Create Qiskit Textbook Sections for Unsupervised Learning #4

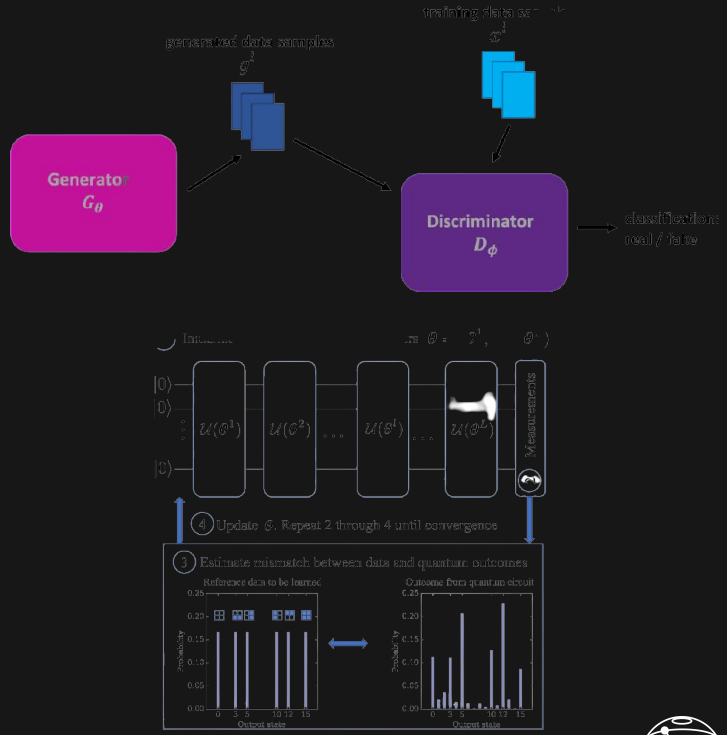
Anna Phan, Mentor

Pavan Jayasinha, Mentee

Alberto Maldonado Romo, Mentee

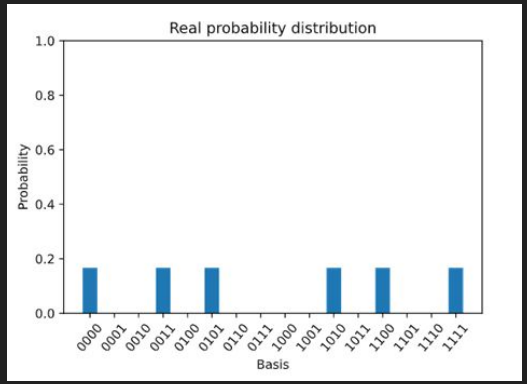
Project deliverables

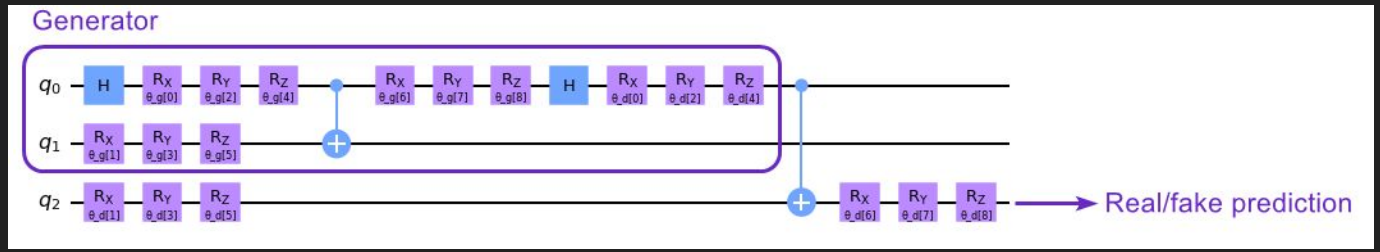
1. Textbook chapter incorporating interactive demos explaining **Quantum Generative Adversarial Networks**
2. Textbook chapter incorporating an interactive demo on **Quantum Boltzmann Machines**



Quantum generative adversarial networks chapter

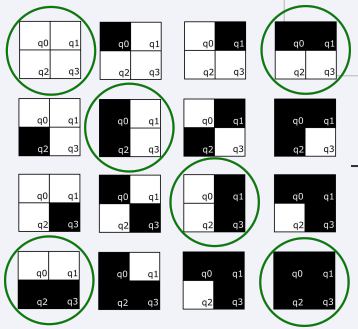
- Outlined clear structure
- Quantum generative adversarial network built natively on qiskit machine learning resources

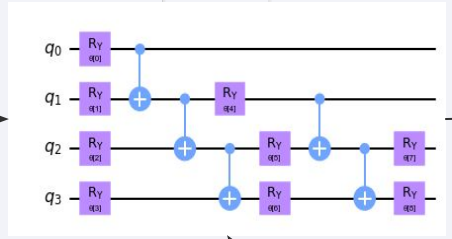




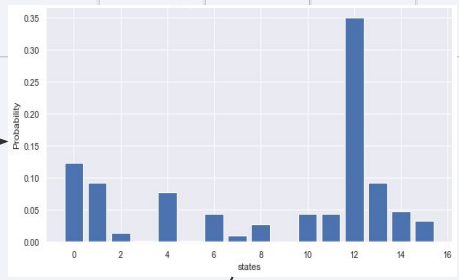
Quantum Boltzmann Machine

Use structural knowledge to define your state preparation problem

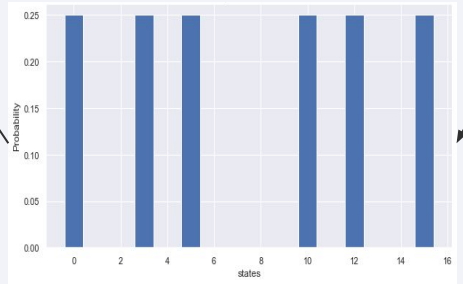




p(x)



θ update



Optimization

Optimization part

Cost function

$$C(\boldsymbol{\theta}) = -\frac{1}{D}\sum_{d=1}^{D} \ln\left(\max\left(\varepsilon, P_{\boldsymbol{\theta}}\left(\mathbf{x}^{(d)}\right)\right)\right)$$

```
def boltzman_machine(params):
    n=4
    D = int(n**2)
    cost = 0
    list_n = range(n)

    qc = QuantumCircuit(n)
    for i in range(len(params)//num_parameters):
        qc.append(gate_layer(n,params[num_parameters*i:num_parameters*(i+1)]),list_n)
    shots= 8192
    simulator = Aer.get_backend('statevector_simulator')
    result = execute(qc, simulator).result()
    statevector = result.get_statevector(qc)
    for j in range(D):
        cost +=(np.log10(max(0.001,statevector[j].real*output_expect[j].real+(statevector[j].imag*outpu
    cost = -cost/D
    #print(cost)
    return cost
```


Obtain from Benedetti, M., Garcia-Pintos, D., Perdomo, O. *et al.* A generative modeling approach for benchmarking and training shallow quantum circuits. *npj Quantum Inf* **5,** 45 (2019). https://doi.org/10.1038/s41534-019-0157-8