File - performance_test

```
C:\Users\usuario\Anaconda3\envs\tfq\python.exe C:/Users/usuario/qGAN/quantumGAN/performance_test.py
[0.83809885 0.59662554 0.08574274 0.23931475 0.13270372 0.67136706]
Epoch 0: Loss: [-0.51708544] [0.45231102 0.         0.4109069  0.        ] [0.40039062 0.37158203 0.17626953 0.05175781]
[0.10325458] [0.10480057]
Epoch 1: Loss: [-0.41802769] [0.44482384 0.         0.41277293 0.        ] [0.40478516 0.33886719 0.19580078 0.06054688]
[0.17808944] [0.18095745]
Epoch 2: Loss: [-0.3584017] [0.45722264 0.         0.43477556 0.        ] [0.38671875 0.35205078 0.21240234 0.04882812]
[0.26187903] [0.26701365]
Epoch 3: Loss: [-0.32810519] [0.4421636  0.         0.44236231 0.        ] [0.39794922 0.35693359 0.19091797 0.05419922]
[0.33543319] [0.34206411]
Epoch 4: Loss: [-0.3142884] [0.44340551 0.         0.47991339 0.        ] [0.39550781 0.36083984 0.19726562 0.04638672]
[0.39006103] [0.39703704]
Epoch 5: Loss: [-0.30841528] [0.45722264 0.         0.43477556 0.        ] [0.38867188 0.36132812 0.18994141 0.06005859]
[0.42679137] [0.43382093]
Epoch 6: Loss: [-0.30592869] [0.4858715  0.         0.47115998 0.        ] [0.40722656 0.33642578 0.20166016 0.0546875 ]
[0.4508546] [0.45786666]
Epoch 7: Loss: [-0.30485919] [0.45231102 0.         0.4109069  0.        ] [0.38818359 0.34423828 0.21875    0.04882812]
[0.46629032] [0.47322492]
Epoch 8: Loss: [-0.30391934] [0.42334983 0.         0.41691616 0.        ] [0.41894531 0.32714844 0.19970703 0.05419922]
[0.47623022] [0.48198257]
Epoch 9: Loss: [-0.30349532] [0.46876017 0.         0.4111282  0.        ] [0.40380859 0.34960938 0.20068359 0.04589844]
[0.48281253] [0.48804613]
Epoch 10: Loss: [-0.30314494] [0.42334983 0.         0.41691616 0.        ] [0.42285156 0.32666016 0.20117188 0.04931641
]
[0.48677289] [0.49139136]
Epoch 11: Loss: [-0.30280091] [0.4421636  0.         0.44236231 0.        ] [0.39501953 0.33251953 0.21582031 0.05664062
]
[0.48947248] [0.49339451]
Epoch 12: Loss: [-0.30229636] [0.4456309  0.         0.40199548 0.        ] [0.39306641 0.35449219 0.20019531 0.05224609
]
[0.49181477] [0.49463434]
Epoch 13: Loss: [-0.30212915] [0.44340551 0.         0.47991339 0.        ] [0.40185547 0.35839844 0.18847656 0.05126953
]
[0.49273141] [0.49518592]
Epoch 14: Loss: [-0.30190912] [0.4456309  0.         0.40199548 0.        ] [0.390625   0.34375    0.21484375 0.05078125
]
[0.49362745] [0.49559144]
Epoch 15: Loss: [-0.3015627] [0.4421636  0.         0.44236231 0.        ] [0.39941406 0.35302734 0.19921875 0.04833984]
[0.49440403] [0.49557966]
Epoch 16: Loss: [-0.30127617] [0.48331707 0.         0.49868168 0.        ] [0.40332031 0.34228516 0.19921875 0.05517578
]
[0.49496051] [0.49548149]
Epoch 17: Loss: [-0.30094585] [0.46876017 0.         0.4111282  0.        ] [0.40087891 0.34521484 0.20849609 0.04541016
]
[0.49573965] [0.49550759]
Epoch 18: Loss: [-0.3002969] [0.48331707 0.         0.49868168 0.        ] [0.40917969 0.34521484 0.19287109 0.05273438]
[0.49695639] [0.49523653]
Epoch 19: Loss: [-0.30048304] [0.4471763  0.         0.41560002 0.        ] [0.40722656 0.34033203 0.20361328 0.04882812
]
[0.49619388] [0.49489402]
