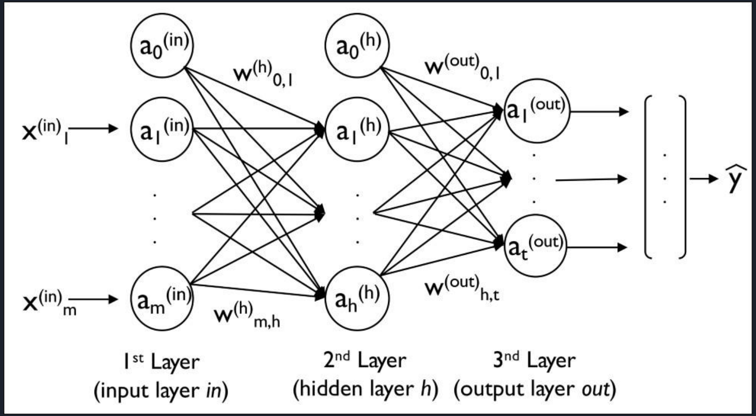
Quantum Neural Network Classifier

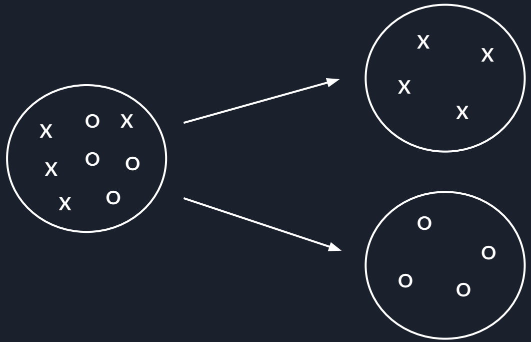
INTRO TO QUANTUM COMPUTING

Team Entangled Neurons:

Kulrajbir Singh Sandhu
Jasleen Bains
Puneet Anand
Tanya
Alamjot Singh

Classical Neural Network Classifier (CNN Classifier)





Input

[x1, x2, x3, … xn]

