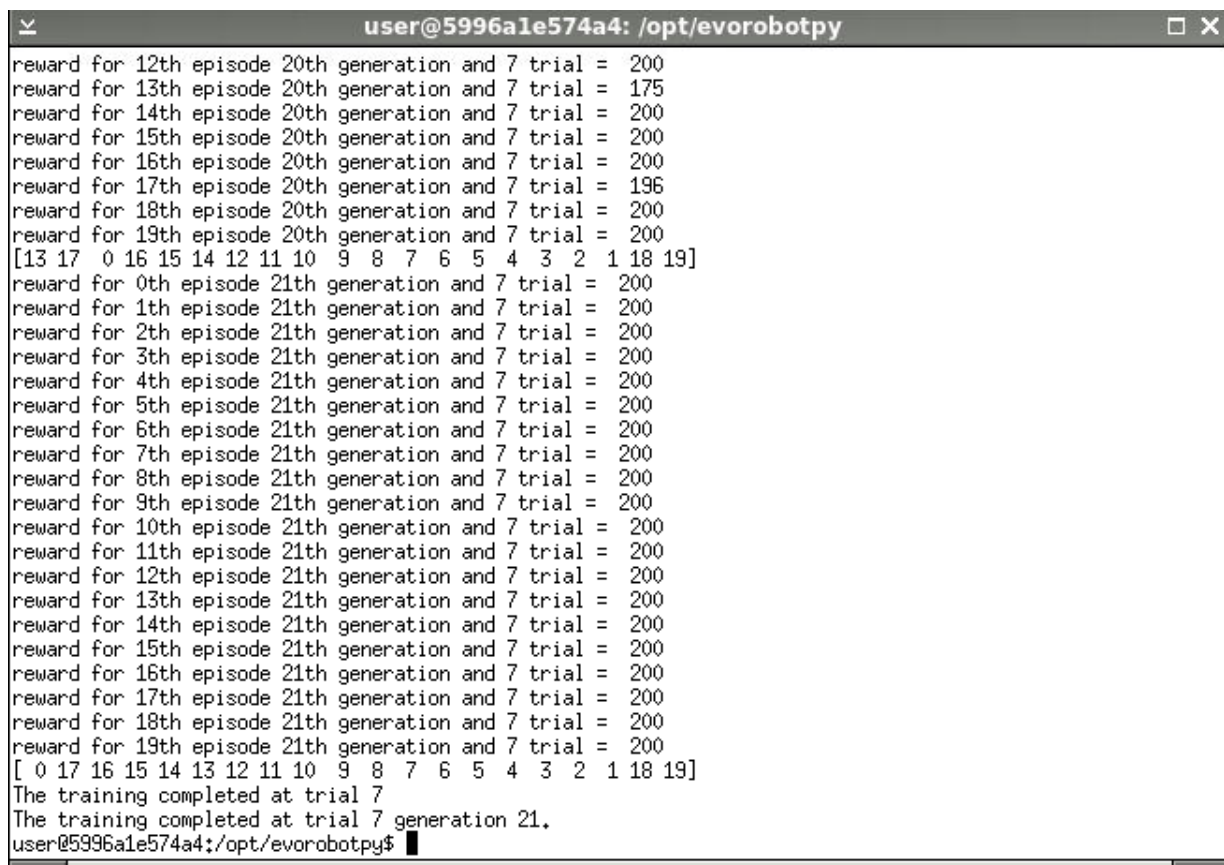


Exercise 2B

a) Training the Neural Network

In this exercise it is required to fine tune the cart-pole system using neural network to stabilize the system. To accomplish this task, a matrix of random parameters (weights and biases of neural network) is initialed first. Each row of this matrix represent one of the settings for neural network. First of all, each row is literately accessed and action is calculated for each step in episode. At the end of each episode, the parameter matrix is sorted depending on the reward of each episode. Next, the half that has worst performance is replaced with best parameters. This process is repeated for 30 generations, if the performance is maximum for each episode the program is terminated. Some times due to randomness of the initial parameters the solution does not converge. So, there is another loop on the top of generation. I called it trials. The training code is given in this directory that has the above mentioned routine. The figure below shows the successful training.



```
user@5996a1e574a4: /opt/evorobotpy
reward for 12th episode 20th generation and 7 trial = 200
reward for 13th episode 20th generation and 7 trial = 175
reward for 14th episode 20th generation and 7 trial = 200
reward for 15th episode 20th generation and 7 trial = 200
reward for 16th episode 20th generation and 7 trial = 200
reward for 17th episode 20th generation and 7 trial = 196
