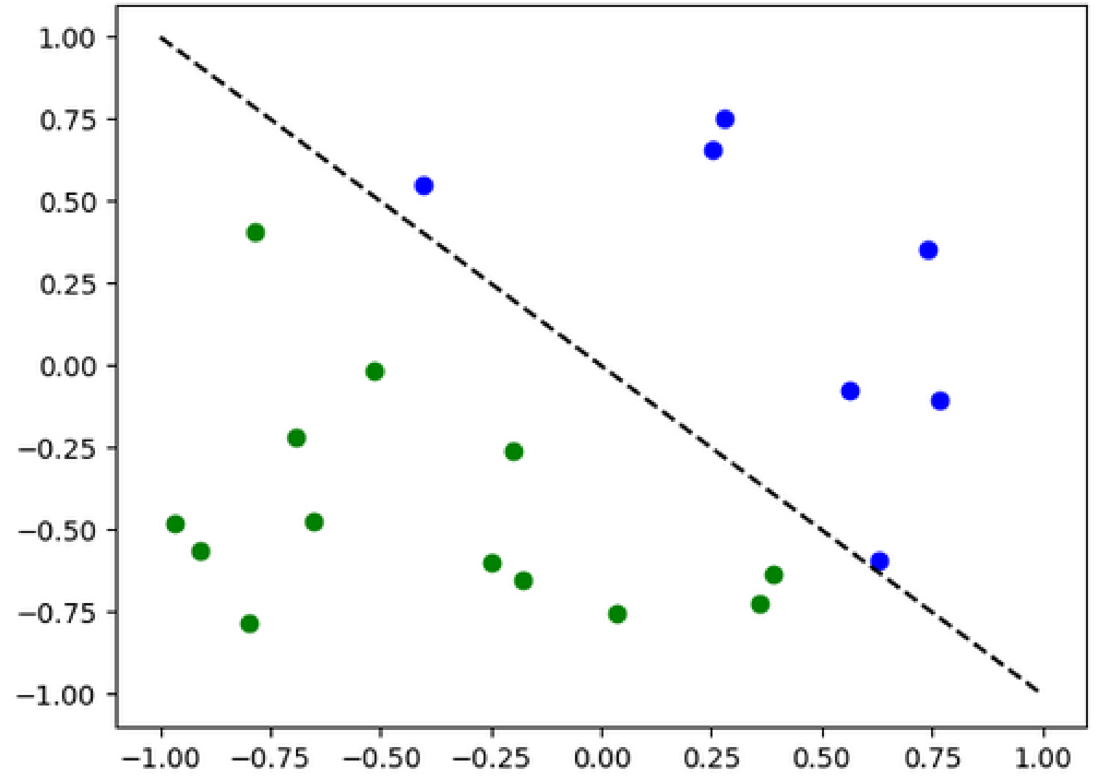



```python
# Evaluate model and compute accuracy
y_predict = []
for x, y_target in zip(X, y):
    output = model1(Tensor(x))
    y_predict += [np.sign(output.detach().numpy())[0]]

print("Accuracy:", sum(y_predict == y) / len(y))

# Plot results
# red == wrongly classified
for x, y_target, y_p in zip(X, y, y_predict):
    if y_target == 1:
        plt.plot(x[0], x[1], "bo")
    else:
        plt.plot(x[0], x[1], "go")
    if y_target != y_p:
        plt.scatter(x[0], x[1], s=200, facecolors="none", edgecolors="r", linewidths=2)
plt.plot([-1, 1], [1, -1], "--", color="black")
plt.show()
```

