Reinforcement Learning - Exercise 1

Muhammad Kamal Memon
600442

Exercise:

The default value of maximum episode length was 200, after we change it to 500 the agent still learn to balance the pole. After the change, on average after multiple runs it does take longer but its very minimal change from the default value of 200. As on some runs it takes longer but on some it converges fast.

1.

```
def new_reward(state):
    return min(4, np.abs(1/state[0]))
```

This reward functions converges quickly (327 Episodes). Since it starts at zero the reward is already high in the beginning and it tends to keep there. We are using the absolute of inverse position which means that when the value is zero the reward is highest and it goes down as the position moves away from zero. However we need to cap it to 4 at most or else the reward go too high and it doesn't remain stable.

2.

```
def new_reward(state):
    if state[0] == args.x_0:
        return 4
    else:
        return min(4, 1/np.abs(state[0]-args.x_0))
```

This reward function follows the same logic as earlier but it takes longer to converge. This model also took multiple attempts to train. The arbitrary point taken was 1.5 and the test rendering shows that it goes there. Also we have observed the furthest the arbitrary point is from the center the quicker agent tries to go there and the probability of it dying also rise.

3.

```
def new_reward(state):
    return 2+state[1]**2
```

For this model we have to reward the agent more if it keeps the velocity high. The above function penalise the reward if the velocity is closer to zero and rewards high if its away from zero. However we observed that the faster the agent is the harder it is to balance the pole hence high velocity is not very stable.

Along with this report the code and model files are attached.

References:

The exercise involved discussion with following students:

- Gadidjah Ogmundsdottir
- Tommi Vehvilainen