reward for 18th episode 20th generation and 7 trial = 200
reward for 19th episode 20th generation and 7 trial = 200
[13 17 0 16 15 14 12 11 10 9 8 7 6 5 4 3 2 1 18 19]
reward for 0th episode 21th generation and 7 trial = 200
reward for 1th episode 21th generation and 7 trial = 200
reward for 2th episode 21th generation and 7 trial = 200
reward for 3th episode 21th generation and 7 trial = 200
reward for 4th episode 21th generation and 7 trial = 200
reward for 5th episode 21th generation and 7 trial = 200
reward for 6th episode 21th generation and 7 trial = 200
reward for 7th episode 21th generation and 7 trial = 200
reward for 8th episode 21th generation and 7 trial = 200
reward for 9th episode 21th generation and 7 trial = 200
reward for 10th episode 21th generation and 7 trial = 200
reward for 11th episode 21th generation and 7 trial = 200
reward for 12th episode 21th generation and 7 trial = 200
reward for 13th episode 21th generation and 7 trial = 200
reward for 14th episode 21th generation and 7 trial = 200
reward for 15th episode 21th generation and 7 trial = 200
reward for 16th episode 21th generation and 7 trial = 200
reward for 17th episode 21th generation and 7 trial = 200
reward for 18th episode 21th generation and 7 trial = 200
reward for 19th episode 21th generation and 7 trial = 200
[0 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 18 19]
The training completed at trial 7
The training completed at trial 7 generation 21.
user@5996a1e574a4:/opt/evorobotpy$
```