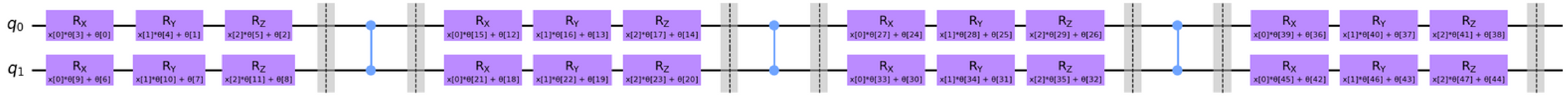
```
Accuracy: 1.0
```



Result: accuracy **100%**

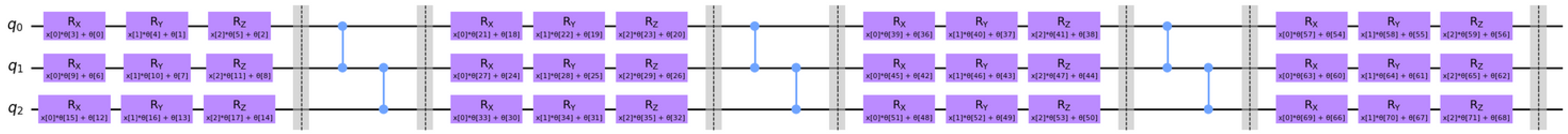
Qiskit

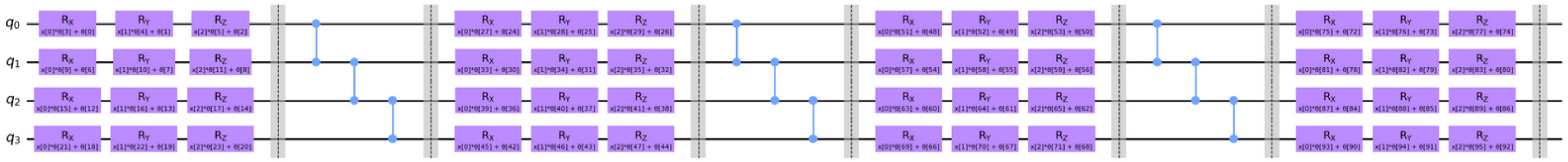
Example Circuits



2 qubits

3 qubits





4 qubits

Work to be done..

Qiskit

- We will implement the prepared class inside the Qiskit Machine Learning.

- We will write some tutorials with different datasets and configurations.

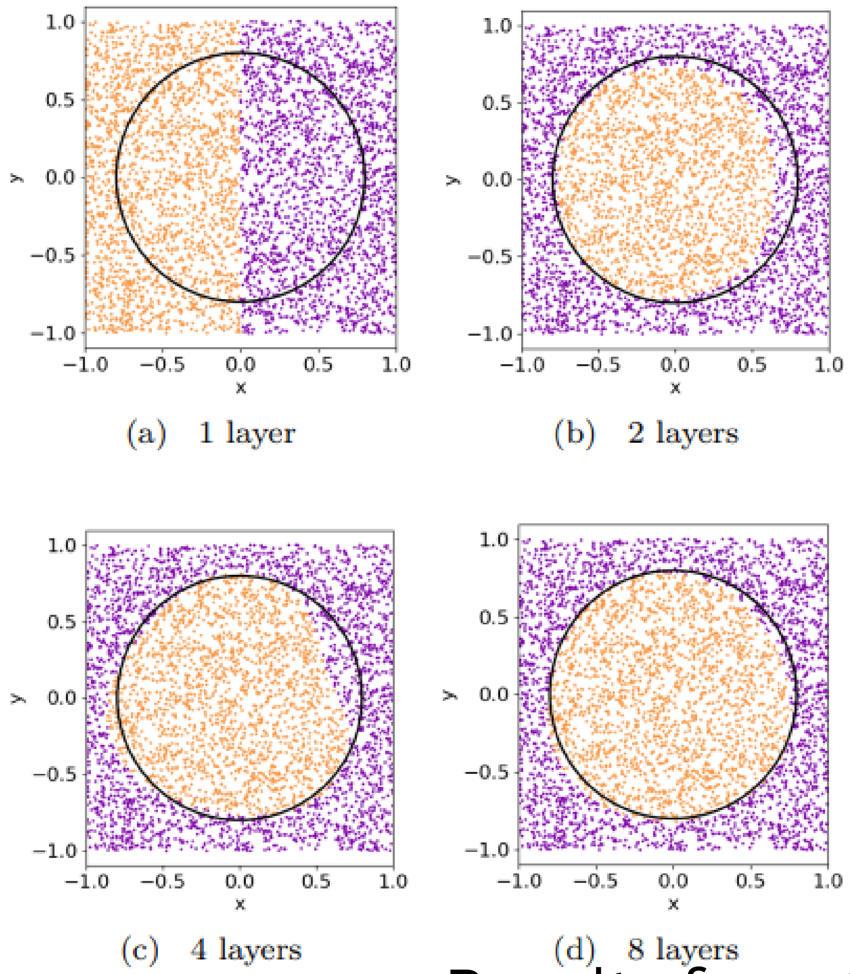
For ex: Training the ansatz with single qubits and multi qubits with different number of features and layers

