Don't Delete the Two Carriage Returns Above. The Title of   
Your Paper Goes Here Using the Required Paper Title Style

Author Names Go Here Using the Author Name Style (Required)

Affiliations and Addresses Go Here Using the Affiliation and Address Style (Required)  
publications21@aaai.org (email is text roman font using Affiliation and Address Style (Required)

[*At least 3 carriage returns must separate the author affiliation block from the beginning of the two-column text block*]

Abstract

This is an example abstract. It goes here in 9 point Times New Roman type. Use of the Abstract Text style is required.

Deleting this Example of a Section Heading will Remove the Copyright Line*[[1]](#footnote-1)*

There are invisible characters after the word Line in the heading above. If you delete them, the copyright line will disappear. The AAAI copyright notice is required. This is example text. It is 10 point Times New Roman. The first sentence after the heading begins without a paragraph indent. Use of the text style (without alteration) is required.

This is example text. The second paragraph is indented 10 points, with no extra space between lines. The text is 10 point Times New Roman.

Experiments

All the experiments were run with an Nvidia GeForce GPU and Intel CPU, with a physical memory (RAM) size of 8 GB. Tabular Q-learning and SARSA were the baseline methods chosen. Our initial approach was to experiment with the performance of discrete observation and action space methods on continuous observation and control tasks. As the ranges were [-inf, inf] for each observation, we sampled across ~10k observations and clipped the maximum and minimum ranges to [-25, 25]. We discretized the continuous values into 2 buckets categorized into {0, 1}, for both the action and observation spaces. We varied the learning rate starting from {0.2, 0.3, …, 0.9}. Other parameters chosen were: = 0.99, number of episodes (epochs) = 500 and number of steps per episode = 1000. Actions were selected using an epsilon-greedy policy with decaying at a rate of . The following curves were observed for Tabular Q-learning and SARSA. As seen, for all cases Q-learning obtained a better average reward over SARSA. This can be attributed to the off-policy updates of Q-learning compared to the on-policy updates of SARSA.

It was followed by DDPG as it was known to perform well on continuous control tasks. The parameter values taken were: = 0.4, = 0.99, = 0.15, = 0.0, = 0.3, = 10000, = 100. The actor network had 2 hidden layers with 32 and 16 neurons for the 1st and 2nd hidden layers respectively. The critic network had 3 hidden layers with 32, 16, and 1 neuron for the 1st, 2nd and 3rd hidden layers respectively. The architecture of both the actor and critic networks were varied. Adaptive moment estimation (Adam) optimizer was used. The critic network was trained on minibatch size = n and the actor network on minibatch size = 1. Number of episodes (epochs) = 10 and number of steps per episode = 1000. We observe that DDPG obtains good rewards in very few episodes.

Policy Gradient

Until now, we considered action-value estimates for learning an optimal policy. As the observation and action spaces tend to grow, tabular methods prove inefficient due to the exponential growth of Q-table size, resulting in the curse of dimensionality problem. Now, we consider the class of methods that can select actions without using a value function. These are called as policy gradient methods. This method is applicable for learning optimal policies in the continuous observation space, using probability distributions over the action space.

For this, we use a parameterized policy given by where, is the policy’s parameter vector. Like the weight parameter vector ‘w’ we use for approximate action-value functions , here we use . The constraints on policy parameterization are:

1. is differentiable with respect to parameter i.e.,

In order to satisfy the above-mentioned conditions/constraints, we use a “softmax policy parameterization”.

/ – (1)

is knows as parameterized numerical preferences where . The action with the highest preferences in each state are given the highest probabilities of being selected according to equation (1). Numerical preferences can be computed by a deep Artificial Neural Network, where is the vector of all connection weights of the network or could simply be linear in features.

where is some feature vector.

The goal of RL is maximizing rewards in the long run In the policy gradient case, our objective maximizing the average reward hence, we use gradient ascent.

=

=

From product rule of calculus,

+

The challenge of this method lies in computing the gradient of the state distribution as it changes with . To address this, we use the “policy gradient theorem” which returns a simplified expression independent of .

Deep Deterministic Policy Gradient

Earlier method worked well with discrete action spaces but fails for continuous control problems. Deep Deterministic Policy Gradient (DDPG) incorporates Deterministic Policy Gradient (DPG) into the Actor-Critic structure to extend to continuous action spaces. It relies on off-policy updates using target networks.

