ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

Human feedback in continuous actor-critic reinforcement learning

Cristian Millán[1] Bruno Fernandes [1] Francisco Cruz [2]

1- Universidade de Pernambuco - Escola Politécnica de Pernambuco
Rua Benfica 455, Recife/PE - Brasil

2- Universidad Central de Chile - Escuela de Computación e Informática
Santa Isabel 1186, Santiago - Chile

Abstract. Reinforcement learning is utilized in contexts where an agent tries to learn from the environment. Using continuous actions, the performance may be improved in comparison to using discrete actions, however, this leads to excessive time to find a proper policy. In this work, we focus on including human feedback in reinforcement learning for a continuous action space. We unify the policy and the feedback to favor actions of low probability density. Furthermore, we compare the performance of the feedback for the continuous actor-critic algorithm and test our experiments in the cart-pole balancing task. The obtained results show that the proposed approach increases the accumulated reward in comparison to the autonomous learning method.

1 Introduction

Reinforcement learning (RL) is a learning approach inspired by behavioral psychology where an agent, able to interact with its environment, tries to find an optimal policy to perform a specific task.[1]. In many applications, it is common to discretize the actions and states space to facilitate the learning, but some regions of space may be more important than others, and such information may be lost [2]. Besides, to leave the agent learning a task by itself is impractical and involves problems to find the proper policy. However, the agent can be guided by expert or non-expert trainers, external to the environment, where they provide feedback based on the performance of the agent during the task [3].

This work focuses on including human feedback in RL in continuous actions and state spaces [4]. The developed strategy based on the policy-shaping method [5] is used in the continuous actor-critic algorithm [1], where the feedback provides information about the policy. To evaluate the performance of this methodology, we apply it to the task of cart-pole balancing. The results show that the reward received by the agent is higher in the presence of human feedback.

Our paper is organized as follows: first, we introduce the theoretical framework of continuous RL and actor-critic algorithm. Next, we define our approach to include human feedback in RL approach and a description of how the feedback favors actions. We describe the experimental setup to apply our methodology. Moreover, we show and compare the performance of human feedback in continuous actor-critic RL. Finally, we present our conclusions and describe future works.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

2 Continuous reinforcement learning

Let $X \subseteq \mathbb{R}^p$ and $U \subseteq \mathbb{R}^q$ $(p, q \in \mathbb{N})$ be a set of states and a set of actions respectively. Let $x_t \in X$ be a state and $u_t \in U$ an action of the agent at time t, then the states x_t transitions to the state x_{t+1} through the function $T : X \times U \longrightarrow X$ called transition function. After state x_t is reached, the action receives a scalar reward r_{t+1} that is determined by the *reward function* $\rho : X \times U \to \mathbb{R}$, this function evaluates the immediate performance of the action selected. The policy is a probability density distribution function $\pi : X \times U \longrightarrow [0, \infty)$, therefore, the action u_t is drawn randomly from π given the current state x_t. The goal of the RL agent is to find the policy π that maximizes the expected value of reward it receives in the long run.

Thus, the agent tries to select actions in U such that the discounted sum of future rewards it receives is maximized. For a given initial state x, the value of state x, over all the actions, is the expected return of the discounted sum starting in the state x under the policy π. Formally we can define this by:

$$V^{\pi}(x) = E_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| x_t = x\right],$$

where γ is a parameter, $0 < \gamma < 1$, called the discount rate. The previous equation is the state-value function for policy π. It is clear that the state-value function and the policy are real valued function, but in tasks with small or finite state sets, these functions can be approximated using tables with one entry for each state or state-action pair [1]. The state-value function can be expressed regarding the value of the next state, this relation is called the Bellman equation and is defined formally by:

$$V^{\pi}(x) = E_{\pi}\left[r_{t+1} + \gamma V^{\pi}(x')\right],$$

where $x' \in X$ is the state at time $t + 1$ based on the current state x.

