

10-703 Deep Reinforcement Learning and Control

Assignment 3

Fall 2018

ielshar
mharding

October 26, 2018

Problem 1: Reinforce

1. Describe your implementation:

- Neural Network Architecture: same as given in JSON file.
- Learning rate: 0.001
- Discount factor γ : 1

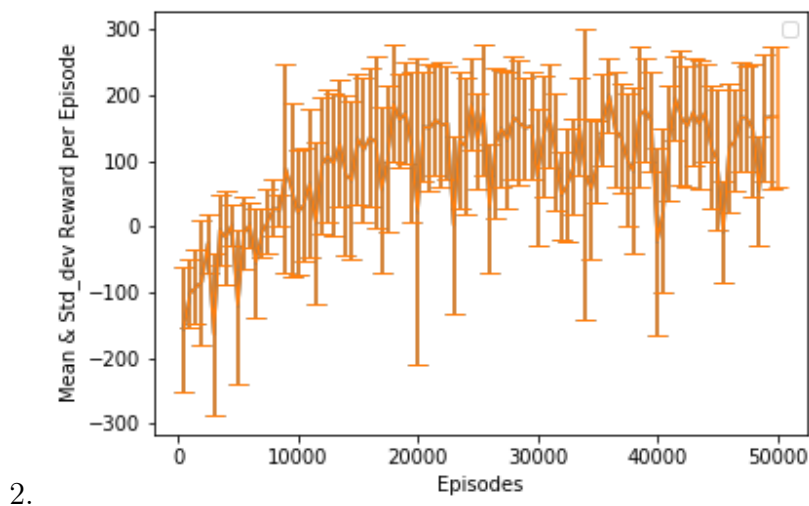


Figure 1: Reinforce Algorithm: learning curve - Every $k=500$ episodes the current policy is tested on 100 episodes. The plot shows the mean and standard deviation of each of these tests.

Figure 1 above shows that our agent was able to achieve a mean reward of 200 or slightly more or less at several points (after 36000, 41500 episodes) through out the training. Since no baseline was used in our Reinforce algorithm one can see that the test results have a high variance. This is somehow expected though as the total return at the end of episode varies highly from one episode to another.

Problem 2: Advantage-Actor Critic

1. Implementation details:

Table 1: A2C Implementation

A2C		
N	Settings	value
1	Actor NN Architecture	Same as given in JSON file
	Critic NN Architecture	MLP with 3 layers each with 30 hidden units and relu activation
	Actor learning rate	0.001
	Critic learning rate	0.001
20	Actor NN Architecture	Same as given in JSON file
	Critic NN Architecture	MLP with 3 layers each with 20 hidden units and relu activation
	Actor learning rate	0.001
	Critic learning rate	0.001
50	Actor NN Architecture	Same as given in JSON file
	Critic NN Architecture	MLP with 3 layers each with 20 hidden units and relu activation
	Actor learning rate	0.001
	Critic learning rate	0.001
100	Actor NN Architecture	Same as given in JSON file
	Critic NN Architecture	MLP with 3 layers each with 20 hidden units and relu activation
	Actor learning rate	0.0008
	Critic learning rate	0.001

* discount factor was set to 1 in all implementations.

2. Plots:

- N=1

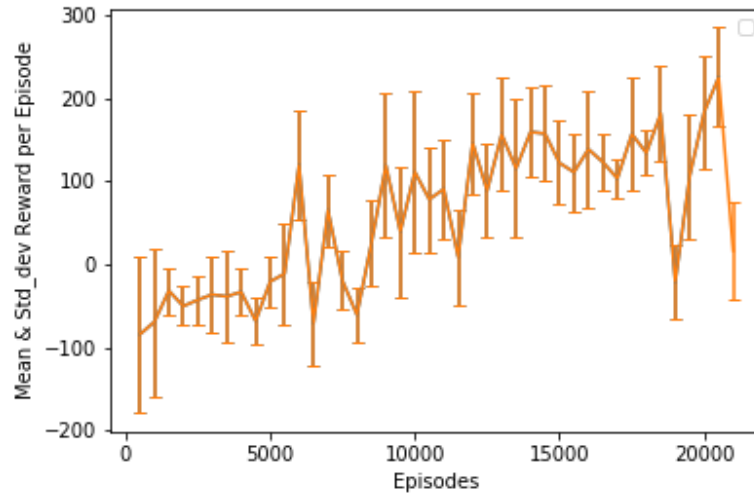


Figure 2: A2C Algorithm N=1: learning curve - Every $k=500$ episodes the current policy is tested on 100 episodes. The plot shows the mean and standard deviation of each of this tests.

