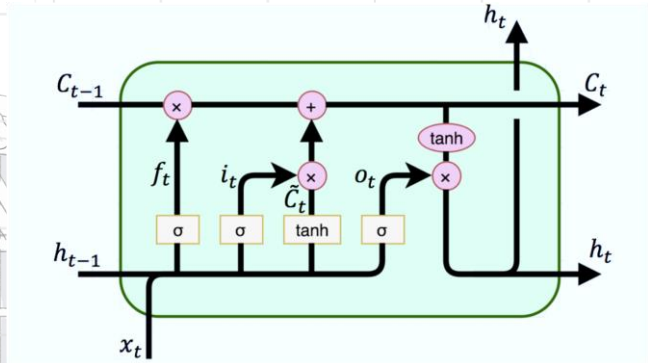
#38 Hybrid algorithm for predicting stock prices



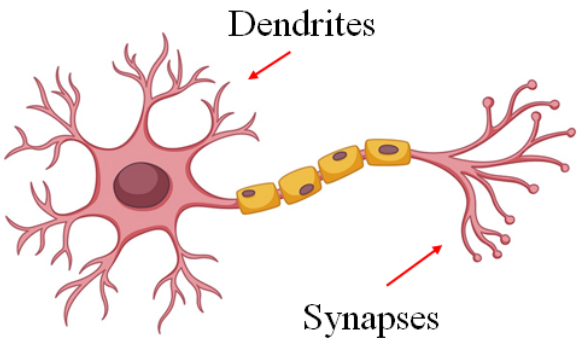
Qiskit Advocate Mentorship Program – Spring 2022

