File - performance_testv2

```
C:\Users\usuario\Anaconda3\envs\tfq\python.exe C:/Users/usuario/qGAN/quantumGAN/performance_testing/performance_testv2.py
[0.69598666 0.68751602 0.36615615 0.88484289 0.94954178 0.53794645]
Epoch 0: Loss: [-0.33465227] [0.43880991 0.         0.48241925 0.        ] [0.01123047 0.25195312 0.18115234 0.05566406]
[0.31611266] [0.32258693]
Epoch 1: Loss: [-0.31354394] [0.42479018 0.         0.48702081 0.        ] [0.00683594 0.25927734 0.18603516 0.04785156]
[0.39161368] [0.39736511]
Epoch 2: Loss: [-0.30744537] [0.46965315 0.         0.49932821 0.        ] [0.00878906 0.25341797 0.17675781 0.06103516]
[0.43694627] [0.44450362]
Epoch 3: Loss: [-0.30196221] [0.49838364 0.         0.47926745 0.        ] [0.01171875 0.25634766 0.17138672 0.06054688]
[0.46949657] [0.46979581]
Epoch 4: Loss: [-0.30079866] [0.48372133 0.         0.47530269 0.        ] [0.00927734 0.26513672 0.17626953 0.04931641]
[0.48430615] [0.48324737]
Epoch 5: Loss: [-0.30059916] [0.42479018 0.         0.48702081 0.        ] [0.01367188 0.26318359 0.16748047 0.05566406]
[0.49094533] [0.489767]
Epoch 6: Loss: [-0.29828735] [0.41758334 0.         0.48413402 0.        ] [0.01074219 0.26708984 0.17333984 0.04882812]
[0.49852013] [0.49214165]
Epoch 7: Loss: [-0.29705461] [0.48372133 0.         0.47530269 0.        ] [0.00878906 0.27099609 0.17529297 0.04492188]
[0.50320191] [0.49400236]
Epoch 8: Loss: [-0.29818675] [0.49011842 0.         0.42136107 0.        ] [0.01123047 0.26757812 0.17089844 0.05029297]
[0.50110241] [0.4945246]
Epoch 9: Loss: [-0.29641034] [0.45442178 0.         0.41738916 0.        ] [0.0078125  0.27246094 0.17285156 0.046875  ]
[0.50492518] [0.49423089]
Timer unit: 1e-07 s

Total time: 239.125 s
File: C:/Users/usuario/qGAN/quantumGAN/performance_testing/performance_testv2.py
Function: mainV2 at line 13

Line #      Hits         Time  Per Hit   % Time  Line Contents
==============================================================
    13                                           def mainV2():
    14
    15         1         29.0     29.0      0.0     seed = 71
    16         1         39.0     39.0      0.0     np.random.seed = seed
    17
    18         1         21.0     21.0      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        627.0    627.0      0.0     randoms = np.random.normal(-np.pi * .01, np.pi * .01, 2)
    23
    24         1       2120.0   2120.0      0.0     init_dist = qiskit.QuantumCircuit(2)
    25         1       1332.0   1332.0      0.0     init_dist.ry(randoms[0], 0)
    26         1        736.0    736.0      0.0     init_dist.ry(randoms[1], 1)
    27
    28         1   23792196.0 23792196.0      1.0   ansatz = TwoLocal(int(np.sum(num_qubits)), 'rx', 'cz', entanglement=entangler_map, reps=2, insert_barriers=True)
    29
    30         1         34.0     34.0      0.0     train_data = []
    31        16       3015.0    188.4      0.0     for _ in range(15):
    32        15       3543.0    236.2      0.0         x2 = np.random.uniform(.5, .4, (2,))
    33        15       1585.0    105.7      0.0         fake_datapoint = np.random.uniform(-np.pi * .01, np.pi * .01, (2,))
    34        15       1225.0     81.7      0.0         real_datapoint = np.array([x2[1], 0., x2[0], 0])
    35        15        349.0     23.3      0.0         train_data.append((real_datapoint, fake_datapoint))
    36
    37         1     135841.0 135841.0      0.0     g_circuit = ansatz.compose(init_dist, front=True)
    38
    39         1         39.0     39.0      0.0     discriminator = Network(training_data=train_data,
    40         1         18.0     18.0      0.0                             mini_batch_size=batch_size,
    41         1         20.0     20.0      0.0                             sizes=[4, 16, 8, 1],
    42         1       1005.0   1005.0      0.0                             loss_BCE=True)
    43         1         24.0     24.0      0.0     generator = PerformanceQuantumGeneratorV2(training_data=train_data,
    44         1         20.0     20.0      0.0                                   mini_batch_size=batch_size,
    45         1         18.0     18.0      0.0                                   num_qubits=num_qubits,
    46         1         19.0     19.0      0.0                                   generator_circuit=g_circuit,
    47         1         18.0     18.0      0.0                                   shots=2048,
    48         1       9012.0   9012.0      0.0                                   learning_rate=.1)
    49         1         44.0     44.0      0.0     generator.set_discriminator(discriminator)
    50
    51        11        328.0     29.8      0.0     for o in range(num_epochs):
    52        10       8054.0    805.4      0.0         mini_batches = discriminator.create_mini_batches()
    53        30       1141.0     38.0      0.0         for mini_batch in mini_batches:
    54        20        505.0     25.2      0.0             output_real = mini_batch[0][0]
    55        20        617.0     30.9      0.0             output_fake = generator.get_output(latent_space_noise=mini_batch[0][1],
    56        20   26846481.0 1342324.1      1.1                                                params=None)
    57        20       2233.0    111.7      0.0             generator.set_mini_batch(mini_batch)
    58        20        427.0     21.4      0.0             generator.shots = 2048
    59        20 2339600390.0 116980019.5     97.8             generator.train_mini_batch()
    60        20     653849.0  32692.5      0.0             discriminator.train_mini_batch(generator.mini_batch, .1, o)
    61        10     122598.0  12259.8      0.0         print("Epoch {}: Loss: {}".format(o, discriminator.ret["loss"][-1]), output_real, output_fake)
    62        10      58573.0   5857.3      0.0         print(discriminator.ret["label real"][-1], discriminator.ret["label fake"][-1])
81
82
83 Process finished with exit code 0
```