Iowa State University

Department of Electrical and Computer Engineering

Deep Machine Learning: Theory and Practice

EE 526X

Final Project Report

Author:
Vishal Deep

Instructor:
Dr. Zhengdao Wang

December 19, 2019

Iowa State University

1 Problem Statement

The main goal of the project is to design and optimize a reinforcement learning algorithm using double Q-learning on Acrobot-v1 environment from Open AI gym package [1]. The function approximation should be done with multi-layer neural networks.

Acrobot-v1

The Acrobot-v1 environment consists of two joints. One joint is fixed and other joint is actuated as shown in fig. 1. In the initial stage the joints are hanging downwards and the goal is to make lower joint reach a certain given height. This environment is unsolved environment which means it does not have a specific reward above which we can say that it is solved.

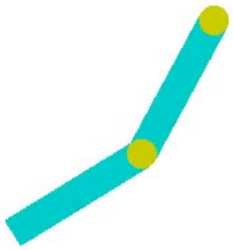

Figure 1: Acrobot-v1 from Open AI gym package

- **State**: In this environment states consists of sin and cos of the two rotational joint angles and their angular velocities. A complete state is
  [cos(theta1) sin(theta1) cos(theta2) sin(theta2) thetaDot1 thetaDot2].
- **Action**: There are three possible actions in this environment. The action is either applying +1, 0 or -1 torque on the joint between the two links.

Deep Q-learning

In the Q-learning algorithm, a virtual table of action and Q-values is used. This table could grow really fast depending on the states and actions. Therefore a better solution is to create Neural Networks (NNs) which can approximate the Q-values for every action. This is called deep Q-learning.

In case of Q-learning, taking the maximum overestimated values introduces a maximization bias in learning. To solve this problem, we use a double Q-learning algorithm which uses two separate Q-value estimators. These estimators are used to update each other and thereby providing unbiased estimation of Q-values. In this project, I will be using double Q-learning function estimators using neural networks.

2 Neural Network Model

In this project, I have used the neural network shown in fig. 2. This NN has three fully connected dense layers. The first layer consists of 48 neurons, the second one has 24 neurons, and last layer consists of 3 neurons to signify the 3 output actions. The total number of parameters are 1587.

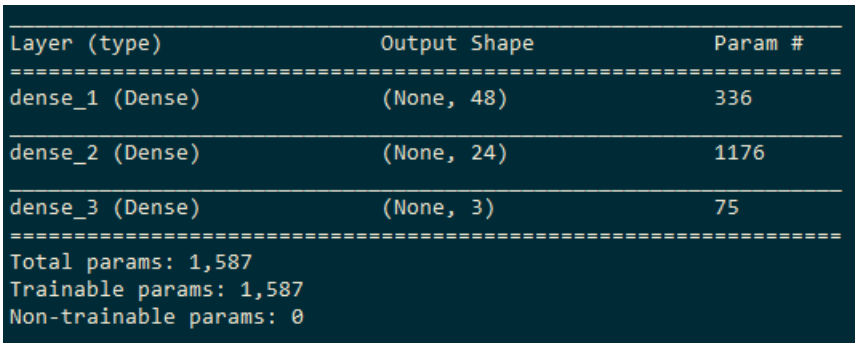

Figure 2: Neural Network Model used for function approximation in double Q-learning algorithm

3 Pseudo-code for the Algorithm

Algorithm 1 Euclid's algorithm

Initialize primary Q_A and target network Q_B
for each episode **do**
    for each step **do**
        Choose action a_t which maximize q-value
        Execute a_t and observe State S_{t+1} and Reward R_t
        Save (S_t,a$_t$, R_t, S_{t+1}) to memory
    end for
end for
procedure EUCLID(a, b)
    $r \leftarrow a \bmod b$
    while $r \neq 0$ **do** ▷ We have the answer if r is 0
        $a \leftarrow b$
        $b \leftarrow r$
        $r \leftarrow a \bmod b$
    end while
    for `<some condition>` **do**
        `<do stuff>`
    end for
    return b ▷ The gcd is b
end procedure

References

[1] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.