Mentors: Alberto Maldonado

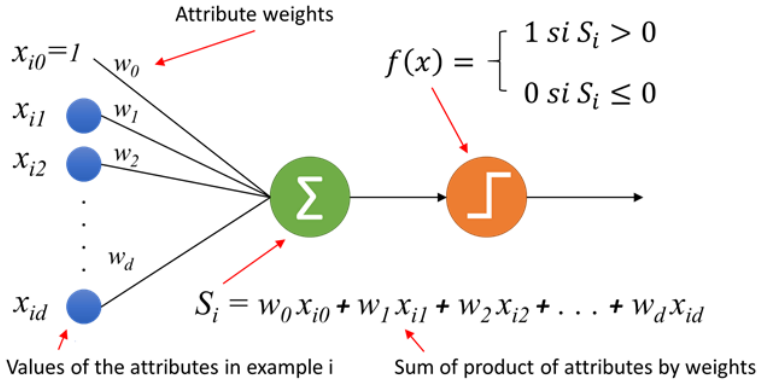
Mentees: Siddhartha Morales

Qiskit

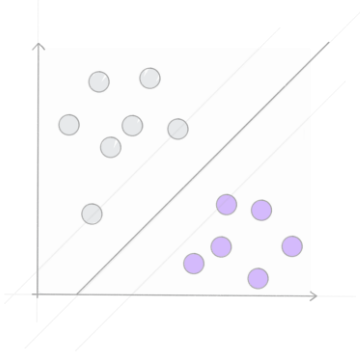
Classical Machine Learning



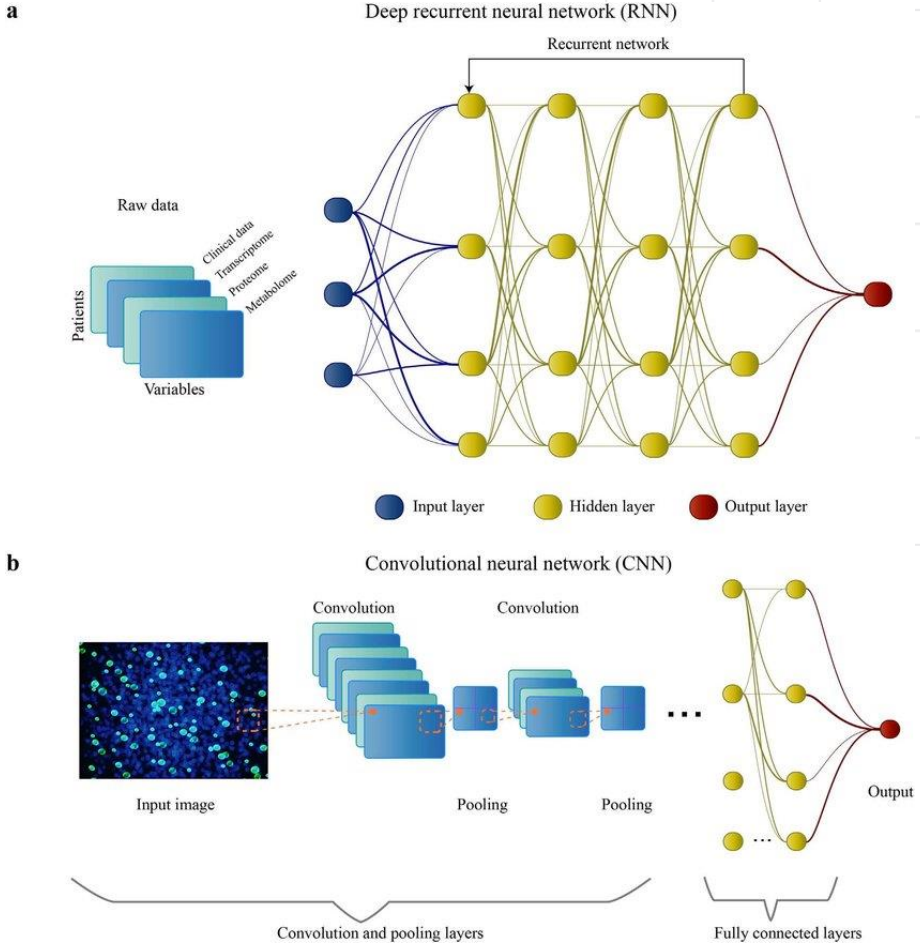
NEURON



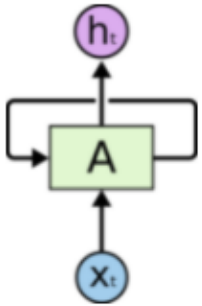
PERCEPTRON



Classical Machine learning

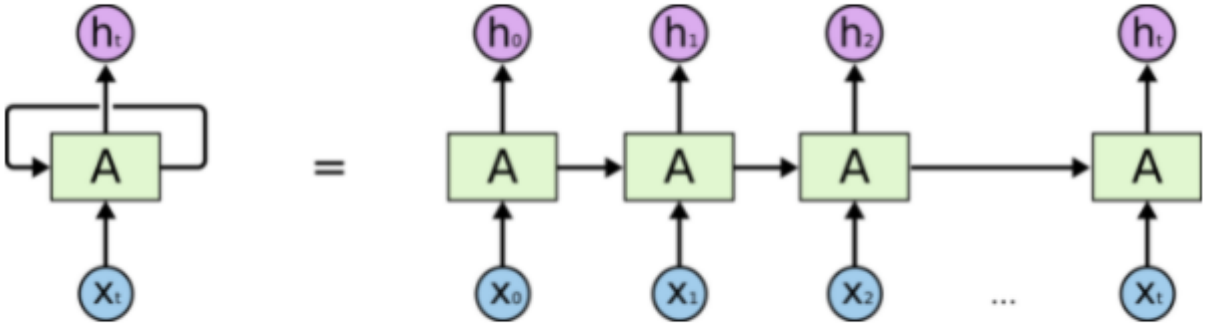


Recurrent Neural Networks



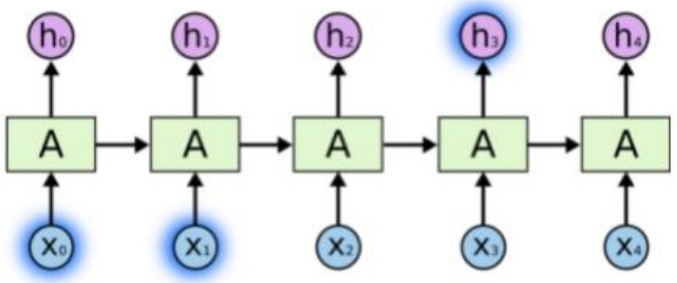
I live in Brazil… I speak fluent *Portuguese*

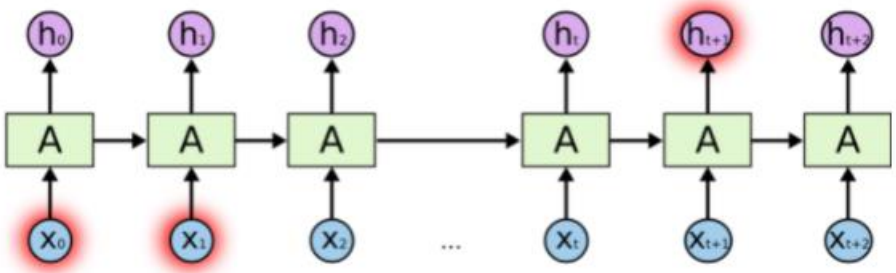
Recurrent Neural Networks have loops.



