Swap Test

July 16, 2021

[533]:
```
from frozen_yoghourt import *
from swap_test import *
from tqdm import tqdm
```

0.0.1 Training

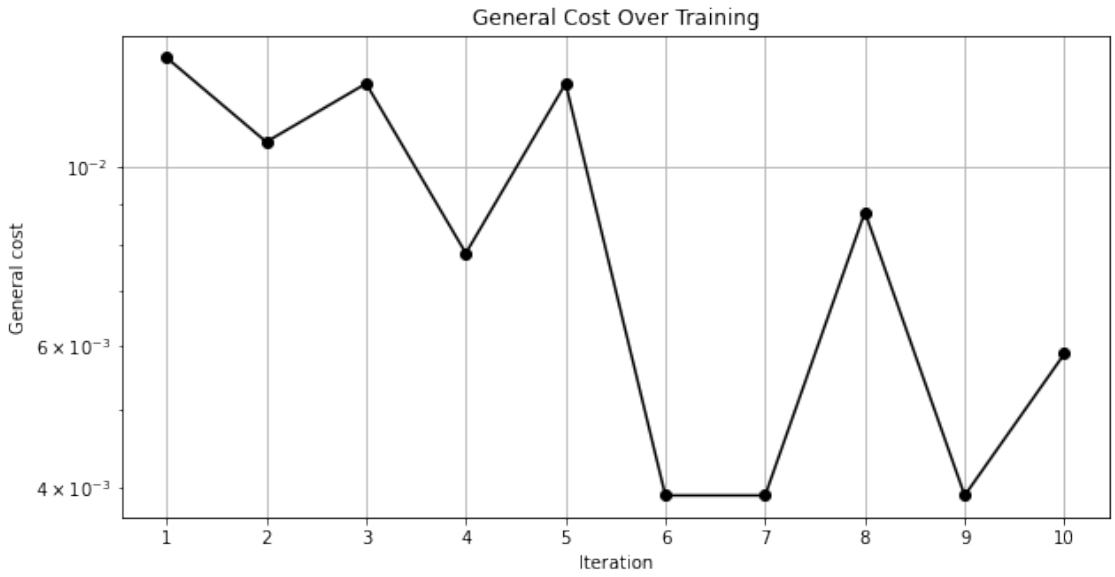
[535]:
```
n = 2
order = [0, 1, 0, 1]

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

[536]:
```
def general_cost(angles):

    ### Circ 1
    circ = q(5, 1)

    circ.h([0, 1])

    circ.cx(2, 3)

    circ.barrier()

    circ.ccx(0, 1, 2)

    circ.barrier()
    circ = circ.compose(general_circ(2, order, angles), [0, 1, 3])

    circ.barrier()

    circ.h(4)
    circ.cswap(4, 3, 2)
    circ.h(4)

    circ.measure(4, 0)

    ### Loss 1
```

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

    ### Circ 2
    circ = q(5, 1)

    ### Input

    circ.h([0, 1])
    circ.x(2)
    circ.cx(2, 3)

    circ.barrier()

    circ.ccx(0, 1, 2)

    circ.barrier()
    circ = circ.compose(general_circ(2, order, angles), [0, 1, 3])

    circ.barrier()

    circ.h(4)
    circ.cswap(4, 3, 2)
    circ.h(4)

    circ.measure(4, 0)

    ### Loss 2

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

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

    return cost
```

[527]:
```
# Further Optimization Iterations

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)
```



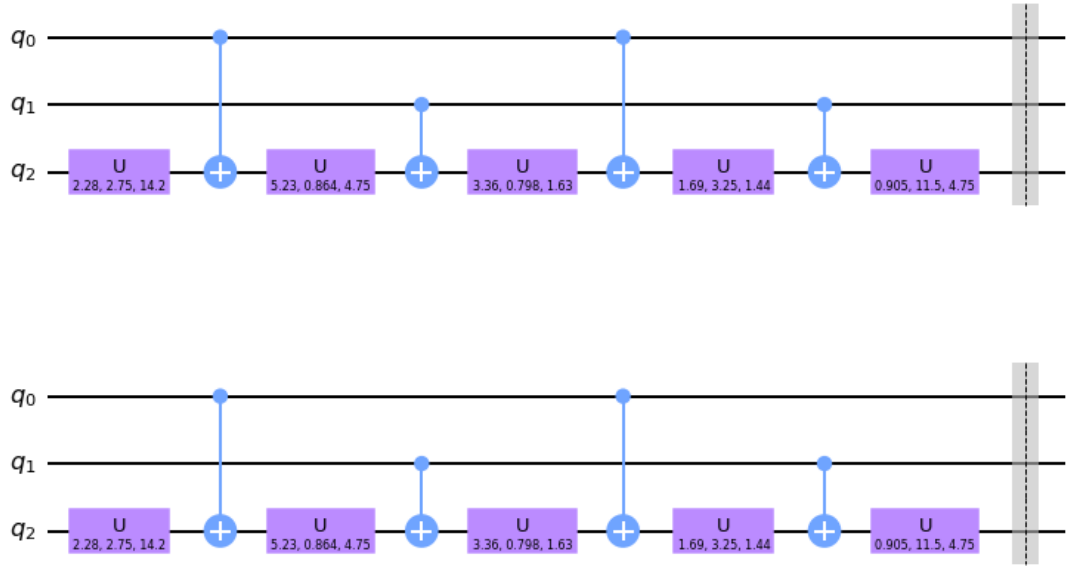

```
100%|      | 10/10 [00:19<00:00,  1.99s/it]
```

[528]:
```
result.x
```

[528]:
```
array([ 2.28365726,  2.74506308, 14.17702796,  5.22896253,  0.86439628,
        4.74671865,  3.3558497 ,  0.79755108,  1.62528628,  1.68802097,
        3.25114341,  1.44444138,  0.90453004, 11.49881604,  4.75330568])
```

[529]:
```
# Draw Circuit

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

circ.barrier()

milk(circ)
```

[529]:



[530]:
```
# View Unitary
view(np.abs(get(circ, nice = False)))
```

$$\begin{bmatrix} 0.9938487531 & 0.0 & 0.0 & 0.0 & 0.1107459076 & 0.0 & 0.0 & 0. \\ 0.0 & 0.997631773 & 0.0 & 0.0 & 0.0 & 0.068781142 & 0.0 & 0. \\ 0.0 & 0.0 & 0.9984910599 & 0.0 & 0.0 & 0.0 & 0.0549145092 & 0. \\ 0.0 & 0.0 & 0.0 & 0.0420063765 & 0.0 & 0.0 & 0.0 & 0.9991 \\ 0.1107459076 & 0.0 & 0.0 & 0.0 & 0.9938487531 & 0.0 & 0.0 & 0. \\ 0.0 & 0.068781142 & 0.0 & 0.0 & 0.0 & 0.997631773 & 0.0 & 0. \\ 0.0 & 0.0 & 0.0549145092 & 0.0 & 0.0 & 0.0 & 0.9984910599 & 0. \\ 0.0 & 0.0 & 0.0 & 0.9991173426 & 0.0 & 0.0 & 0.0 & 0.0420 \end{bmatrix}$$

[ ]:
```

```