Walking Humanoid

Reinforcement Learning Project

Master's Degree in Artificial Intelligence and Robotics

Eugenio Bugli (1934824)

Academic Year 2023/2024

SAPIENZA
UNIVERSITÀ DI ROMA

This Project's aim is to use Action Critic Algorithm (A2C) applied to the Humanoid Mujoco environment of Gymnasium.

After the training the Humanoid should be able to walk on his own and balance itself during the movements.

Table of Contents

Environment

- The Humanoid environment is part of the Mujoco set from Gymnasium.
- The goal of this Environment is to walk forward as fast as possible without falling over
- The 3d bipedal Robot is designed to simulate a human.
- The figure is composed by a torso with legs and arms, each of them composed respectively by three and two body parts.

Action Space

1 Environment

Description of the Action Space:

- An Action represent the torques applied at the hinge points.
- We have a total of 17 actions.
- All the actions assume continuos values from a minimum of -0.4 to a maximum of 0.4.
- The unit used is torque [N m]

Observation Space

1 Environment

Description of the Observation Space:

- Observation consists of positional values of different body parts of the Humanoid, followed by the velocities of those parts.
- The Observation is composed of 378 float64 elements (376 if we do not include X and Y coordinates of the Torso.
- The elements inside the observation do not have strict limitations on the values like happened for the actions.

Observation Structure

1 Environment

Each Observation has a structure like the following one:

Observation (378,1)

- 3 positions [m]
- 21 angles [rad]
- 3 linear velocities [m/s]
- 20 angular velocities [m/s]
- cinert: 140 elements (mass and inertia)
- cvel: 84 elements (center of mass based velocity)
- qfrc_actuator: 23 elements (actuator force)
- cfrc_ext: 84 elements (center of mass external force on the body)

Rewards

1 Environment

The reward consist of four different parts:

- **Healthy_Reward** : fixed value obtained at every timestep in which the Humanoid is considered healthy.
- **Forward_Reward** : reward given when the Humanoid moves forward.
- **Ctrl_Cost** : penalization if the Humanoid has too large control force.
- **Contact_Cost** : penalization if the external forces on the Humanoid are too large.

I have tried to increase the reward every time the humanoid was mantaining its position (check on Y-axis position and velocity) and moving forward (check on X-axis position and velocity).

Termination of the Episode

The Humanoid is considered unhealthy if the z-coordinate of the torso is no longer inside the closed interval defined by the **healthy_z_range**, which by default is [1.0,2.0].

- When **terminate_when_unhealthy** is *True*, the episode ends by Truncation (the episode reaches 1000 timesteps) or Termination (Humanoid is unhealthy).
- Otherwise, the episode terminates only after 1000 timesteps.

Table of Contents

2 Actor Critic Algorithm

Actor Implementation

2 Actor Critic Algorithm

Network structure:

- Fully Connected Layer 1 (378, 32) + ReLU
- Fully Connected Layer 2 (32, 32) + ReLU
- Fully Connected Layer 3.1 (32, 17) **Mean**
- Fully Connected Layer 3.2 (32, 17) **Log_Std**

Notes on Actor Implementation

2 Actor Critic Algorithm

I have made some adjustments to obtain the actions due to some problems that I have encountered:

- Use of **Reparameterization Trick**.
- Clipping the **Log_Std** values to a closed interval of [-20, 2] due to **NaN** errors.
- **Ornstein-Uhlenbeck Process Action Noise** to increase exploration.

Critic Implementation

2 Actor Critic Algorithm

Network structure:

- Fully Connected Layer 1 (378, 32) + ReLU
- Fully Connected Layer 2 (32, 32) + ReLU
- Fully Connected Layer 3 (32, 1)

Differences from the Paper

2 Actor Critic Algorithm

Due to time complexity I have not applied the gradient update to the Actor suggested by the paper:

- Natural Gradient

$$\tilde{\nabla} L(\theta) = \mathbf{F}(\theta)^{-1} \nabla L(\theta)$$

where $\mathbf{F}(\theta)$ is the Fisher Information Matrix (known as "All Actions" matrix):

$$\mathbf{F}(\theta) = \pi_\theta(a|s) \cdot \nabla log(\pi_\theta(a|s)) \cdot \nabla log(\pi_\theta(a|s))^T$$

Loss Function

2 Actor Critic Algorithm

Loss used:

- Actor:

$$L_A(\theta) = -\mathbf{log}(\pi_\theta(\mathbf{a}|\mathbf{s}))^{\mathbf{T}}\mathbf{A}(\mathbf{s})$$

- Critic:

$$L_C(\theta) = \mathbf{MSE}(\mathbf{A}, \mathbf{V})$$

with A as the Advantage function approximated with TD error:

$$\mathbf{A}(\mathbf{s_t}) = r_t + \gamma V(s_{t+1}) - V(s_t)$$

Table of Contents

3 Performances

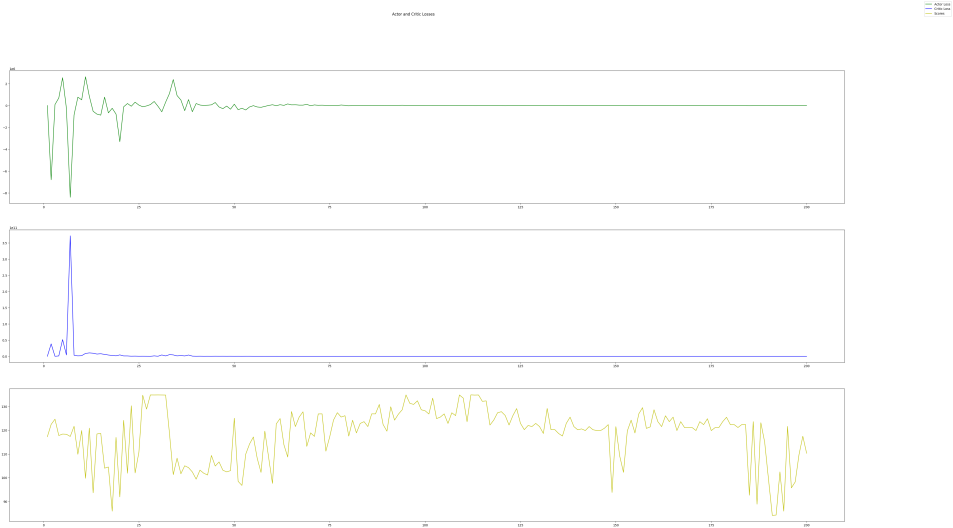
Losses

3 Performances

- Both Actor and Critic Losses are decreasing
- Scores are not increasing so much (Stop increase after some thousand iterations)
- Training done for 10k epochs but displayed only 200 to fit the slide.
- Next image is a slice of the losses and scores behaviour .

Final Model

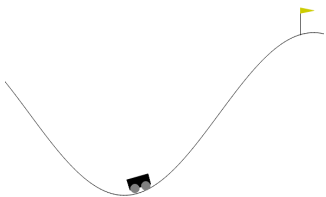
3 Performances



Mountain Car Test

3 Performances

I have applied the same Algorithm to the Continuos Mountain Car Environment. In this case the agent is actually able to solve the requested task successfully. (Note: open the videos with VS Code)



Bipedal Walker

3 Performances

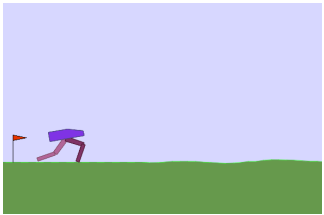
I tried also on the Bipedal Walker Environment, in which we have a sort of simplified humanoid with the same task in 2D.

- Observation Space: velocity components and LIDAR measurements ((24,1) shape)
- Action Space: motor speed values for the 4 joints ((4,1) shape)
- Reward: given for moving forward, -100 if the agent falls.

Bipedal Walker Test

3 Performances

Due to the complexity I was not able to actually made it walk. My agent get stuck in a sort of *split gymnastic position*, without falling and actually moving forward. I have an increase of the reward but my agent is not actually achieving the task. In other situations it falls on its knees or forward. (Note: open the videos with VS Code)



Adjustments Made and Problem Encountered

3 Performances

- Different ways on initialize the Noise
- Use some magnitude values to decrease Noise due to the clamping for the actions.
- Ignorance about the observations.
- High sensibility to the choice of hyperparameters and optimizers.

Possible Solutions

3 Performances

- Improve Exploration (Entropy penalty inside the Actor's Loss, ...)
- Increase the reward according specific information from the observation (coordinates, velocity, ...)
- Try other variants of the Actor Critic algorithm (A3C, Natural Actor Critic, SAC, ...)
- Gradient clipping

Walking Humanoid