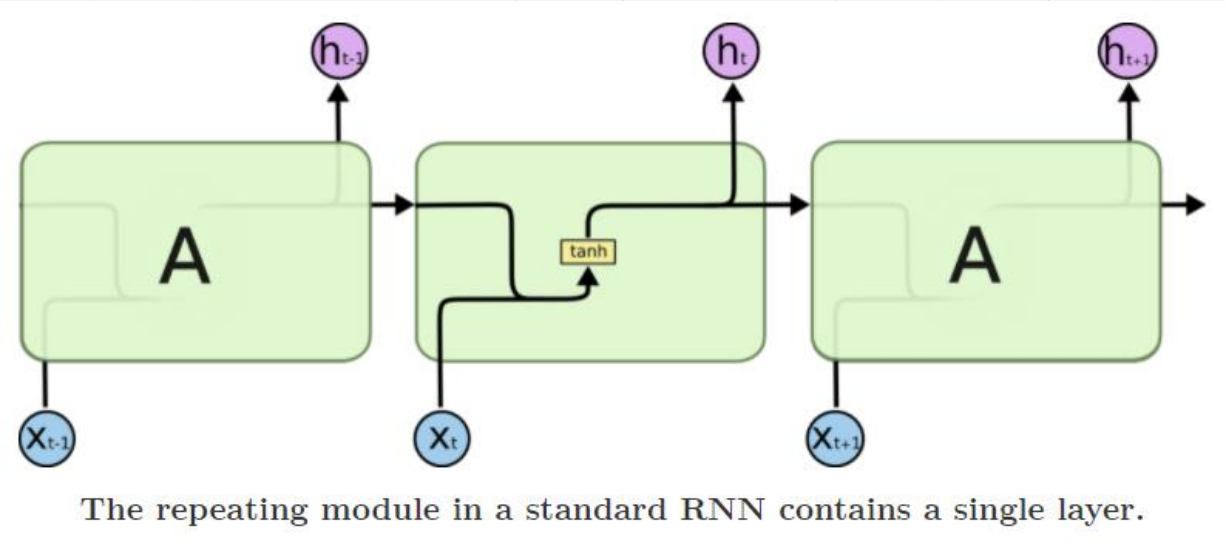
An unrolled recurrent neural network.

Recurrent Neural Networks





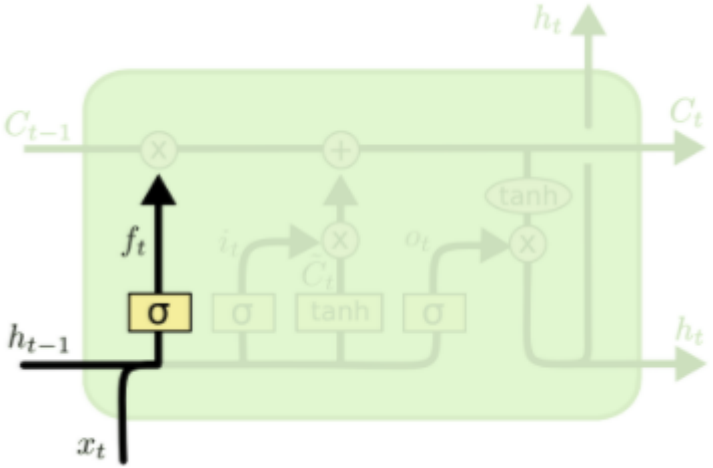
Long short-term Memory



The repeating module in a standard RNN contains a single layer.

Hochreiter & Schmidhuber (1997)
http://www.bioinf.jku.at/publications/older/2604.pdf

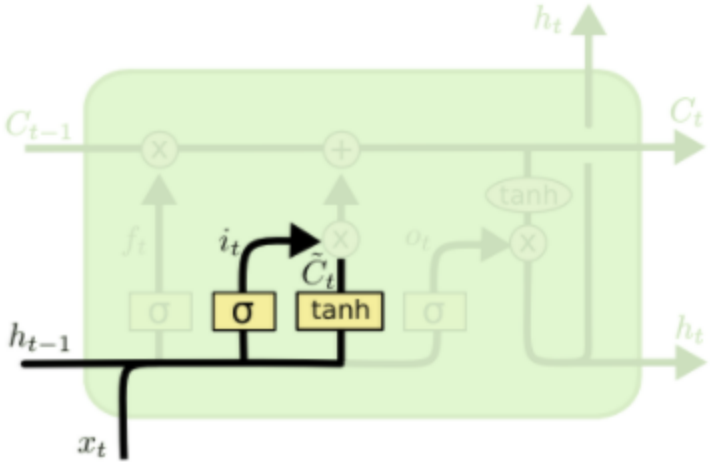
Long short-term Memory: forget layer



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Hochreiter & Schmidhuber (1997)
http://www.bioinf.jku.at/publications/older/2604.pdf