- N=20

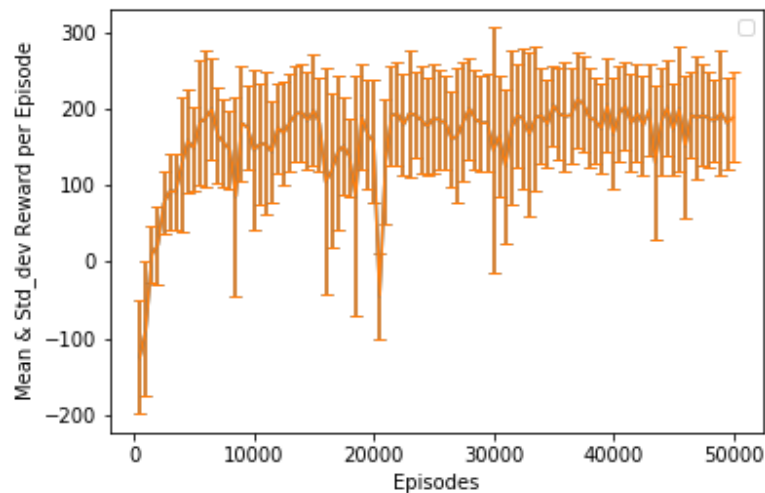


Figure 3: A2C Algorithm N=20: learning curve - Every $k=500$ episodes the current policy is tested on 100 episodes. The plot shows the mean and standard deviation of each of this tests.

- N=50

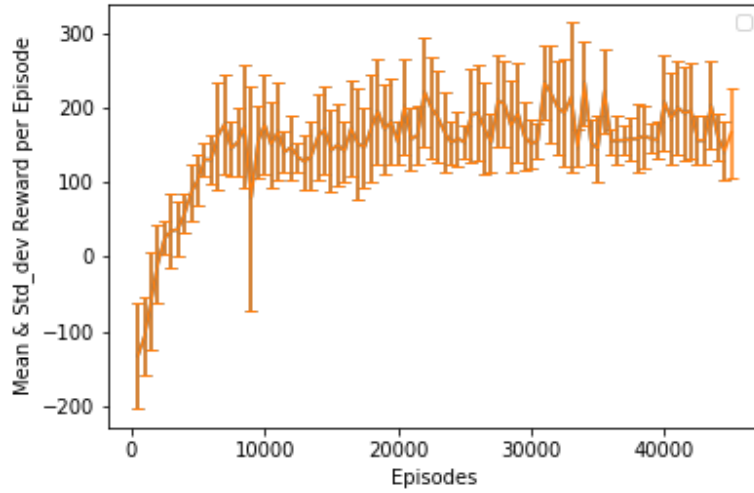


Figure 4: A2C Algorithm N=50: learning curve - Every $k=500$ episodes the current policy is tested on 100 episodes. The plot shows the mean and standard deviation of each of this tests.