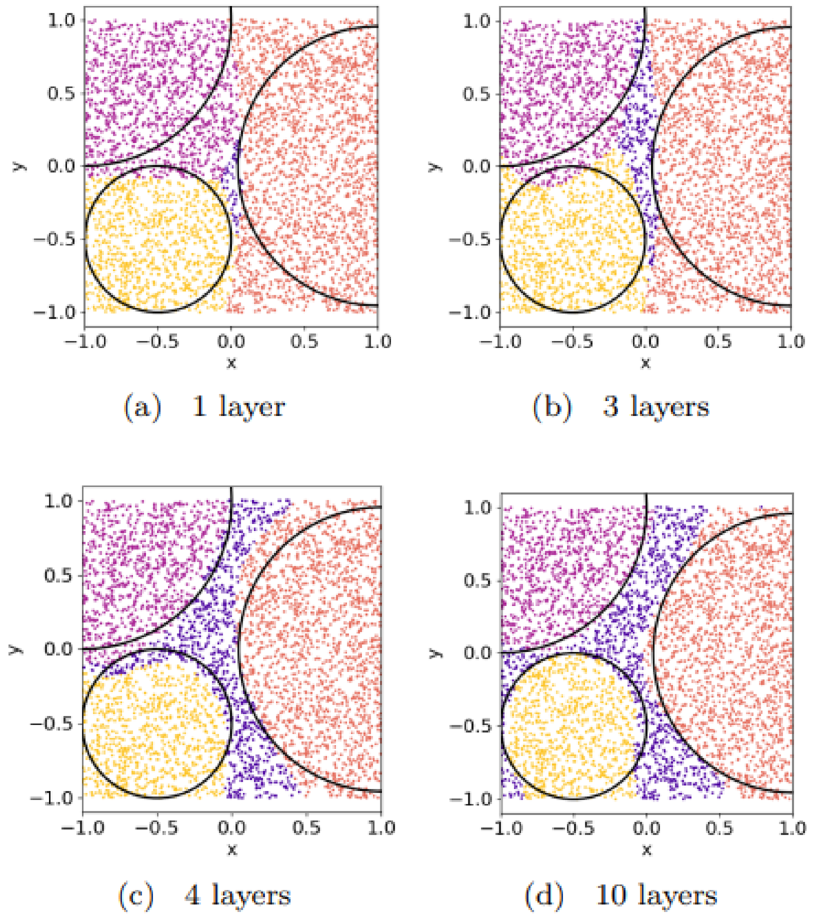
Original Paper: https://quantum-journal.org/papers/q-2020-02-06-226/pdf/

Work to be done..

Qiskit

- Circle Classification





Results from the original paper : https://quantum-journal.org/papers/q-2020-02-06-226/pdf/

Results of the circle classification obtained with a single-qubit classifier with different number of layers using the L-BFGS-B minimizer and the weighted fidelity cost function.

Results of the 3-circles problem classification obtained with a single-qubit classifier with different number of layers using the L-BFGS-B minimizer and the weighted fidelity cost function.

To do- WE WILL TRY TO DO THE SAME CLASSIFICATION USING OUR PREPARED CLASS AND WE WILL BE USING REGULAR FIDELITY

THANK YOU!