Long short-term Memory: update cell state

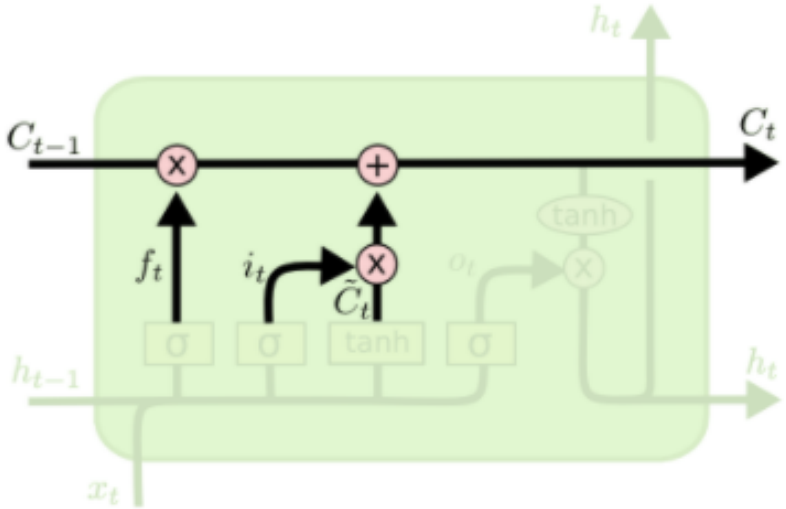


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Hochreiter & Schmidhuber (1997)
http://www.bioinf.jku.at/publications/older/2604.pdf

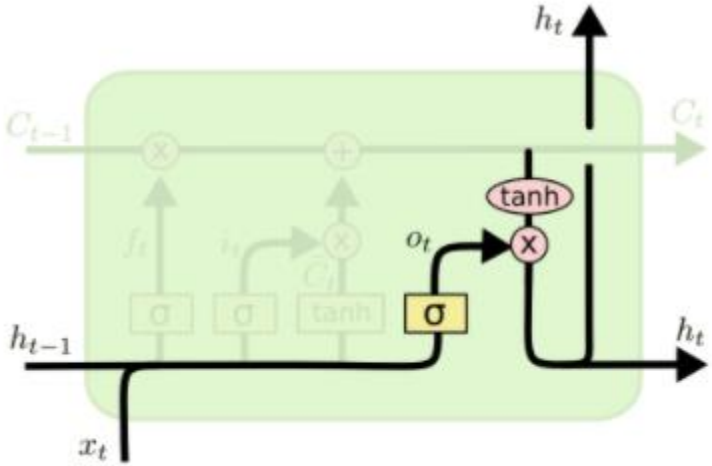
Long short-term Memory: update cell state



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Hochreiter & Schmidhuber (1997)
http://www.bioinf.jku.at/publications/older/2604.pdf

Long short-term Memory: output

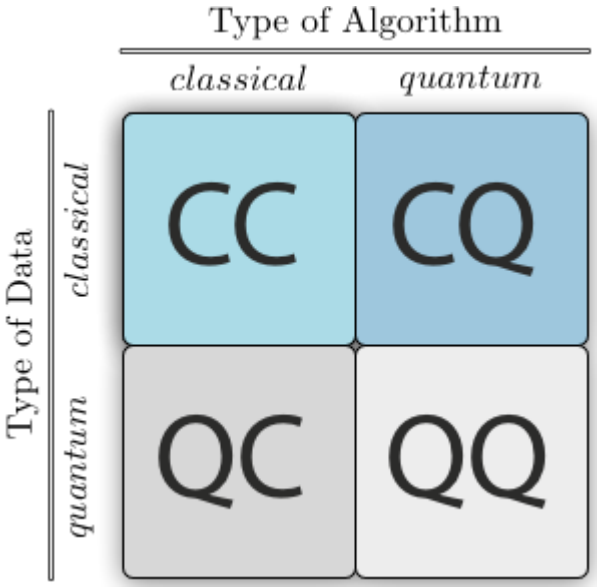


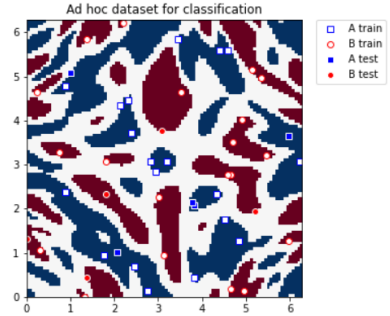
$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

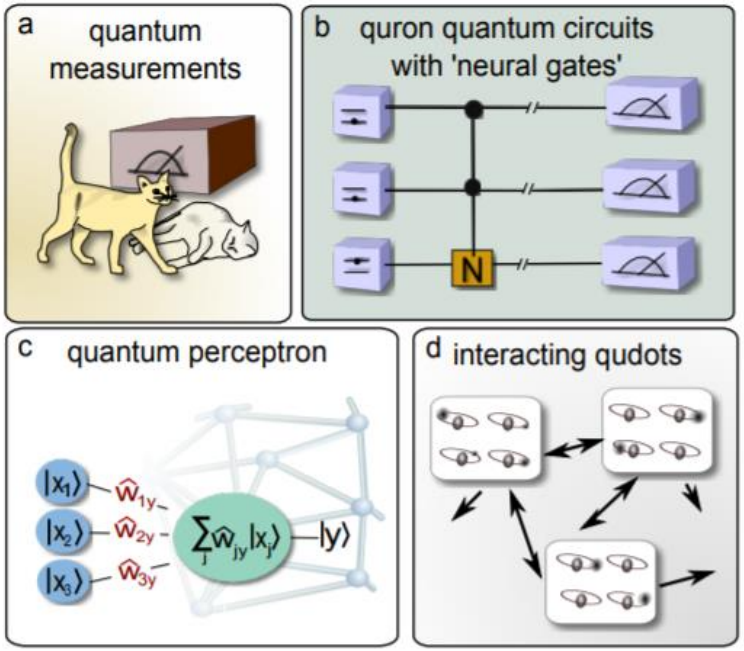
$$h_t = o_t * \tanh\left(C_t\right)$$

Hochreiter & Schmidhuber (1997)
http://www.bioinf.jku.at/publications/older/2604.pdf

