File - performance_test (1)

```
C:\Users\usuario\Anaconda3\envs\tfq\python.exe C:/Users/usuario/qGAN/quantumGAN/performance_testing/performance_test.py
[0.68629349 0.21612429 0.31385589 0.78720836 0.87458356 0.76394449]
Epoch 0: Loss: [-0.54206342] [0.46076452 0.         0.46971214 0.        ] [0.08642578 0.38427734 0.44726562 0.08203125]
[0.08991251] [0.08366766]
Epoch 1: Loss: [-0.39259394] [0.42656312 0.         0.42886134 0.        ] [0.08398438 0.39160156 0.43798828 0.08642578]
[0.20266537] [0.19084343]
Epoch 2: Loss: [-0.32400884] [0.40703161 0.         0.49894454 0.        ] [0.07714844 0.37011719 0.46875    0.08398438]
[0.3261466] [0.31044413]
Epoch 3: Loss: [-0.30168559] [0.44454905 0.         0.47212619 0.        ] [0.07958984 0.36914062 0.46240234 0.08886719]
[0.41255232] [0.39584304]
Epoch 4: Loss: [-0.29512413] [0.43229261 0.         0.44304952 0.        ] [0.07666016 0.34472656 0.47802734 0.10058594]
[0.45897347] [0.4402886]
Epoch 5: Loss: [-0.29339179] [0.40311052 0.         0.42179919 0.        ] [0.07177734 0.3828125  0.46044922 0.08496094]
[0.48268929] [0.4635259]
Epoch 6: Loss: [-0.29205014] [0.44368162 0.         0.45882899 0.        ] [0.06640625 0.35302734 0.49658203 0.08398438]
[0.49575779] [0.47443048]
Epoch 7: Loss: [-0.29161389] [0.44368162 0.         0.45882899 0.        ] [0.06298828 0.34423828 0.50634766 0.08642578]
[0.50196255] [0.47988316]
Epoch 8: Loss: [-0.29057427] [0.43229261 0.         0.44304952 0.        ] [0.06982422 0.35693359 0.49316406 0.08007812]
[0.50482675] [0.48035219]
Epoch 9: Loss: [-0.29016866] [0.46846721 0.         0.40921075 0.        ] [0.06396484 0.34863281 0.50048828 0.08691406]
[0.50714952] [0.48176509]
Timer unit: 1e-07 s

Total time: 243.193 s
File: C:/Users/usuario/qGAN/quantumGAN/performance_testing/performance_test.py
Function: mainV1 at line 13

Line #      Hits         Time  Per Hit   % Time  Line Contents
==============================================================
    13                                           def mainV1():
    14
    15         1         33.0     33.0      0.0     seed = 71
    16         1         48.0     48.0      0.0     np.random.seed = seed
    17
    18         1         20.0     20.0      0.0     num_qubits = [2]
    19         1         18.0     18.0      0.0     batch_size = 10
    20         1         19.0     19.0      0.0     entangler_map = [[0, 1]]
    21
    22         1        756.0    756.0      0.0     randoms = np.random.normal(-np.pi * .01, np.pi * .01, 2)
    23
    24         1       2508.0   2508.0      0.0     init_dist = qiskit.QuantumCircuit(2)
    25         1       1730.0   1730.0      0.0     init_dist.ry(randoms[0], 0)
    26         1        747.0    747.0      0.0     init_dist.ry(randoms[1], 1)
    27
    28         1   20253007.0 20253007.0      0.8   ansatz = TwoLocal(int(np.sum(num_qubits)), 'rx', 'cz', entanglement=
   entangler_map, reps=2, insert_barriers=True)
    29
    30         1         32.0     32.0      0.0     train_data = []
    31        16        315.0     19.7      0.0     for _ in range(15):
    32        15       1745.0    116.3      0.0         x2 = np.random.uniform(.5, .4, (2,))
    33        15       1234.0     82.3      0.0         fake_datapoint = np.random.uniform(-np.pi * .01, np.pi * .01, (2
   ,))
    34        15       1024.0     68.3      0.0         real_datapoint = np.array([x2[1], 0., x2[0], 0])
    35        15        333.0     22.2      0.0         train_data.append((real_datapoint, fake_datapoint))
    36
    37         1     126329.0 126329.0      0.0     g_circuit = ansatz.compose(init_dist, front=True)
    38
    39         1         48.0     48.0      0.0     discriminator = Network(training_data=train_data,
    40         1         20.0     20.0      0.0                             mini_batch_size=batch_size,
    41         1         23.0     23.0      0.0                             sizes=[4, 16, 8, 1],
    42         1        881.0    881.0      0.0                             loss_BCE=True)
    43         1         30.0     30.0      0.0     generator = PerformanceQuantumGeneratorV1(training_data=train_data,
    44         1         19.0     19.0      0.0                                   mini_batch_size=batch_size,
    45         1         19.0     19.0      0.0                                   num_qubits=num_qubits,
    46         1         18.0     18.0      0.0                                   generator_circuit=g_circuit,
    47         1         18.0     18.0      0.0                                   shots=2048,
    48         1       7286.0   7286.0      0.0                                   learning_rate=.1)
    49         1         47.0     47.0      0.0     generator.set_discriminator(discriminator)
    50
    51        11        384.0     34.9      0.0     for o in range(num_epochs):
    52        10       8660.0    866.0      0.0         mini_batches = discriminator.create_mini_batches()
    53        30       1085.0     36.2      0.0         for mini_batch in mini_batches:
    54        20        506.0     25.3      0.0             output_real = mini_batch[0][0]
    55        20        628.0     31.4      0.0             output_fake = generator.get_output(latent_space_noise=
   mini_batch[0][1],
    56        20   27501832.0 1375091.6     1.1                                                params=None)
    57        20       2399.0    120.0      0.0             generator.set_mini_batch(mini_batch)
    58        20        439.0     21.9      0.0             generator.shots = 2048
    59        20 2382921775.0 119146088.8     98.0                 generator.train_mini_batch()
    60
    61        20     889522.0  44476.1      0.0             discriminator.train_mini_batch(generator.mini_batch, .1, o)
    62        10     139298.0  13929.8      0.0         print("Epoch {}: Loss: {}".format(o, discriminator.ret["loss"][-
   1]), output_real, output_fake)
    63        10      65455.0   6545.5      0.0         print(discriminator.ret["label real"][-1], discriminator.ret["
   label fake"][-1])


Process finished with exit code 0
```