2.1 Actor-critic algorithm

The actor-critic algorithm [6] is characterized by having a separate memory structure to represent the policy independent of the value function. The policy, represented by the actor, is used to select actions and the value function criticizes the drawing actions. In this work, the actor-critic based on temporal difference is used as the baseline algorithm to compare our methodology. Let $V_{\theta}(x_t)$ be the approximated value function parametrized by θ and $\pi_{\vartheta}(u_t|x_t)$ be the approximate policy parametrized by ϑ. The critic takes the form of the temporal difference error, defined by:

$$\delta_t = r_{t+1} + \gamma V_{\theta}(x_{t+1}) - V_{\theta}(x_t)$$

Using gradient descent methods, the update rule to the critic is:

$$\theta_{t+1} = \theta_t + \alpha_c \delta_t \nabla_{\theta_t} V_{\theta}(x_t)$$

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

were $\alpha_c \in (0, 1]$ is the learning rate of the critic and $\nabla_{\theta_t} V_\theta(x_t)$ is the gradient. At each time step, the actions are selected from a probability density function $\pi_\vartheta(u_t|x_t)$. The update rule to the actor using gradient descent is:

$$\vartheta_{t+1} = \vartheta_t + \alpha_a \delta_t \frac{\nabla_{\vartheta_t} \pi_\vartheta(u_t|x_t)}{\pi_\vartheta(u_t|x_t)}$$

where $\alpha_a \in (0, 1]$ is the learning rate of the actor and the vector $\frac{\nabla_{\vartheta_t} \pi_\vartheta(u_t|x_t)}{\pi_\vartheta(u_t|x_t)}$ is denominated compatible features [7].

3 Modified policy with human feedback

Let $J_t \in \mathbb{J}$ be feedback provided by an external trainer at time t, and $\mathbb{J}$ the set of all possibles instructions that allow reaching a goal. In some iteration steps, the trainer may not provide feedback. Thus, the likelihood of receiving feedback [5] has probability $0 < \mathcal{L} < 1$. We suppose that any action u_t is drawn given that an external trainer provides feedback J_t in the state x_t. Let $\pi(u|x_t, J_t)$ be the probability density function of the actions after taking into account the feedback J_t and the state x_t, at time t. From properties of conditional probability, this policy can be expressed by:

$$\begin{aligned} \pi(u|x, J) &= \frac{P_J(J|u, x)P(u, x)}{P(x, J)} \\ &= \frac{P_J(J|u, x)P(u|x)P_x(x)}{P_J(J|x)P_x(x)} \\ &= \frac{P_J(J|u, x)}{P_J(J|x)}\pi(u|x) \ , \end{aligned} \tag{1}$$

where $\pi(u|x)$ is the standard policy without the feedback.

The factor $\frac{P_J(J|u,x)}{P_J(J|x)}$ in (1) indicates how much better is the feedback to complete the task. If $\frac{P_J(J|u,x)}{P_J(J|x)} \leq 1$, $(\pi(u|x, J) \leq \pi(u|x))$ the feedback is irrelevant or decreases the probability of selecting one action. If $\frac{P_J(J|u,x)}{P_J(J|x)} > 1$, $(\pi(u|x, J) \geq \pi(u|x))$ the feedback improves the policy and increases the probability of select one action. Like this, the policy gives options to choose actions in one region around the action with greater probability. However, $P_J(J|u, x)$ grants a privilege to the actions in another region of the space. Thus the actions with higher probability are placed in another region that favors the actions chosen for the policy as much as actions selected by the feedback. It is clear that $P_J(J|u, x)$ and $P_J(J|x)$ are not known, and it is not possible to obtain a large sample of J_t during each step. We can consider any probability density function $\pi^*_J(u|x)$ to approximate $P_J(J|u, x)$ in (1).

The policies derived from multiple information sources must combine them because the human feedback is applied in a reinforcement learning algorithm [5], then the policy based in human feedback $\pi_\vartheta(u|x, J)$ can be expressed by:

$$\pi_\vartheta(u|x, J) \quad \propto \quad \pi^*_J(u|x) \times \pi_\vartheta(u|x). \tag{2}$$

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

The full implementation of the actor-critic with interaction is shown in Algorithm 1.