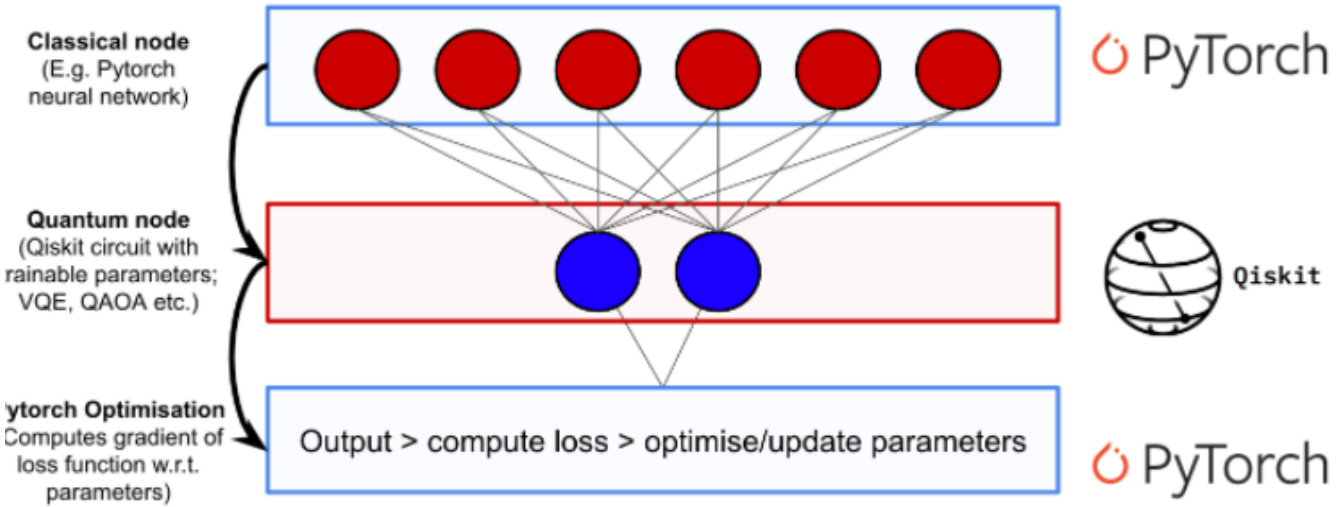
Quantum Machine learning



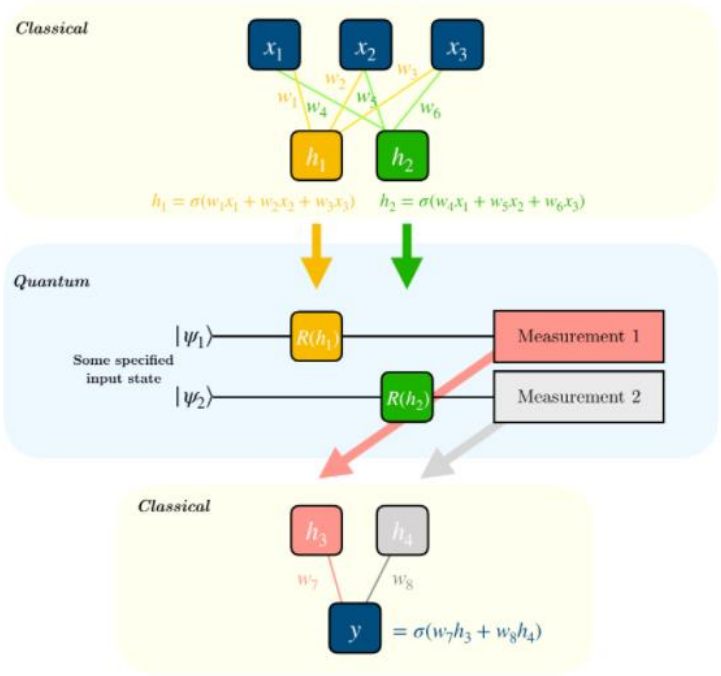




Hybrid quantum-classical Neural Networks with PyTorch and Qiskit

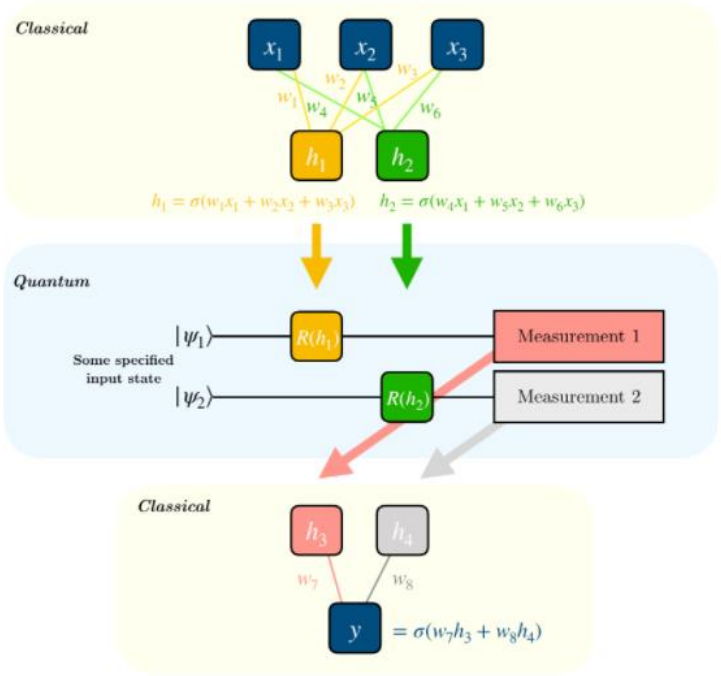


Quantum Machine learning Qiskit



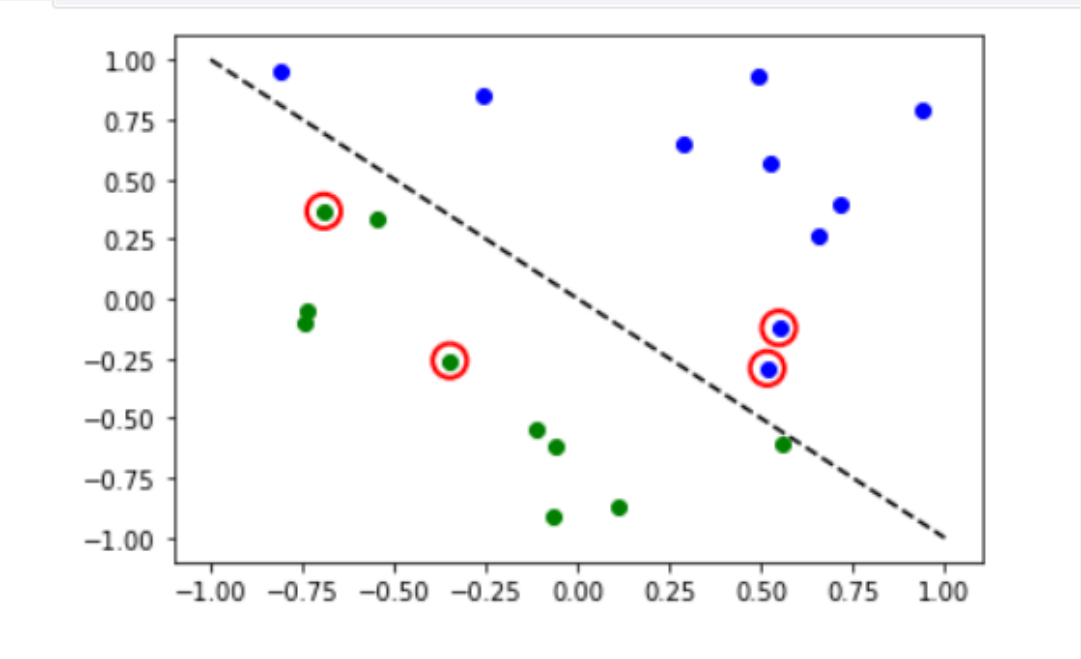
