File - performance_test

```
C:\Users\usuario\Anaconda3\envs\tfq\python.exe C:/Users/usuario/qGAN/quantumGAN/performance_test.py
[0.33928449 0.50965994 0.24682033 0.03262591 0.61093277 0.92918652]
Epoch 0: Loss: [-0.30578982] [0.49557007 0.         0.44217426 0.        ] [0.35742188 0.34667969 0.19238281 0.10351562]
[0.37647422] [0.3503415]
Epoch 1: Loss: [-0.29081268] [0.4993665  0.         0.46070319 0.        ] [0.33886719 0.36279297 0.1875     0.11083984]
[0.45627328] [0.42568573]
Epoch 2: Loss: [-0.28585074] [0.44600582 0.         0.42140997 0.        ] [0.30224609 0.39941406 0.18847656 0.10986328]
[0.4955307] [0.45896176]
Epoch 3: Loss: [-0.28316383] [0.44600582 0.         0.42140997 0.        ] [0.33007812 0.38330078 0.18261719 0.10400391]
[0.51199804] [0.46984356]
Epoch 4: Loss: [-0.28142478] [0.49378013 0.         0.42531685 0.        ] [0.30175781 0.3984375  0.18945312 0.11035156]
[0.52042547] [0.47423473]
Timer unit: 1e-07 s

Total time: 151.549 s
File: C:\Users\usuario\qGAN\quantumGAN\quantum_generator.py
Function: train_mini_batch at line 129

Line #      Hits         Time  Per Hit   % Time  Line Contents
==============================================================
   129                                           def train_mini_batch(self):
   130        10        940.0     94.0      0.0          nabla_theta = np.zeros(self.parameter_values.shape)
   131        10        136.0     13.6      0.0          new_images = []
   132
   133        85       4066.0     47.8      0.0          for _, noise in self.mini_batch:
   134       525      13884.0     26.4      0.0              for index in range(len(self.parameter_values)):
   135       450      39475.0     87.7      0.0                  perturbation_vector = np.zeros(len(self.parameter_values
   ))
   136       450      12676.0     28.2      0.0                  perturbation_vector[index] = 1
   137
   138       450      60492.0    134.4      0.0                  pos_params = self.parameter_values + (np.pi / 4) *
   perturbation_vector
   139       450      32910.0     73.1      0.0                  neg_params = self.parameter_values - (np.pi / 4) *
   perturbation_vector
   140
   141       450  696649511.0 1548110.0     46.0                  pos_result = self.get_output(noise, params=pos_params)
   142       450  699409240.0 1554242.8     46.2                  neg_result = self.get_output(noise, params=neg_params)
   143
   144       450     510604.0   1134.7      0.0                  pos_result = self.discriminator.predict(pos_result)
   145       450     239800.0    532.9      0.0                  neg_result = self.discriminator.predict(neg_result)
   146       450     243774.0    541.7      0.0                  gradient = self.BCE(pos_result, np.array([1.])) - self.
   BCE(neg_result, np.array([1.]))
   147       450     114983.0    255.5      0.0                  nabla_theta[index] += gradient
   148        75  118150606.0 1575341.4      7.8              new_images.append(self.get_output(noise))
   149
   150        70       1687.0     24.1      0.0          for index in range(len(self.parameter_values)):
   151        60       2739.0     45.6      0.0              self.parameter_values[index] -= (self.learning_rate / self.
   mini_batch_size) * nabla_theta[index]
   152
   153        10       1939.0    193.9      0.0          self.mini_batch = [(datapoint[0], fake_image) for datapoint,
   fake_image in zip(self.mini_batch, new_images)]


Process finished with exit code 0

```