Swap Test

July 16, 2021

[1]:
```
from frozen_yoghourt import *
from swap_test import *
```

```
Duplicate key in file '/Users/minhpham/.matplotlib/matplotlibrc', line 2
('backend: TkAgg')
Duplicate key in file '/Users/minhpham/.matplotlib/matplotlibrc', line 3
('backend: TkAgg')
/Library/Frameworks/Python.framework/Versions/3.8/lib/python3.8/site-
packages/qiskit/aqua/operators/operator_globals.py:48: DeprecationWarning:
`from_label` is deprecated and will be removed no earlier than 3 months after
the release date. Use Pauli(label) instead.
  X = make_immutable(PrimitiveOp(Pauli.from_label('X')))
```

0.0.1 Training

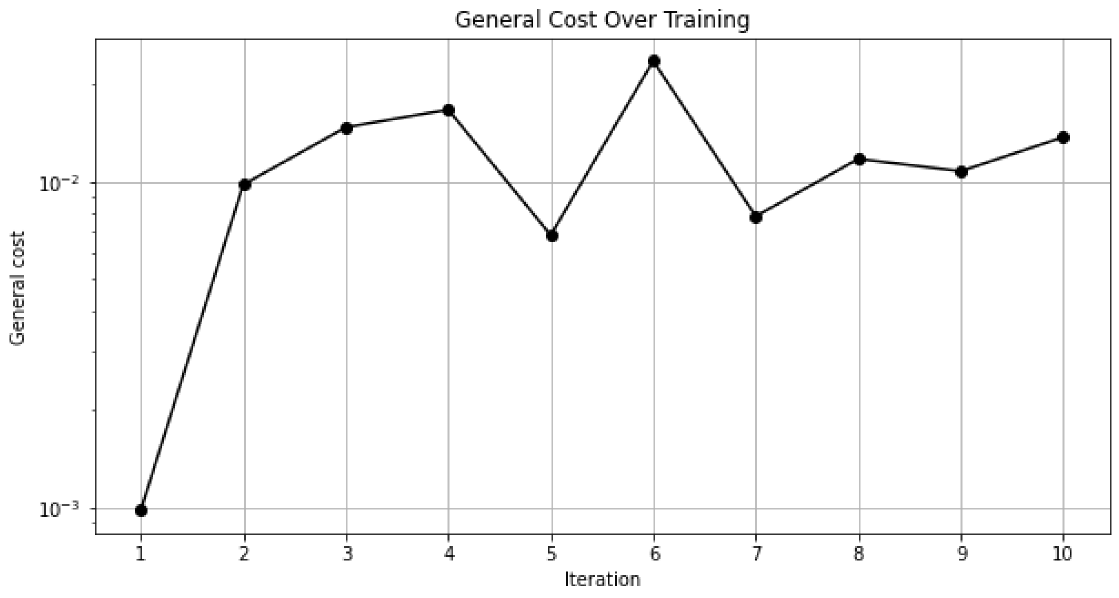
[61]:
```
n = 3
order = [2, 0, 1, 0, 1, 2]

angles = np.random.uniform(0, 2*np.pi, len(order)*3+3)

angles = np.array([2.05319891, 5.40103283, 1.43042012, 5.51275836, 4.92298739,
       3.77899832, 6.00581757, 2.61364636, 2.58550785, 0.71379243,
       3.41054398, 3.75929438, 2.48863724, 4.6144643 , 1.09507972,
       4.82638581, 2.56634188, 5.9678315 , 1.01855075, 1.21909221,
       3.78572889])
```

[62]:
```
def general_cost(angles):

    ### Circ 1
    circ = q(6, 1)

    circ.h([0, 1, 2])

    circ.cx(3, 4)

    circ.barrier()

    circ.mct([0, 1, 2], 3)
```

```python
circ.barrier()
circ = circ.compose(general_circ(n, order, angles), [0, 1, 2, 4])

circ.barrier()

circ.h(5)
circ.cswap(5, 4, 3)
circ.h(5)

circ.measure(5, 0)

### Loss 1

prob0 = sim(circ, None)['0']/1024
loss0 = 2*(1-prob0)

### Circ 2
circ = q(6, 1)

circ.h([0, 1, 2])

circ.x(3)
circ.cx(3, 4)

circ.barrier()

circ.mct([0, 1, 2], 3)

circ.barrier()
circ = circ.compose(general_circ(n, order, angles), [0, 1, 2, 4])

circ.barrier()

circ.h(5)
circ.cswap(5, 4, 3)
circ.h(5)

circ.measure(5, 0)

### Loss 2

prob0 = sim(circ, None)['0']/1024
loss1 = 2*(1-prob0)

### Cost 1
cost = (loss0 + loss1)/2
```

```
    return cost
```