$$\nabla_\theta \text{Quantum Circuit}(\theta) = \text{Quantum Circuit}(\theta + s) - \text{Quantum Circuit}(\theta - s)$$

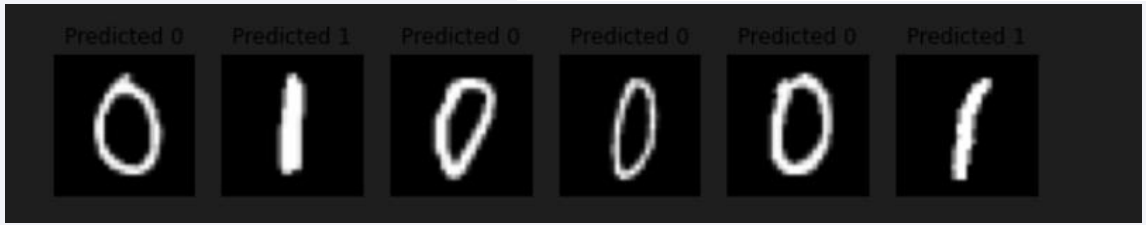
Quantum Machine learning Qiskit



$$\nabla_\theta \text{Quantum Circuit}(\theta) = \text{Quantum Circuit}(\theta + s) - \text{Quantum Circuit}(\theta - s)$$