Algorithm 1: Actor-critic with human feedback

Inputs: γ, α_c, α_a.

for each episode **do**
  initialize x_t
  repeat
    Choose action u_t given by $\pi_\vartheta(u|x_t, J_t)$ with $J_t = \emptyset$ (Non feedback)
    if $rand(0,1) < \mathcal{L}$ **then**
      Get advice J_t
      Change actions $u_t \sim \pi_\vartheta(u|x_t, J_t)$
    Observe reward r_{t+1} and next state x_{t+1}
    $\delta_t \leftarrow r_{t+1} + \gamma V_{\theta_t}(x_{t+1}) - V_{\theta_t}(x_t)$
    $\theta_{t+1} \leftarrow \theta_t + \alpha_c \delta_t \nabla_{\theta_t} V_\theta(x_t)$
    $\vartheta_{t+1} \leftarrow \vartheta_t + \alpha_a \delta_t \frac{\nabla_{\vartheta_t} \pi_\vartheta(u_t|x_t, J_t)}{\pi_\vartheta(u_t|x_t, J_t)}$
  until x_t is terminal
end for

3.1 Gaussian exploration

For discrete actions, the Gibbs distribution and $\epsilon-$greedy are often used for action selection. In this paper, we define the policy of agent as a Gaussian distribution to take continuous actions [8]. The probability density function is defined as $\pi_\vartheta(u|x) = \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp\left\{-\frac{(u-\mu_\vartheta(x))^2}{2\sigma_x^2}\right\}$ where $\mu_\vartheta(x)$ and σ_x are respectively the mean and the standard deviation of the distribution. The mean is defined as a linear combination between the parameter vector ϑ and a characteristic vector $X_\vartheta(x)$ based in an artificial neural network, whereas the standard deviation is considered a scalar fixed value in this work. The function $\pi_{U_J}(u|x)$ is also considered as a Gaussian distribution with mean and standard deviation $\mu_J(x)$ and σ_J respectively. In our implementation $\mu_J(x) = \mu_\vartheta(x) + 3 * J$ where J is the feedback with values -1 or 1, and the standard deviation σ_J is considered a fixed value. The combined policy in (2) is a Gaussian distribution defined by:

$$\pi_\vartheta(u|x, J) = \frac{1}{\sqrt{2\pi\sigma_{XJ}^2}} \exp\left\{-\frac{(u-\mu_{XJ})^2}{2\sigma_{XJ}^2}\right\},$$

where $\mu_{XJ} = \frac{\sigma_J^2 \mu_\vartheta(x) + \sigma_x^2 \mu_J(x)}{\sigma_J^2 + \sigma_x^2}$ and $\sigma_{XJ}^2 = \frac{\sigma_J^2 \sigma_x^2}{\sigma_J^2 + \sigma_x^2}$.

4 Experimental setup

To evaluate the performance of our methodology, we apply it to the cart-pole balancing task. The physical parameters of the cart-pole were fixed as in [6]. The action is a force applied by the cart, set in $u \in [-10, 10]$, the state is a 4-dimensional vector defined by: the position of car, set in $x_1 \in [-2.4, 2.4]$, the velocity of the car, x_2, the pole angle, set in $x_3 \in [-\pi/15, \pi/15]$, and angular velocity, x_4. The episode ends when the pole falls, i.e., $|x_3| > \pi/15$ or the cart hits the track boundary, i.e., $|x_1| > 2.4$. In each iteration, the agent receives a reward given by $\rho(x) = cos(\frac{15}{2}x_3)$, but if the episode ends, an amount of -10 is added. If the episode has reached the 200th iterations, it ends without negative reward.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