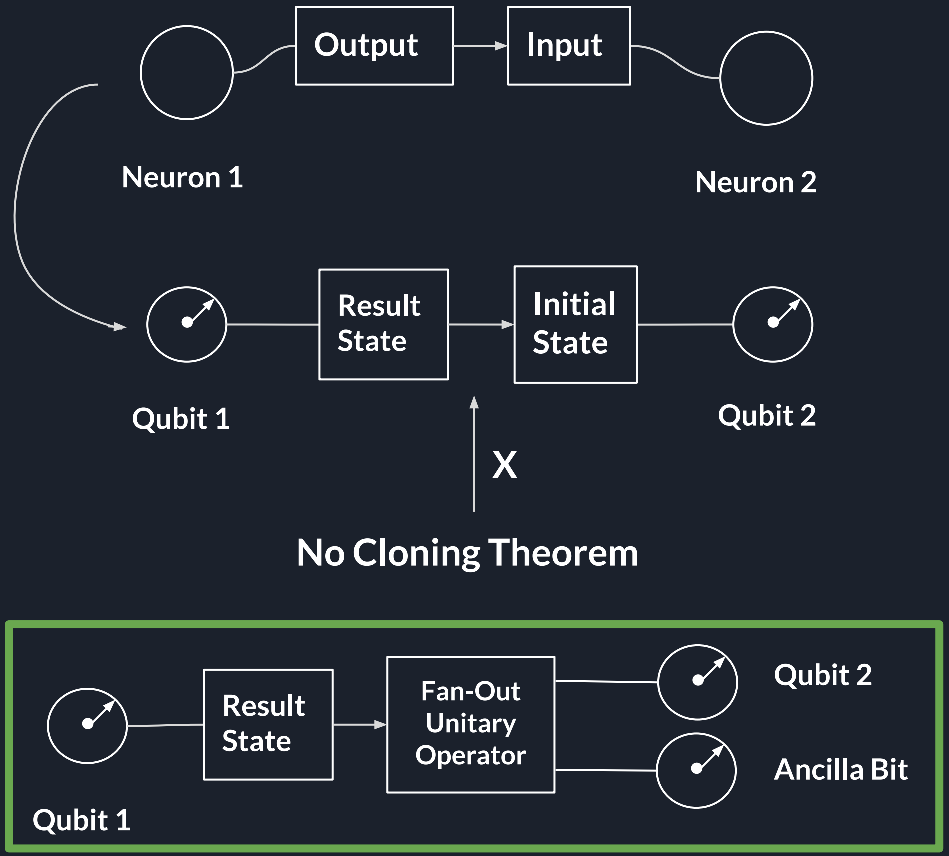
Output

[p1, p2, p3, … pn]

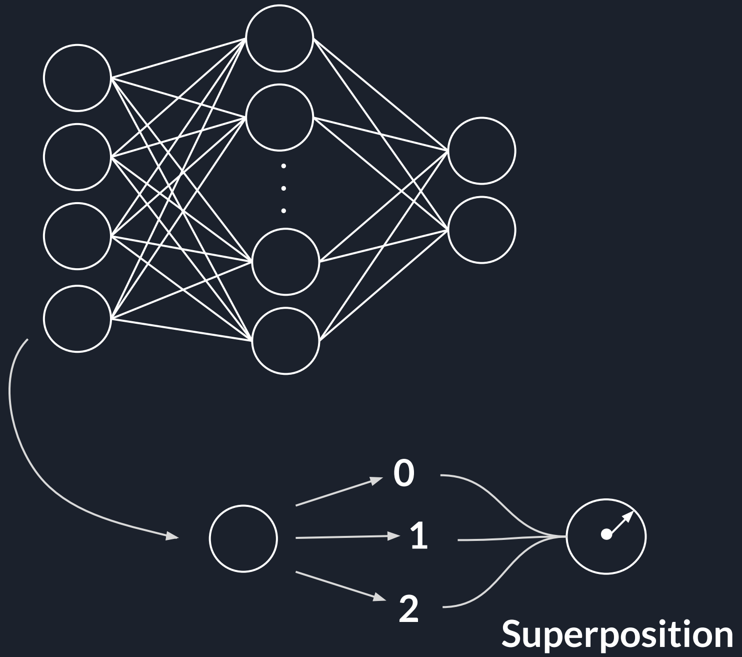
Classification

X or **O**

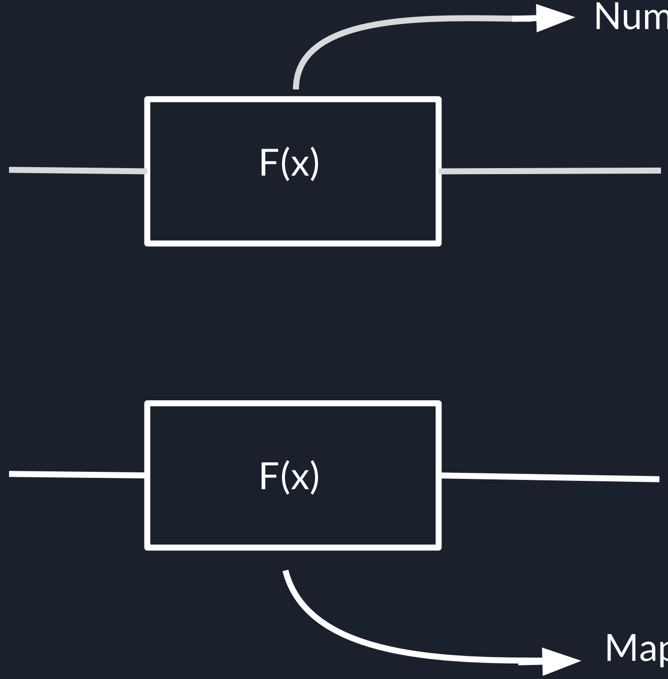
Implementation of Quantum NN Classifier



Quantum Benefit



Estimator and Sampler (Working)