Some Results





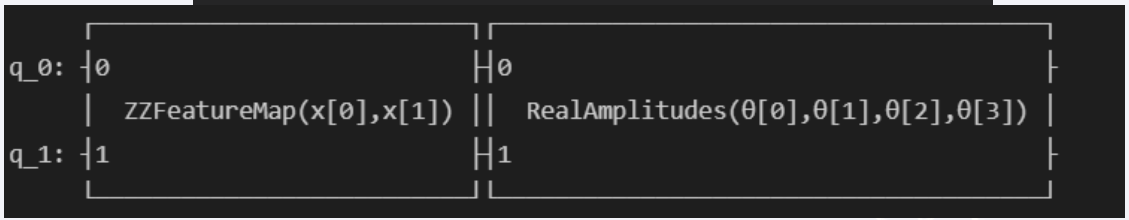

```
Performance on test data:
        Loss: -0.9287
        Accuracy: 100.0%
```

Classical

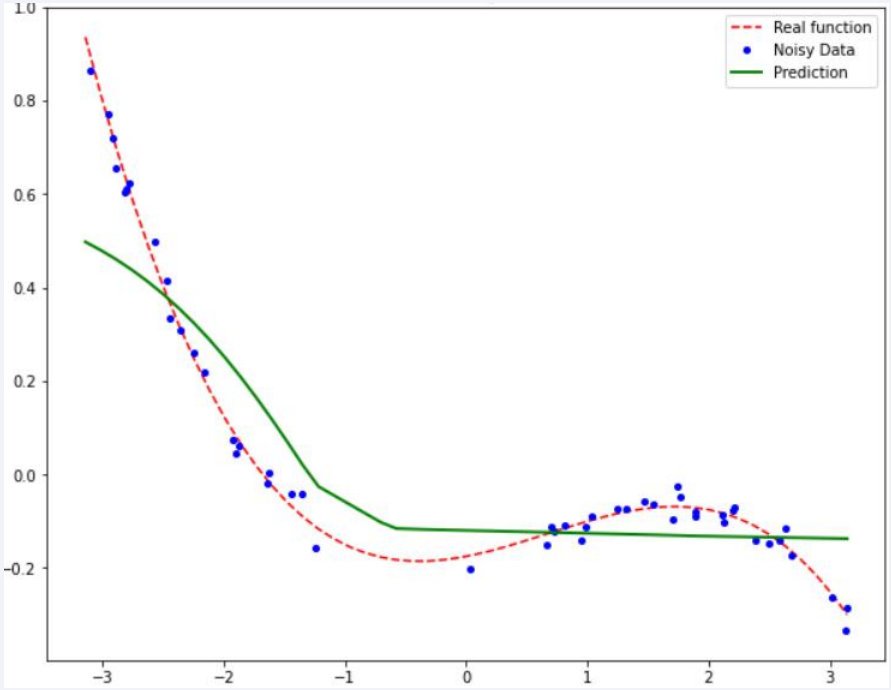
```python
class Net_c(nn.Module):
    def __init__(self):
        super(Net_c, self).__init__()
        self.flatten = nn.Flatten()
        self.fc1 = nn.Linear(1, 4)
        self.fc11 = nn.Linear(4, 6)
        self.fc12 = nn.Linear(6, 2)
        self.fc13 = nn.Linear(2, 1)
        self.qnn = TorchConnector(qnn1)
        #self.hybrid = Hybrid(qiskit.Aer

    def forward(self, x):
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc11(x))
        x = F.tanh(self.fc12(x))
        x = F.tanh(self.fc13(x))

        return x
```