Epoch 20: Loss: [-0.30017981] [0.4471763  0.         0.41560002 0.        ] [0.40234375 0.35595703 0.19677734 0.04492188
]
[0.49674341] [0.49474773]
Epoch 21: Loss: [-0.29940173] [0.48331707 0.         0.49868168 0.        ] [0.41357422 0.34814453 0.18701172 0.05126953
]
[0.49836712] [0.49458612]
Epoch 22: Loss: [-0.29943368] [0.45722264 0.         0.43477556 0.        ] [0.39306641 0.33740234 0.21386719 0.05566406
]
[0.49751183] [0.49379174]
Epoch 23: Loss: [-0.29857057] [0.4421636  0.         0.44236231 0.        ] [0.43798828 0.33544922 0.18652344 0.04003906
]
[0.49976548] [0.49406746]
Epoch 24: Loss: [-0.29847003] [0.4421636  0.         0.44236231 0.        ] [0.43847656 0.31933594 0.19775391 0.04443359
]
[0.49851762] [0.49256616]
Epoch 25: Loss: [-0.29793837] [0.48548539 0.         0.43194116 0.        ] [0.41357422 0.33837891 0.19726562 0.05078125
]
[0.50014561] [0.49297802]
Epoch 26: Loss: [-0.29835464] [0.42334983 0.         0.41691616 0.        ] [0.41650391 0.33935547 0.20068359 0.04345703
]
[0.49856246] [0.49234211]
Epoch 27: Loss: [-0.2977076] [0.48548539 0.         0.43194116 0.        ] [0.39599609 0.36181641 0.19482422 0.04736328]
[0.49926399] [0.49154261]
Epoch 28: Loss: [-0.29682537] [0.4456309  0.         0.40199548 0.        ] [0.40966797 0.34814453 0.20068359 0.04150391
]
[0.50149353] [0.49174235]
Epoch 29: Loss: [-0.29606766] [0.45722264 0.         0.43477556 0.        ] [0.41552734 0.33447266 0.20849609 0.04150391
]
[0.50271769] [0.49120771]
Epoch 30: Loss: [-0.29590153] [0.47960695 0.         0.49233286 0.        ] [0.42480469 0.35595703 0.18408203 0.03515625
]
[0.50245838] [0.49055553]
Epoch 31: Loss: [-0.29642237] [0.4421636  0.         0.44236231 0.        ] [0.43017578 0.33496094 0.18945312 0.04541016
]
[0.50078302] [0.49007574]
Epoch 32: Loss: [-0.29506462] [0.46876017 0.         0.4111282  0.        ] [0.41943359 0.34863281 0.19384766 0.03808594
]
[0.50374254] [0.489892]
Epoch 33: Loss: [-0.29525821] [0.45231102 0.         0.4109069  0.        ] [0.42285156 0.33740234 0.19580078 0.04394531
```

File - performance_test

```
 69 ]
 70 [0.50239094] [0.48897544]
 71 Epoch 34: Loss: [-0.2949515] [0.4695876  0.         0.40827971 0.        ] [0.41845703 0.34667969 0.19628906 0.03857422
    ]
 72 [0.5021059] [0.48796261]
 73 Epoch 35: Loss: [-0.29319614] [0.4858715  0.         0.47115998 0.        ] [0.43847656 0.34130859 0.17919922 0.
    04101562]
 74 [0.50615083] [0.48793186]
 75 Epoch 36: Loss: [-0.29441378] [0.45722264 0.         0.43477556 0.        ] [0.42480469 0.35595703 0.17578125 0.
    04345703]
 76 [0.50220361] [0.48679296]
 77 Epoch 37: Loss: [-0.29376369] [0.4471763  0.         0.41560002 0.        ] [0.41357422 0.36914062 0.17773438 0.
    03955078]
 78 [0.50315597] [0.48622852]
 79 Epoch 38: Loss: [-0.2932563] [0.44482384 0.         0.41277293 0.        ] [0.42382812 0.36376953 0.16943359 0.04296875
    ]
 80 [0.50397811] [0.48586672]
 81 Epoch 39: Loss: [-0.29278594] [0.45722264 0.         0.43477556 0.        ] [0.42529297 0.36865234 0.16455078 0.
    04150391]
 82 [0.5042344] [0.48501374]
 83 Epoch 40: Loss: [-0.2906624] [0.4791745 0.        0.4725308 0.       ] [0.42529297 0.37548828 0.16113281 0.03808594]
 84 [0.50867478] [0.48449249]