Estimator

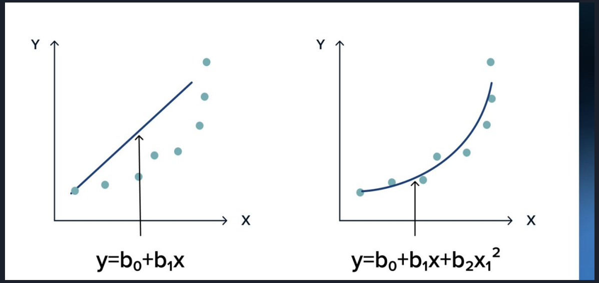
F(x)

Numerical

Sampler

F(x)

Mapping

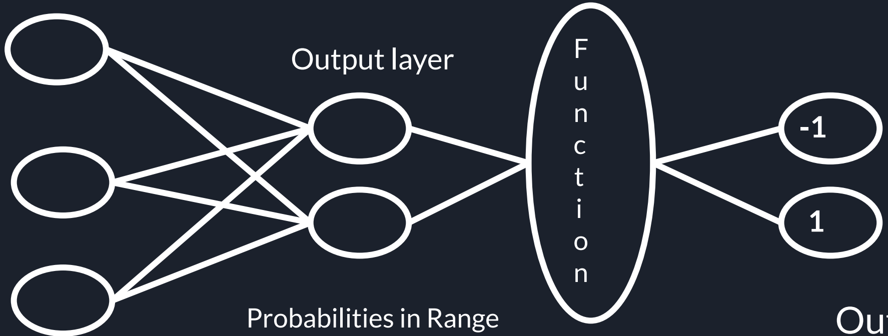


Finding an equation to find y

$$f(x)=\begin{cases} 1, & if\ x > 0 \\ 0, & if\ x = 0 \\ -1, & if\ x < 0 \end{cases}$$

Mapping x to classes
x>0, x=0, x<0 conditions of classes

Classification with Estimator QNN



Input

Weight

Output layer

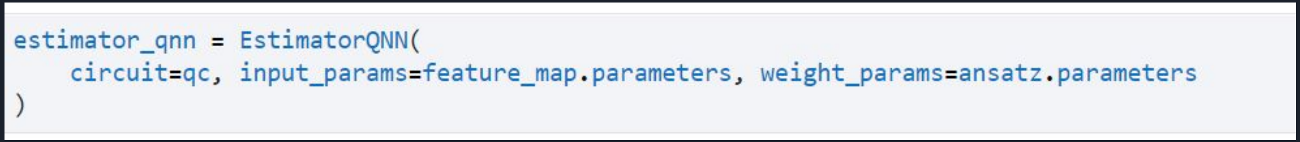
Second last layer of neural network

Probabilities in Range (-1, +1)

Estimator function

Output (class)