Quantum

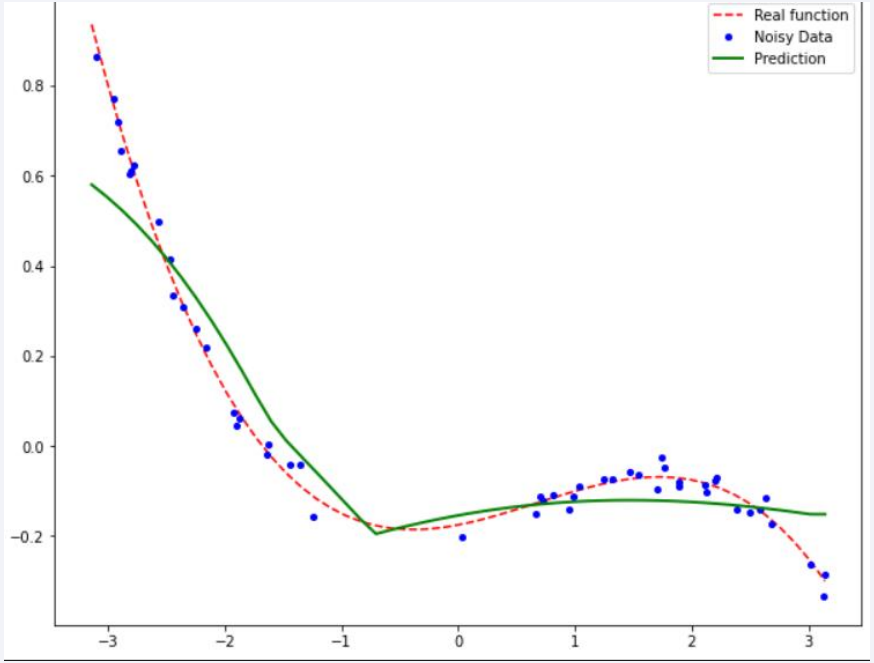
```python
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.flatten = nn.Flatten()
        self.fc1 = nn.Linear(1, 4)
        self.fc11 = nn.Linear(4, 6)
        self.fc12 = nn.Linear(6, 2)
        #self.hybrid = Hybrid(qiskit.Aer

    def forward(self, x):
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc11(x))
        x = F.tanh(self.fc12(x))
        x = self.qnn(x)
        return x
```



~~Clas~~sical

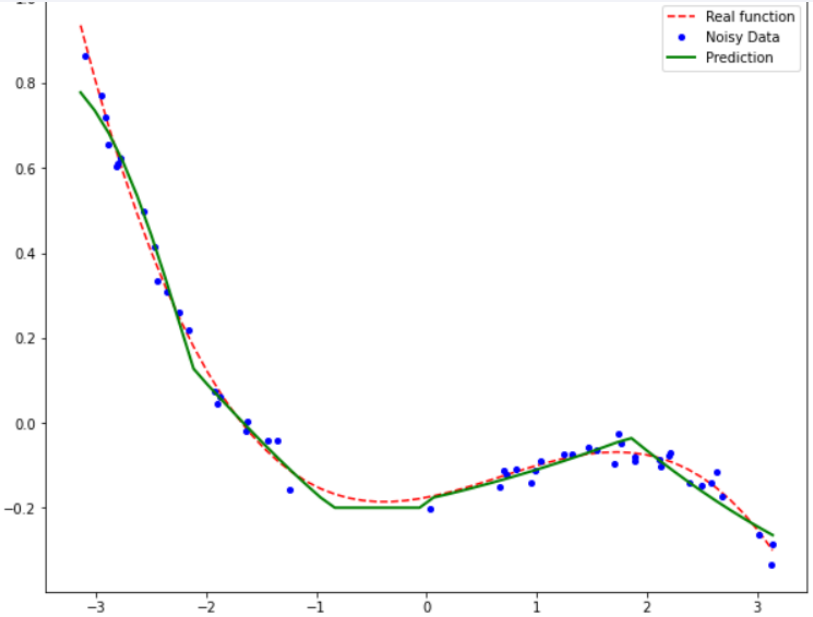


Quantum

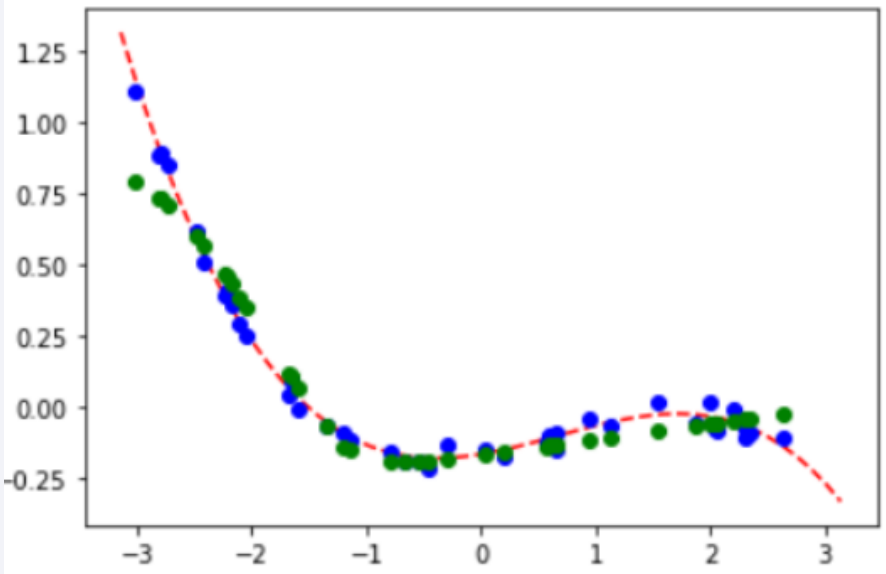


Qiskit

Classical



Quantum



Qiskit

Future Work

- Start stripping a LSTM NN and replace their components by quantum parts.

- Create a LSTM NN to predict stock prices!

Thank you!