 85 Epoch 41: Loss: [-0.29001426] [0.48548539 0.         0.43194116 0.        ] [0.43701172 0.37109375 0.16015625 0.
    03173828]
 86 [0.50883978] [0.48311917]
 87 Epoch 42: Loss: [-0.28925511] [0.4471763  0.         0.41560002 0.        ] [0.41845703 0.37988281 0.15673828 0.
    04492188]
 88 [0.50953163] [0.48201328]
 89 Epoch 43: Loss: [-0.28857524] [0.4858715  0.         0.47115998 0.        ] [0.44482422 0.37109375 0.14501953 0.0390625
     ]
 90 [0.51065367] [0.48153068]
 91 Epoch 44: Loss: [-0.28878623] [0.4791745 0.        0.4725308 0.       ] [0.42675781 0.38525391 0.14208984 0.04589844]
 92 [0.50906462] [0.48041737]
 93 Epoch 45: Loss: [-0.28696625] [0.45231102 0.         0.4109069  0.        ] [0.4296875  0.39355469 0.14404297 0.
    03271484]
 94 [0.511743] [0.47878657]
 95 Epoch 46: Loss: [-0.28653143] [0.46876017 0.         0.4111282  0.        ] [0.44628906 0.37011719 0.14941406 0.
    03417969]
 96 [0.51303215] [0.47905418]
 97 Epoch 47: Loss: [-0.28757953] [0.4471763  0.         0.41560002 0.        ] [0.44189453 0.38085938 0.14306641 0.
    03417969]
 98 [0.50918255] [0.47764298]
 99 Epoch 48: Loss: [-0.28555609] [0.4695876  0.         0.40827971 0.        ] [0.43847656 0.38525391 0.14355469 0.
    03271484]
100 [0.51441684] [0.4781176]
101 Epoch 49: Loss: [-0.2841952] [0.48548539 0.         0.43194116 0.        ] [0.44775391 0.40136719 0.12109375 0.02978516
    ]
102 [0.51298653] [0.47337236]
103 Timer unit: 1e-07 s
104
105 Total time: 1408.91 s
106 File: C:\Users\usuario\qGAN\quantumGAN\quantum_generator.py
107 Function: train_mini_batch at line 129
108
109 Line #      Hits         Time  Per Hit   % Time  Line Contents
110 ==============================================================
111    129                                           def train_mini_batch(self):
112    130       100       9456.0     94.6      0.0          nabla_theta = np.zeros(self.parameter_values.shape)
113    131       100       1534.0     15.3      0.0          new_images = []
114    132
115    133       850      35391.0     41.6      0.0          for _, noise in self.mini_batch:
116    134      5250     122553.0     23.3      0.0              for index in range(len(self.parameter_values)):
117    135      4500     381119.0     84.7      0.0                  perturbation_vector = np.zeros(len(self.
    parameter_values))
118    136      4500     107277.0     23.8      0.0                  perturbation_vector[index] = 1
119    137
120    138      4500     434522.0     96.6      0.0                  pos_params = self.parameter_values + (np.pi / 4) *
    perturbation_vector
121    139      4500     285990.0     63.6      0.0                  neg_params = self.parameter_values - (np.pi / 4) *
    perturbation_vector
122    140
123    141      4500 6513287643.0 1447397.3     46.2                  pos_result = self.get_output(noise, params=pos_params)
124    142      4500 6481559230.0 1440346.5     46.0                  neg_result = self.get_output(noise, params=neg_params)
125    143
126    144      4500    4769147.0   1059.8      0.0                  pos_result = self.discriminator.predict(pos_result)
127    145      4500    2344015.0    520.9      0.0                  neg_result = self.discriminator.predict(neg_result)
128    146      4500    2283285.0    507.4      0.0                  gradient = self.BCE(pos_result, np.array([1.])) - self.
    BCE(neg_result, np.array([1.]))
129    147      4500     811930.0    180.4      0.0                  nabla_theta[index] += gradient
130    148       750 1082594178.0 1443458.9      7.7              new_images.append(self.get_output(noise))
131    149
132    150       700      11500.0     16.4      0.0          for index in range(len(self.parameter_values)):
133    151       600      22507.0     37.5      0.0              self.parameter_values[index] -= (self.learning_rate / self.
    mini_batch_size) * nabla_theta[index]
134    152
135    153       100      13418.0    134.2      0.0          self.mini_batch = [(datapoint[0], fake_image) for datapoint,
    fake_image in zip(self.mini_batch, new_images)]
136
137
```

File - performance_test

```
138 Process finished with exit code 0
139
```