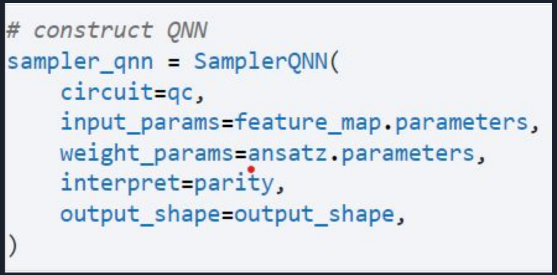
Quantum Circuit



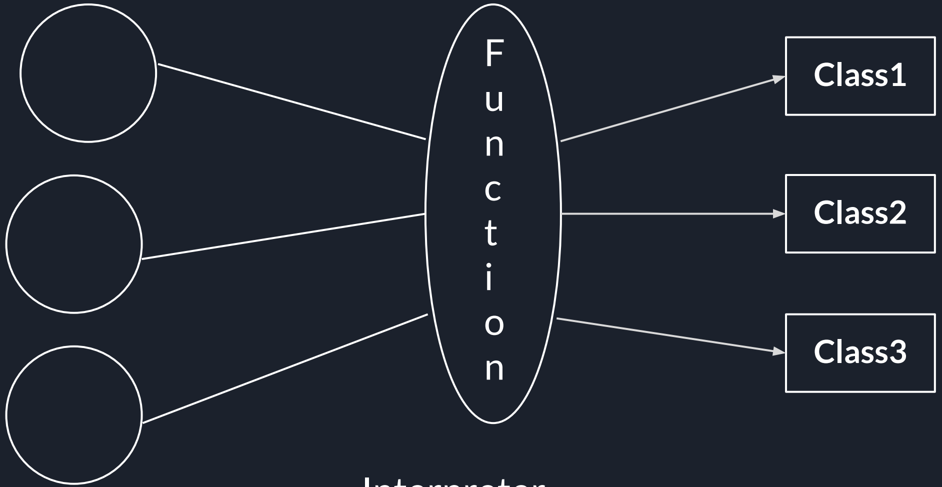
```
estimator_qnn = EstimatorQNN(
    circuit=qc, input_params=feature_map.parameters, weight_params=ansatz.parameters
)
```

Classification with a Sampler QNN

Quantum Circuit
Input
Weights
Interpreter function
Output Shape



```
# construct QNN
sampler_qnn = SamplerQNN(
    circuit=qc,
    input_params=feature_map.parameters,
    weight_params=ansatz.parameters,
    interpret=parity,
    output_shape=output_shape,
)
```

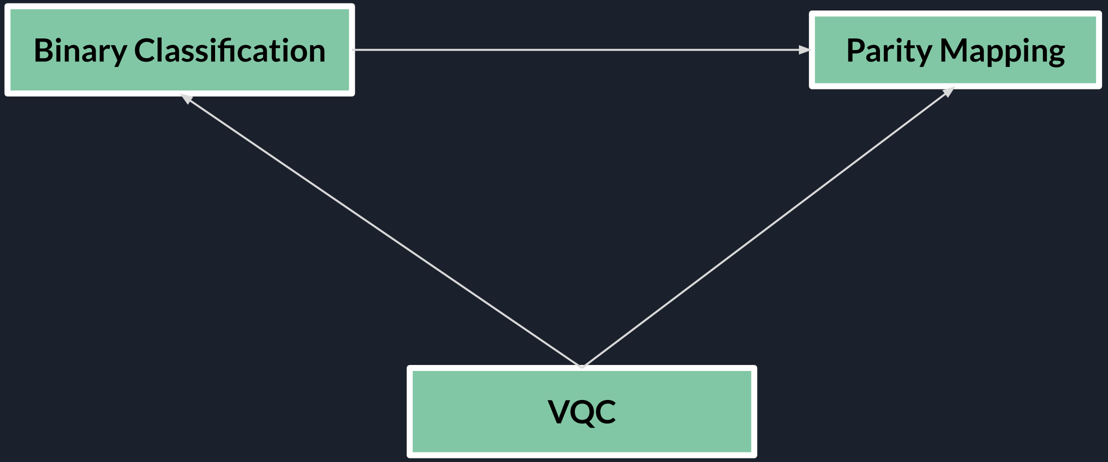


Interpreter

Output shape
(probabilities)

Variational Quantum Classifier (VQC)

- Special variant of NeuralNetworkClassifier with a SamplerQNN
- Produces the d-dimensional probability vector as one-hot encoded result
- Applies the CrossEntropyLoss function, by default.



Binary Classification

Parity Mapping

VQC

Probability vector in SamplerQNN is *d*-dimensional where d indicates the number of classes.

Conclusion

Thank you!

Highlights

- Quantum Neural Network Classifier and it’s Implementations
- Classification with EstimatorQNN, SamplerQNN and VQC
- Regression (Simple Dataset)
- Questions?