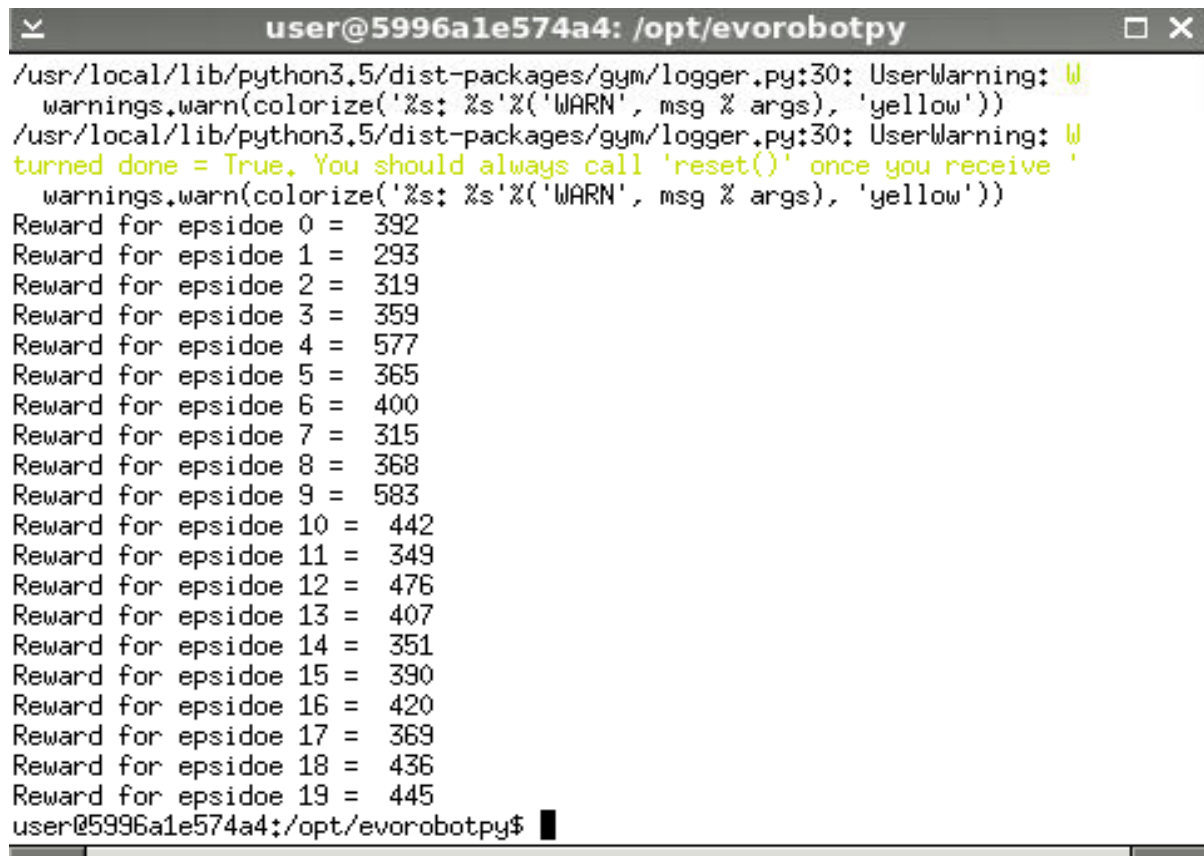
Figure. 1

It is important to note that there are 200 steps in each episode and there are 20 episodes in each generations. Overall there are 30 generations to make sure

that the algorithm converges. On top of generations loop there is a trial loop. The basic purpose of this loop is to only reset the neural network to a random initial value. The training completed after 7th trial and 21st generation.

b) Testing the Neural Network

Now, the second step is to simulate the environment with the trained model parameters. For this purpose the cart-pole is set to run for 600 steps, although it was trained for 200 steps. But the results were pretty good. In some episodes it was able to stabilize even for 600 steps. It is shown in figure 2.

A terminal window titled 'user@5996a1e574a4: /opt/evorobotpy' displays the output of a simulation. It shows a series of 'Reward for epsidoe' (sic) values for episodes 0 through 19. The rewards fluctuate, with episode 9 showing the highest value at 583. There are also two yellow warning messages from the gym logger about not calling 'reset()' after an episode is done.

```
user@5996a1e574a4: /opt/evorobotpy
/usr/local/lib/python3.5/dist-packages/gym/logger.py:30: UserWarning: W
warnings.warn(colorize('%s: %s'%( 'WARN', msg % args), 'yellow'))
/usr/local/lib/python3.5/dist-packages/gym/logger.py:30: UserWarning: W
turned done = True. You should always call 'reset()' once you receive '
warnings.warn(colorize('%s: %s'%( 'WARN', msg % args), 'yellow'))
Reward for epsidoe 0 = 392
Reward for epsidoe 1 = 293
Reward for epsidoe 2 = 319
Reward for epsidoe 3 = 359
Reward for epsidoe 4 = 577
Reward for epsidoe 5 = 365
Reward for epsidoe 6 = 400
Reward for epsidoe 7 = 315
Reward for epsidoe 8 = 368
Reward for epsidoe 9 = 583
Reward for epsidoe 10 = 442
Reward for epsidoe 11 = 349
Reward for epsidoe 12 = 476
Reward for epsidoe 13 = 407
Reward for epsidoe 14 = 351
Reward for epsidoe 15 = 390
Reward for epsidoe 16 = 420
Reward for epsidoe 17 = 369
Reward for epsidoe 18 = 436
Reward for epsidoe 19 = 445
user@5996a1e574a4:/opt/evorobotpy$
```

Figure. 2

c) Discussion

It was observed that the evolutionary algorithm used in this exercise is pretty good and performed very well. But it is also observed that the success of the event depends very much on the initial parameter matrix. This is the reason that some times the algorithm converges to the maximum reward value very quickly.