- N=100 actor $lr=0.0008$

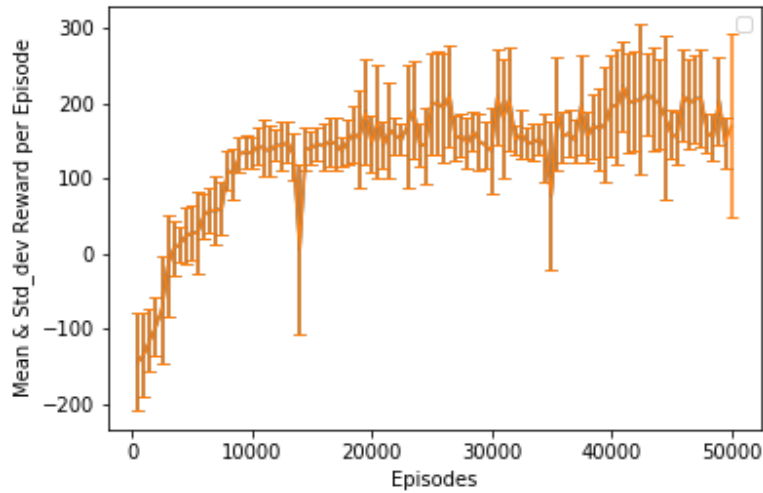


Figure 5: A2C Algorithm N=100: learning curve - Every $k=500$ episodes the current policy is tested on 100 episodes. The plot shows the mean and standard deviation of each of this tests.

- N=100 actor $lr=0.001$

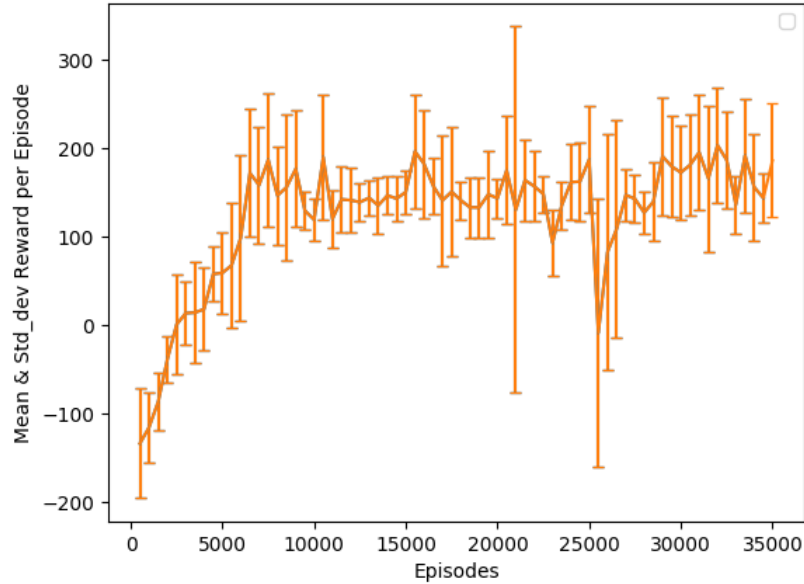


Figure 6: A2C Algorithm $N=100$: learning curve - Every $k=500$ episodes the current policy is tested on 100 episodes. The plot shows the mean and standard deviation of each of these tests.

3. Reinforce and A2C comparisons:

Compared to Reinforce, n -steps A2Cs have much lower variance and accelerated learning expect for the case when $n=1$. This is mainly due the added effect of the critic which bootstraps unlike the Reinforce algorithm which uses the MC return that has high variance. The case of A2C with one-step look critic introduce much more bias than $n=20, 50$ and 100 A2Cs that is why it's much harder to learn. It seems that A2C algorithm with $n=20, 50$ and 100 learn faster than all the others. Even though we were able to achieve a better performance for $n=100$ with an actor learning rate of 0.0008 as seen in Figure 5 above, we run A2C $n=100$ again for 35000 episodes with an actor $lr=0.001$ (Figure 6) which is the same as the one used for A2C $n=20$ and $n=50$. This is done to make sure the comparison regarding speed of learning is fair enough since now the only difference between them is n , the number of steps. (Note that we fix the random seed in all of the implementations).

A2C $n=20$ was the fastest to reach a mean reward of 200 or greater. It achieved that after only 6500 episodes. A2C $n=50$ (50-steps return) comes second with 20500 episodes. Third comes A2C $n=100$ with 32000 episodes. Fourth comes Reinforce with 36000 episodes. A2C $n=1$ was not able to achieve 200 with the same neural architecture as the other algorithms. However using a NN architecture that include 3 layers each with 30 units this time A2C $n=1$ achieved a reward of 225 after 20500 episodes similar to A2C $n=50$. Comparing figures 3, 4 and 6, it can be seen that A2C $n=20$ has the steepest rate of learning then comes $n=50$ and finally $n=100$. We believe this is the case since 20-step and 50-step A2Cs have the best balance between bootstrapping using

the value function and using the full MC return.