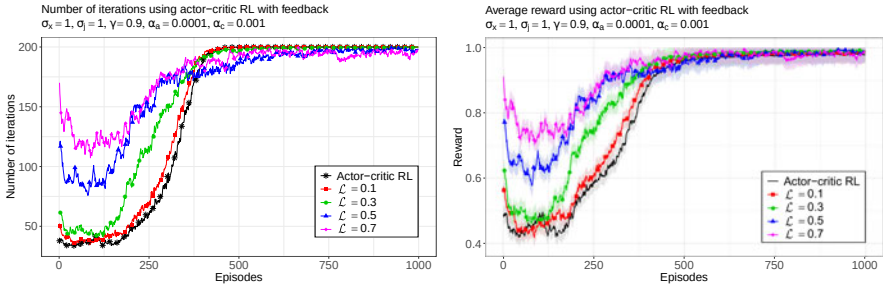

Fig. 1: Left: Average number of iterations in which the pole remains balanced per episode. The agent takes advantage of likelihood probabilities by increasing the number of iterations. Right: Average collected reward over 50 runs for continuous actor-critic and human feedback actor-critic with different probabilities of likelihood $\mathcal{L}$ and 95% confidence region of the mean based in the smoothed value. All curves were smoothed by exponential smoothing.

To imitate a real scenario, the human feedback is considered as the direction where the cart must move to balance the pole, left or right. We used a simulated oracle in the place of the human feedback. After selecting an action, the oracle computes the real necessary force for balancing the pole in the next state. Then, it returns -1 (left) if the force is less than the action, or 1 (right) if the force is greater or equal than the action.

The actor-critic is combined with a function approximator to learning the value function $V_\theta(x)$ and the mean $\mu_\vartheta(x)$. We used a neural network architecture that has a 4-dimensional input space and a 1-dimensional output space. For this setup, we only used a single hidden layer of 100 units with activation *relu6*. For the mean, the activation in the output is a hyperbolic tangent, but it scales to $[-10, 10]$.

5 Experimental results

We studied the performance of including human feedback in the continuous actor-critic algorithm. We performed simulations for different values of the likelihood $\mathcal{L}$, the standard deviation σ_X, from the policy was set at 1. The discount rate was set in 0.9 and the learning rates of the actor and the critic were set at 0.0001 and 0.001 respectively. Figure 1 (left) shows the average number of iterations from 50 runs in 1000 episodes for different likelihood probability values $\mathcal{L} \in [0.1, 0.7]$ and $\sigma_J = 1$. It is observed that, in the presence of feedback, the agent can improve its performance, i.e., for even longer the pole is upright and the car is within the bound, especially in the first episodes. However, the confidence region shows that feedback increases the error during training. Additionally, Figure 1 (right) shows the average collected reward by the agent over the episodes for different values of $\mathcal{L}$. Better performance is observed in terms of collected reward, and the maximal amount of reward is reached for all exper-

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

iments.

6 Conclusion

In this paper, we focused on including human feedback in reinforcement learning in continuous action space. We considered an external trainer that provided a guide to improve the task using an actor-critic algorithm as an approach to learning. Including feedback, the cumulative reward increases in the continuous actor-critic, nevertheless, the error also increases when the agent receives guidance during more iterations.

As future work, we consider investigating variations in the parameters of Gaussian distributions, especially defining the standard deviation dependent on the current state. Also, we want to vary the probability of likelihood during the training process to avoid adding error from the external trainer. Finally, to study the performance of our methodology in more complex scenarios where actions and feedback are represented in a larger dimension.

Acknowledgment

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and the Brazilian agencies FACEPE and CNPq. The authors also would like to gratefully acknowledge partial support by Universidad Central de Chile under the research project CIP2017030.

References

[1] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.

[2] Hado Van Hasselt and Marco A Wiering. Reinforcement learning in continuous action spaces. In *Approximate Dynamic Programming and Reinforcement Learning, 2007. ADPRL 2007. IEEE International Symposium on*, pages 272–279. IEEE, 2007.

[3] W Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, pages 9–16. ACM, 2009.

[4] Kenji Doya. Reinforcement learning in continuous time and space. *Neural computation*, 12:219–245, 2000.

[5] Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L Isbell, and Andrea L Thomaz. Policy shaping: Integrating human feedback with reinforcement learning. In *Advances in neural information processing systems*, pages 2625–2633, 2013.

[6] Andrew G Barto, Richard S Sutton, and Charles W Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics*, 13(5):834–846, 1983.

[7] Shalabh Bhatnagar, Richard S Sutton, Mohammad Ghavamzadeh, and Mark Lee. Natural actor–critic algorithms. *Automatica*, 45(11):2471–2482, 2009.

[8] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer, 1992.