DDPG makes use of 4 networks in total – actor network, critic network, target-actor network, and target-critic network. The actor network computes the deterministic policy

), where are the weights for the actor network. However, this policy might not explore the full state and action space. To encourage exploration, it makes use of a random process called the Ornstein-Uhlenbeck Noise . In a continuous setting, it is defined as

. In the discrete case,

: noise at time ‘t’

: decay or growth rate of the system

: asymptotic mean

: variation or size of noise

W: wiener process

The Weiner process also known as Brownian motion is a stationary process with white noise increments of a noise distribution with = 0 and = 1.

The Critic network ) evaluates state-action pairs where are its weights. The target actor and critic network denoted by and with weights and respectively are a soft copy of the weights of actor and critic network and respectively.

Replay Buffer *R* stores the transition dynamics of the environment i.e., *R* = ; t where is the memory limit. Whenever an agent takes an action in the environment, the transition tuple is added to the replay buffer. The objective of the critic network is minimizing the temporal difference between the target-critic network’s output and the estimated Q-value from its network.

): estimated target-actor network’s policy

): estimated actor network’s policy

: estimated target-critic network’s Q-value

: estimated critic network’s Q-value

n: number of random samples from replay buffer

L: critic loss

The objective of the actor network is to learn the optimal policy that maximizes the expected return. It uses the policy gradient to achieve its goal.

Across ‘n’ minibatch samples from replay buffer,

The target networks are updated using a moving average equation with parameter , which indicates the fraction of weights carried over from the original actor-critic networks to the corresponding target networks.

The pseudocode for DDPG is illustrated below

This is example indented text. It is 10 point Times New Roman. This is example indented text. It is 10 point Times New Roman. This is example indented text. It is 10 point Times New Roman. This is example indented text. It is 10 point Times New Roman.

This is an example of an extract or quotation. Note the indent on both sides. Quotation marks are not neces­sary if you offset the text in a block like this, and properly identify and cite the quotation in the text.

This Is an Example of a Figure Caption. Use of the figure caption style is required.

This Is a Section Heading   
This Style Sheet Must Be Applied   
to All Section Headings

Use of the section heading style is required.

Figures and Tables

This Is an Example of a Figure Caption or Table Caption. Captions must be placed *below* the table or figure.

You must leave at least 9 points minimum additional space between your figure or table and the figure or table caption. You must have at least 1 carriage return (12 points minimum space) after the caption. You must also have at least 1 carriage return of space (12 points additional space) before a figure or table that does not begin at the top of a page. Figures and tables must not intrude into *any* margin or gutter (this includes boxes and rules and frames).

Margins

Papers **must** be printed in two-column format on 8-1⁄2 x 11 inch paper. The margins if you use 8-1/2 x 11 inch paper must be **exactly** as follows:

• Top margin: 3⁄4 of an inch

• Left margin: 3⁄4 inch

• Right margin: 3⁄4 inch

• Bottom margin: 1-1⁄4 inches

**Be certain that the margins apply to all pages in your document, not just the first section.** The actual image size for your paper (minus margins) is 7x 9 inches (17.78 x 22.86 cm). Papers that deviate from these measurements will not be published.

Style Sheet Use and Alterations

You are required to use the custom style sheets embedded in this document without alteration, including, but not limited to Paper Title, Author Name, Affiliation and Address, Abstract Head, Abstract Text, Section Heading, Text, Text-Indent, Subsection Heading, Subsubsection Heading, List Paragraph, Figure, and Figure Caption. Do not use the built-in Word styles such as Heading 1, Header, or Normal.

Fonts

All fonts must be embedded for your paper to be published. No exceptions.

Citation Style and Reference Style

All AAAI papers must use author year citations in the text (Engelmore and Morgan 1986). Numbered references may not be used. The references must be formatted using the exact references style shown the Author Formatting Instructions for Authors Using Word. Citations must be placed in alphabetical order. Lines must be justified. References may not be numbered. No indent is allowed. The style sheet will automatically add the proper amount of space between entries; no additional space should be added.

References

Clancey, W. J. 1984. Classification Problem Solving. In *Proceedings of the Fourth National Conference on Artifi­cial Intelligence,* 49-54. Menlo Park, Calif.: AAAI Press.

Engelmore, R., and Morgan, A. eds. 1986. *Blackboard Sys­tems.* Reading, Mass.: Addison-Wesley.

Hasling, D. W.; Clancey, W. J.; and Rennels, G. R. 1983. Strategic Explanations in Consultation. *The International Journal of Man-Machine Studies* 20(1): 3–19.

Rice, J. 1986. Poligon: A System for Parallel Problem Solving, Technical Report, KSL-86-19, Department of Computer Science, Stanford University, Stanford, CA.

Robinson, A. L. 1980a. New Ways to Make Microcircuits Smaller. *Science* 208:1019-1026.

1. Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved. [↑](#footnote-ref-1)