Capstone Proposal
Arise and walk

Guitard Alan

April 5, 2018

1 Domain Background

For a human, the ability to walk is one of the most basic knowledge he learnt during his childhood. Since we are trying with AI to, when we can't do better, mimic the ability of human being, the idea of teaching a robot to walk comes naturally in our mind.

One of the first attempt to do that was to develop an algorithm by trying to understand the physics formula of that movement. In 1993, Yamaguchi et al. [5] tried to make a little robot by trunk motion using the ZMP (Zero Moment Point) and the three axis (pitch, yaw and roll)[5]. In 1999, Nagasaka et al. added the OGM (Optimal Gradient Method) to that approach to "optimizes the horizontal motion of a trunk to reduce the deviation of the calculated ZMP from its reference." [3] and design the KHR-2 robot. The result was good but the gait of the robot wasnot very natural because it is very difficult to take all the factors in account for making the robot walk with a human gait.

Nowadays, Boston Dynamics made a huge advances in humanoÃŕd robotic by designing Atlas, who was able to do a perfect backflip, or walk on a non-flat ground. In terms of simulation, Geijtenbeek et al. desgined a muscle based avatar with two legs which can take a lot of shape. According to that shape and environment (e.g. the gravity), they were able to teach the creature to stand and walk, and it learned the proper gait (one creature with small legs figured out itself it is easier to jump).[2] All that studies could be used in many areas. Healthcare institute could design better articial arm or leg, or could give more efficient companion robots to their patient. That last one could be use in all fields of life, like a buttler. In a more ludic ways, we will be able to design a more realistic gait for our 3D avatar, even for non-player characters.

2 Problem Statement

The problem I want to solve is then to teach a 3D avatar to walk. That kind of problem is solved with reinforcement learning and this is the solution I will try to solve. For a human, walking is a simple task but we did learn after many trials, step by step, with the help of our hand at the beggining and then stand and walk.

To teach that to a 3D avatar, we need to define algorithm which allow not only the avatar to walk but to walk with a human gait to avoid too much movement for a single step, or avoid doing a false movement and fail down after three steps because of that false movement. At each step, without thinking about it, humans take care of the two/three steps we will do in the future. We have to design our model to be able to do that and optimally with the same gait as human, that we can consider as the optimal gait for our shape.

3 Datasets and Inputs

I will use OpenAI with the Gym python library[1], load the Roboschool environment (because the default Mujoco is not free) and use deep reinforcement learning to make the robot stands and walks.

3.1 Action space

The action space is a vector of 17 float values in the range [-1, 1]. Each value corresponds to the joints of the avatar by this order XML:

- abdomen_y
- abdomen_z
- abdomen_x
- right_hip_x
- right_hip_z
- right_hip_y
- right_knee
- left_hip_x
- left_hip_z
- left_hip_y
- left_knee
- right_shoulder1
- right_shoulder2
- right_elbow
- left_shoulder1
- left_shoulder2
- left_elbow

At each step, these values are applied to all the joints of the body by the code

```
for n,j in enumerate(self.ordered_joints):
    j.set_motor_torque( self.power*j.power_coef \
                        *float(np.clip(a[n], -1, +1)) )
```

in the `apply_action` function in the class which extends the `gym.Env` class (`RoboschoolMujocoXmlEnv`) to set the torque value into the respective motor.

3.2 Observation space

The state space (or observation space) is a vector of 44 float values in the range [-5, 5] (Roboschool clip the vector with numpy before returning it in the `step` function). That vector is a concatenation of three subvectors:

- **more**: It is a vector of 8 values defined as follows:
  - The distance between the last position of the body and the current one.
  - The sinus of the angle to the target.
  - The cosinus of the angle to the target.
  - The three next values is the X, Y and Z values of the matrix multiplication between
    - $$\begin{pmatrix} \cos(-yaw) & -\sin(-yaw) & 0 \\ \sin(-yaw) & \cos(yaw) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$
    - The speed vector of the body.
  - The roll value of the body
  - The pitch value of the body
- **j**: This is the current relative position of the joint described earlier and their current speed. The position is in the even position, and the speed in the odds (34 values).
- **feet_contact**: Boolean values, 0 or 1, for left and right feet, indicating if the respective feet is touching the ground or not.

3.3 Reward

The reward is a sum of 5 computed values:

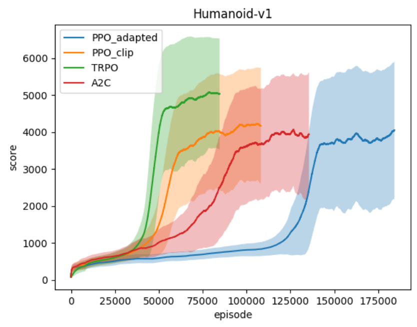
- **alive**: -1 or +1 wether is on the ground or not
- **progress**: potential minus the old potential. The potential is defined by the speed multiplied by the distance to target point, to the negative.
- **electricity_cost**: The amount of energy needed for the last action
- **joints_at_limit_cost**: The amount of collision between joints of body during the last action
- **feet_collsion_cost**: The amount of feet collision taken during the last action

4 Solution Statement

In my research, I figured out two I can't decide which one would be the best to train my avatat to walk: A2C (Advantage Actor Critic) and Deep Q-Learning. For the sake of my learning, I am really interested in the A2C algorithm since it is the one who made great progress in Reinforcment Learning (I am thinking about AlphaGO[4]). But I did understand that this algorithm will suit more on that problem since the walk of the robot is a continuous learning, and not an episodic, for which Deep Q-Learning is more suitable.

5 Benchmark Model

The random action model makes the avatar lying down on the ground convulsing because it doesn't know how to stand up and it is just moving its joints randomly.



In the above benchmark, we can see that Advantage Actor Critc algorithm is able to converge at around 100000 episodes. That tells me that I have to be patient on my training and have good metrics to tell if my model will converge or not.

6 Evaluation Metrics

7 Project Design

References

[1] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[2] Thomas Geijtenbeek, Michiel van de Panne, and A. Frank van der Stappen. Flexible muscle-based locomotion for bipedal creatures. *ACM Transactions on Graphics*, 32(6), 2013.

[3] Ken'ichiro Nagasaka, Hirochika Inoue, and Masayuki Inaba. Dynamic walking pattern generation for a humanoid robot based on optimal gradient method. *1999 IEEE International Conference*, 6:908 – 913 vol.6, 02 1999.

[4] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354–, October 2017. URL `http://dx.doi.org/10.1038/nature24270`.

[5] J.-I Yamaguchi, Atsuo TAKANISHI, and Ichiro KATO. Development of a biped walking robot compensating for three-axis moment by trunk motion. *1993 IEEE/RSJ International Conference*, 11:561 – 566 vol.1, 08 1993.

Glossary

Zero Moment Point A definition. 1