[64]:
```
# Further Optimization Iterations

%matplotlib inline

reps = 10

idx = []
cost1 = []
# cost2 = []

for j in tqdm(range(reps)):
    result = sp.optimize.minimize(general_cost, angles, method = "COBYLA" )
    angles = result.x

    # Cost 1
    cost1.append(general_cost(angles))

    '''# Cost 2
    circ = general_circ(n, order, angles)
    cost2.append(cx_diff(np.abs
    (get(circ, nice = False)), n))'''

    idx.append(j + 1)
    live_plot(idx, cost1, 10)
```



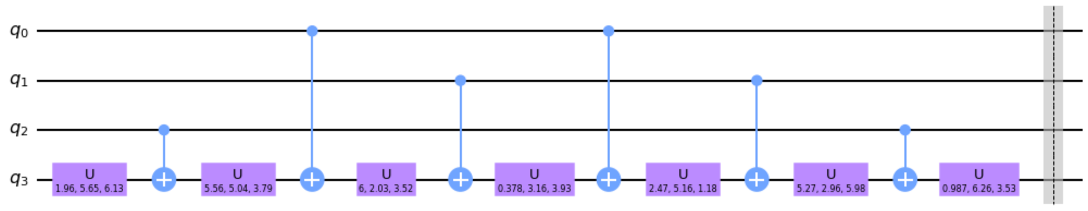
```
100%|     | 10/10 [00:34<00:00,  3.41s/it]
```

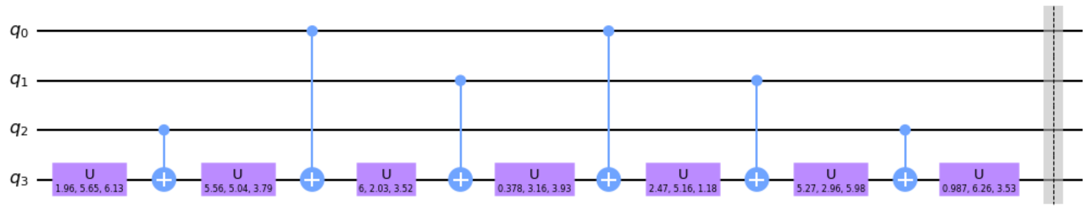
[65]:
```
result.x
```

[65]:
```
array([1.9586629 , 5.64571659, 6.13339897, 5.56451767, 5.03853348,
       3.78734464, 5.99569838, 2.03193286, 3.5152344 , 0.37752069,
       3.16108512, 3.9310483 , 2.46561439, 5.16252921, 1.17649577,
       5.27346294, 2.9648937 , 5.98428532, 0.98716526, 6.25808287,
       3.52531344])
```

[66]:
```
# Draw Circuit

circ = general_circ(n, order, result.x)

circ.barrier()

milk(circ)
```

[66]:





[67]:
```
# View Unitary
view(np.abs(get(circ, nice = False)))
```

$$\begin{bmatrix}
0.9998074088 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & \\
0.0 & 0.99908008 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & \\
0.0 & 0.0 & 0.997768715 & 0.0 & 0.0 & 0.0 & 0.0 & \\
0.0 & 0.0 & 0.0 & 0.9693940397 & 0.0 & 0.0 & 0.0 & \\
0.0 & 0.0 & 0.0 & 0.0 & 0.9995680498 & 0.0 & 0.0 & \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.9872959973 & 0.0 & \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.9961718833 & \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.07 \\
0.0196251174 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & \\
0.0 & 0.0428834914 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & \\
0.0 & 0.0 & 0.0667651952 & 0.0 & 0.0 & 0.0 & 0.0 & \\
0.0 & 0.0 & 0.0 & 0.2455100725 & 0.0 & 0.0 & 0.0 & \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0293890087 & 0.0 & 0.0 & \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.1588918302 & 0.0 & \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0874161254 & \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.99
\end{bmatrix}$$

[68]:
```
cx_diff(circ, 3)
```

[68]:
```
1.5481567327522454
```